

Title Rexroth MTC 200/ISP 200/
MTA 200/TRANS 200
Function Interface 08VRS

Type of Documentation Application Manual

Document Typecode DOK-CONTRL-FUN*INT*V08-AW01-EN-P

Internal File Reference Document Number 120-0400-B394-01/EN

Purpose of Documentation This documentation describes the structure of function interface; it explains the listing of the new commands and the error codes.

Record of Revisions

Description	Release Date	Notes
120-0400-B394-01/EN	03.2004	Valid from version 23

Copyright © 2004 Bosch Rexroth AG

Copying this document, giving it to others and the use or communication of the contents thereof without express authority, are forbidden. Offenders are liable for the payment of damages. All rights are reserved in the event of the grant of a patent or the registration of a utility model or design (DIN 34-1).

Validity The specified data is for product description purposes only and may not be deemed to be guaranteed unless expressly confirmed in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.

Published by Bosch Rexroth AG
Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main
Telephone +49 (0)93 52/40-0 • Tx 68 94 21 • Fax +49 (0)93 52/40-48 85
<http://www.boschrexroth.com/>
Dept. BRC/ESS2 (FW)
Dept. BRC/ESM6 (DiHa)

Note This document has been printed on chlorine-free bleached paper.

Contents

1	Introduction	1-1
1.1	Division of Documentation	1-1
1.2	Application Manual (of this Document)	1-1
1.3	Reference.....	1-1
2	New in Versions FIVRS	2-1
2.1	Version 08V00.....	2-1
2.2	Version 07V00.....	2-4
2.3	Version 06V00.....	2-7
2.4	Version 05V00.....	2-10
2.5	Version 04V03.....	2-11
2.6	Version 04V02.....	2-13
2.7	Version 04V01.....	2-13
2.8	Version 04V00.....	2-14
2.9	The Data Interface Newsletter	2-15
3	General	3-1
3.1	Introduction	3-1
3.2	The Function Interface from the User's Point of View	3-1
3.3	Protection Against Dangerous Movements	3-2
4	Structure and Configuration Examples	4-1
4.1	The Structure of the Function Interface	4-1
	Logic Process	4-1
	Communication Process.....	4-2
	Management Process	4-2
4.2	Configuration Examples and Connection Options	4-3
	MPI Connection with Profibus FMS	4-3
	Bosch Rexroth GUI and Server	4-4
	Connection to the Function Interface.....	4-5
	Communication Between a Client and Bosch Rexroth Devices.....	4-5
	Communication Between Several Clients and Bosch Rexroth Devices.....	4-6
5	Programming	5-1
5.1	Guidelines	5-1
	Software for Developing User Programs (Clients) (PRO VERSION)	5-2
	Settings for the C++ Development Environment	5-2
5.2	Routines for Logging in and Logging Out	5-4

	"LogInIf" Login Routine	5-4
	"LogOutIf" Log out Routine	5-7
5.3	Data Transfer and Result Evaluation Routines.....	5-8
	"DataTransfer" Routine	5-9
	"ReadGroupItem" Routine	5-12
	"GetNumberOfGroups" Routine.....	5-13
	"GetNumberOfRows" Routine.....	5-14
	"GetNumberOfItems" Routine.....	5-14
5.4	Routine for Cyclical Reading via Pipes	5-15
	"StartCyclicPipe" Routine.....	5-15
	"ReadCyclicPipe" Routine.....	5-16
	"StopCyclicPipe" Routine	5-19
	"SuspendCyclicPipe" Routine	5-20
	"ResumeCyclicPipe" Routine.....	5-21
5.5	System Messages in the Network.....	5-22
	Data Types.....	5-23
	Programming Interface	5-23
	Example of Programming	5-27
	What SYS Messages are There and How Should I React to Them?.....	5-30
5.6	COM - Automation Interface	5-32
	IFIObjct.....	5-32
	IFIData	5-34
	Example for the Total Script.....	5-36
5.7	Network	5-37
5.8	Tips and Tricks when Working with the Interface	5-37
	Clearing the Memory Using the "KILLTASK.EXE" Tool.....	5-38
	Issuing FI Commands Using the "VBDemo" Application	5-40
	Outputting Diagnostic Messages	5-42
	Windows NT Key Combinations	5-43

6 Installing Windows NT and the Function Interface 6-1

6.1	The Windows NT Operating System.....	6-1
	Multitasking and Windows NT	6-1
	Windows NT Task Manager.....	6-2
6.2	Setting the Windows NT System Properties	6-3
	Performance Features	6-3
	Date/Time Properties	6-4
6.3	Installing the Function Interface	6-4
6.4	Directory and File Structure of the Function Interface	6-5
	Contents of the "INDRAMAT.INI" File.....	6-5
	Example Entries in the "INDRAMAT.INI" File	6-6
	Contents of the "IND_DEV.INI" File	6-6
	Example Entries in the "IND_DEV.INI" File	6-8
	Contents of the "[LW]:\Winnt\System32" System Directory.....	6-10
	Contents of the "[LW]:\Winnt\System32\Drivers" Driver Directory.....	6-10
	Contents of the "[LW]:\...\MTGUI\BasicData\Help" Directory.....	6-10

Contents of the "[LW]:\...\MTGUI\BasicData\Resource" Directory	6-11
Example Entries in the "VERSION.DAT" File	6-11
Contents of the "[LW]:\...\MTGUI\Bin" Directory	6-12
7 Error Codes	7-1
7.1 General Error Result Line	7-1
7.2 Error Codes 200 to 999	7-2
7.3 Error Codes 1000 to 1999	7-17
7.4 Error Codes 2000 to 2999	7-20
7.5 Error Codes 4000 to 4999	7-23
7.6 Error Codes 5000 to 5999	7-27
7.7 Error Codes 6000 to 6999	7-28
7.8 Error Codes 7000 to 7999	7-29
7.9 Error Codes 8000 to 8999	7-30
7.10 Error Codes 10000 and Above	7-32
7.11 Error Codes 35000 and Above	7-33
7.12 Error Codes 100000 and Above	7-33
7.13 Error Codes 110000 and Above	7-34
7.14 Error Codes 210000 and Above	7-35
7.15 SERCOS Error	7-36
7.16 Global SERCANS Error	7-38
7.17 Structure of Error File after Download	7-39
Examples for the MWCX Device Group	7-39
8 Answers to Frequently Asked Questions	8-1
8.1 Function Interface FAQs	8-1
8.2 Windows NT FAQs	8-2
9 Index	9-1
10 Service & Support	10-1
10.1 Helpdesk	10-1
10.2 Service-Hotline	10-1
10.3 Internet	10-1
10.4 Vor der Kontaktaufnahme... - Before contacting us	10-1
10.5 Kundenbetreuungsstellen - Sales & Service Facilities	10-2

1 Introduction

1.1 Division of Documentation

The "Function Interface V08" documentation was divided into two documents.

- Function Interface V08 – Application manual, DOK-CONTRL-FUN*INT*V08-AW01-EN-P, R911295419
- Function Interface V08 – Reference, DOK-CONTRL-FUN*COM*V08-AW01-EN-P, R911299217

1.2 Application Manual (of this Document)

The document describes the structure of function interface, explains the installation and gives notes for programming. Further it contains a list of the new added commands. Listing the error codes completes this documentation part.

1.3 Reference

In this document, the structure and availability of function interface command are described. Divided in device groups, the individual function interface commands are described in detail with examples.

2 New in Versions FIVRS

2.1 Version 08V00

- General Information**
- The Visual Motion device group MVMX without WinPCL was included. The Visual Motion component has been realized under SCP (Scalable Communication Platform).
- New FI Commands**
- The FI command "AAS2" reads the current spindle speed of the selected axis (MWAX device group).
 - The FI command "ADA" returns information on the currently set device address (MPCX device group).
 - The FI command "ADB" return a list of the NC databases currently set up (MWAX device group).
 - The FI command "ADW" releases diagnosis data not returned for reasons of performance (MWCX, MWAX, MWMX, MWSX, MSYX, and MWYX device groups).
 - The FI command "AFR" returns the current track speed (MWAX device group).
 - With the FI command "BCD", data is transferred for the two supported bus configurations by Phoenix and Hilscher (MWCX, MWSX and MWAX device groups).
 - With the FI command "BCI", the two supported bus configurations by Phoenix and Hilscher are configured (MPCX device group).
 - The FI command "CAM" is used for defined exiting of an external application at the FI; also refer to SDM (MPCX device group).
 - After a change of the device address, the FI command "CDA1" informs other applications of the SYS message handling (MPCX device group).
 - The FI command "CDF1" converts the 'Parameter download' file for fast access (MWAX device group).
 - The FI command "CEI2" provides information on deletion of the counts for the communication errors recorded in the protocol (MPCX device group).
 - The FI command "CFL" is used to block the log-in process for other users of the function interface (MPCX device group).
 - The FI command "CMD" generates the data the Message Integrator needs for the import process (MWCX, MWAX, MWMX, MWSX and MWYX device group).
 - The FI command "ASD" supplies the NC spindle information (MWCX device group).
 - The FI command "CNP" converts an NC program and prepares it for execution (MWAX device group).
 - The FI command "CPI" returns information on the selected process (and MWCX device groups).
 - The FI command "CPR3" starts a WIN32 application which is not logged into FI (MPCX device group).
 - The FI command "CVA" returns information on whether the axis in question is real or virtual (MWCX and MWAX device group).
 - The FI command "CWL" returns information on the current Window name (MPCX device group).
 - The FI command "DAP" writes an NC program into the controller (MWAX device group).

- The FI command "DCI" reads information on the current device configuration (MWSX, MWMX, MWAX, and MWCX device groups).
- The FI command "DCN1" is used to indicate a user-defined list box with selection options (MPCX device group).
- The FI command "DEM" deactivates the FI Exclusive Mode (all device groups).
- The FI command "DFH1" is used to copy, rename and delete files in the NC database (MWAX device group).
- The FI command "DIF" returns information on the selected device (all device classes).
- The FI command "DPA" is upload or download drive parameters (MWCX device group).
- The FI command "DPR2" exits a Win32 application (MPCX device group).
- The FI command "DWD" is available for Synax (MSYX device group).
- In case of a system error, the FI command "ECI1" returns detailed information on the cause (MWCX, MVMX, MWMX, MWSX, MSYX, MWYX and MWAX device groups).
- The FI command "PAF" deletes a file (MPCX device group).
- The FI command "GFD" reads a file list of a certain file type from a database (MWAX device group).
- The FI command "GMF" reads all active file information from the NC (MWAX device group).
- The FI command "GPP" returns global process parameters (MWAX device group).
- The FI command "HPF1" reads and writes floatingpoint hand-parameters (MWAX device group).
- The FI command "HPI1" reads and writes integer hand-parameters (MWAX device group).
- The FI command "ICA1" has been supplemented by the command "ICA2" (MWCX device group).
- The FI command "IFR1" returns any parameters from an Ident file (MWAX device group).
- The FI command "IPA" reads and writes ASCII hand-parameters in a file (MWAX device group).
- The FI command "IPF" reads and writes floatingpoint parameters in a file (MWAX device group).
- The FI command "IPI" reads and writes integer parameters in a file (MWAX device group).
- Now, the FI command "LNG" is also available for Synax (MSYX device group).
- The FI command "MDA3" deletes the files of one or all defined machine pages (MWCX device group).
- The FI command "MDI" provides for entry of an NC block for direct execution in manual mode (MWAX device group).
- The FI command "MIS" reads or writes the status of the fixed path key (jogging) in mm or 10th of an Inch (MWAX device group).
- The FI command "MTC1" reads out monitor version (MWCX, MWSX, and MWAX device groups).
- The FI command "NPA5", Reading out of parameters, is also available for the MWAX device group.

- The FI command "PCF" is used to copy WinPCL files during Remote PG (MWCX, MWAX, MWMX, MWSX, and MWYX device groups).
 - The FI command "PFR" returns the programmed value of the federate (MWAX device group).
 - With the FI command "PSD1", the system of coordinates of the controller can be shifted (MWAX device group).
 - The FI command "SCO" is used to open and close a Sercos channel. That means that the FI commands "SPA" and "SPH" exist as well (MWAX device group).
 - The FI command "SDD7" returns the module number of a step chain (MWCX, MWAX, MWMX, MWSX, and MWYX device groups).
 - The FI command "SDM" is used to read the configuration file IND_DEV.ini to exit an external application; also refer to CAM (MPCX device group).
 - The FI command "SDT" reads and writes device date and time (all device groups).
 - The FI command "SEM" activates the FI Exclusive Mode (all device groups).
 - Now, the FI command "SID" is generally available (MPCX device group).
 - The FI command "TDL1" supplies the data of all defined tools (MWAX device group).
 - The FI command "TLD7" reads and writes elements of a tools in the tool magazine (MWAX device group).
 - The FI command "TLA" uploads or downloads tool lists (MWCX device group).
 - The FI command "TQE2" returns the torque of the selected axis (MWAX device group).
 - The FI command "VMD" is used to write Visual Motion data (MVMX and MWMX device groups).
 - The FI command "ZOD7" reads and writes the data of the zero-offset table data in respect of each axis (MWAX device group).
 - The FI command "ZOD8" reads the data of the zero-offset table data of all axes (MWAX device group).
- Modifications of FI Commands**
- The FI command "APP" has been supplemented by information in setup mode (MWCX device group).
 - The FI command "CRT" has been encapsulated and implemented as BW_CRT. Please do not use CW_CRT any more (MWAX, MWCX, MWSX, MVMX, MWMX and MWYX device groups).
 - The FI command "DSI" has been supplemented by additional partial results indicating whether a PLC is available or whether the controller is in monitor mode (all device groups).
 - The FI command "IFJ" has been supplemented by the partial result "FI error class" (MPCX device group).
 - In the FI command "MAR", it must be WinPCL program entity, and not WinPCL program (MWCX, MWMX, MWSX, MWAX, MWYX device groups).
 - In the FI command "MKT2", an error in the example was remedied (MWCX device group).
 - The FI command "MTC" has been encapsulated and implemented as BW_MTC. Please do not use CW_MTC any more (MWAX, MWCX, MWSX, MVMX, and MWYX device groups).

- The FI command "MTC" for the MTA 200 has different transfer parameters than the other device groups (MWAX device group).
- In the FI command "PVF", it must be WinPCL program entity, and not WinPCL program (MWCX, MWMX, MWSX, MWAX, MWYX device groups).
- In the FI command "PVT", it must be WinPCL program entity, and not WinPCL program (MWCX, MWMX, MWSX, MWAX, MWYX device groups).
- The FI command "SID" has been supplemented by other partial results (all device groups).
- The FI command "TDD" has been supplemented by the command "TDD" (MWCX device group).
- The FI command "TDF" has been reworked (MWCX device group).
- The FI command "TDI" has been reworked (MWCX device group).
- The FI command "TDR" has been reworked (MWCX device group).

2.2 Version 07V00

- General Information**
- The Visual Motion device group MWMX was included. For the commands, see the MWSX device group, as the WinPCL part is identical with this group. The Visual Motion component has been realized under SCP (Scalable Communication Platform).
 - The MWYX device group has been included. The commands are a combination of the MSYX and MWSX device group, but united in a chapter of their own.
 - The item „COM – Automation Interface“ was included in the chapter “Programming”.
 - The FI command “PVS” was removed from the description as the existing FI command “PVF” possesses the efficiency of “PVS”, plus the array handling. Accordingly, do not use “PVS” any more.
 - From IF Version 07, the FI commands “DPN”, “DPP”, “IPP”, “NPC”, “NPD”, “NPI”, “PPD” and “PPN” are re-activated and have been included in the description (MWCX device group).
- New FI Commands**
- FI command "ANM" supplies the size of the current NC magazine (MWCX device group).
 - FI command "ASD" supplies the current spindle data (MWCX device group).
 - FI command "CCA" causes an Upload of NC cycles by an Upload file (MWCX device group).
 - FI command "CEI" displays the counts of the logged communication errors (MPCX device group).
 - FI command "DCA" causes an Upload of NC D corrections by an Upload file (MWCX device group).
 - FI command "DCT" sets the Timeout for a device or sets the time back to the default value (MWCX-, MWSX-, MWMX-, MWAX-, MSYX-, MWYX- and MSCX device group).
 - FI command "DSF" deletes the FI command Stack management (MPCX device group).
 - FI command "ICA" initializes a communication address with new parameters (MPCX device group).
 - FI command "IFS" supplies the current occupancy state of FI command Stack management (MPCX device group).

- FI command „LDT“ reads and writes the local PC date and the local PC time of the day (MPCX device group).
- FI command "MDA" has been extended by the command MDA4 with which all Machine Data Page definitions can be deleted in the selected device (MWCX device group).
- FI command "MSG" is used to read System Messages (MPCX-, MSCX-, MWCX-, MWSX-, MWMX-, MWAX- device group).
- FI command "MTC" is used to read the Firmware identification from the different control components (MWCX-, MWSX-, MWMX- and MWAX- device group).
- FI command "NCA" causes an Upload of NC programs by an Upload file (MWCX device group).
- FI command "NEA" causes an Upload of NC events by an Upload file (MWCX device group).
- FI command "NUA" causes an Upload of NC zero points by an Upload file (MWCX device group).
- FI command "NVA" causes an Upload of NC variables by an Upload file (MWCX device group).
- FI command "PAD" sets a parameter set inactive if the device is in Offline mode (MWCX device group).
- FI command "PAF" sorts a parameter download file (MPCX device group).
- FI command "PAS" sets a parameter set active if the device is in Offline mode (MWCX device group).
- FI command "PDD" supplies data for the ProVi criteria analysis (MWCX-, MWSX-, MWMX- and MWAX device group).
- FI command "POB" is used to write and read a PC port address (BYTE access) (MPCX device group).
- FI command "POI" supplies the current position information of all axes (MWAX device group).
- FI command "POW" is used to write and read a PC port address (WORD access) (MPCX device group).
- FI command "PVA" is used to write and read Provi Message files (MWCX-, MWSX-, MWYX- and MWAX device group).
- FI command "PVM" was extended by the command "PVM4" (MWCX-, MWSX-, MWMX- and MWAX device group).
- FI command "PVR" executes an Upload or Download of PLC retain variables (MWAX-, MWCX-, MWMX-, MWSX- and MWYX device group).
- FI command "REP" supplies data to return to the contour (MWCX device group).
- FI command "SDP" starts a FI device interrogation cycle (MPCX device group).
- FI command "SDS" sets the device status (ON/OFF) which is also entered in IND_DEV.ini (MSCX-, MWCX-, MWSX-, MWMX-, MSYX-, and MWAX device group).
- FI command "TPI" supplies information about grippers, spindles and tool magazine locations (MWCX device group).

Modifications of FI Commands

- FI command "AMM" reports the active mechanism errors and was increased by a file name for additional information for the message text or the extended text in the return value (MWCX-, and MWAX device group).

- FI command "API" supplies always 0 as index of the active parameter set (MWCX device group).
- FI command "APP" supplies the active NC block number and was extended in the reply by an output in the setup mode.
- FI command "ART" was extended by "**binary** reading of the current axis reference table for an Offline device" (DeviceStatus=OFF) (MWCX device group).
- FI command "ASM" reports the active system errors and was increased by a file name for additional information for the message text or the extended text in the return value (MWCX-, and MWAX device group).
- FI command "CCP" supplies the configuration settings from IND_DEV.ini and was extended by columns 14 (device protocol) and 15 (device simulation) in the reply (MPCX device group).
- FI command "DCD" supplies the value of D correction index and was extended by measuring unit [mm/inch] in the return (MWCX device group).
- FI command "DCR" reads and writes the values of a D correction set and was extended by measuring unit [mm/inch] in the return (MWCX device group).
- FI command "DIS" was revised and realized as B command (MWCX-, MWSX-, MWMX-, MWAX device group).
- FI command "DSI" supplies the most essential information about device status and was extended by columns 12 (device simulation switched on) and 13 (device status information) in the reply (MSCX-, MWCX-, MWSX-, MWMX-, and MWAX device group).
- FI command "DTY" was revised and realized as B command (MWCX-, MWSX-, MWMX-, MWAX-, MSCX-, MWSYX- and MWYX device group).
- FI command "DWD" supplies diagnosis messages and was extended in return value by the criteria analysis and a file name for additional information (MWCX-, MWSX-, MWMX-, and MWAX device group).
- FI command "ERI" now also supplies an error text with a WIN NT error code (MPCX device group).
- FI command "MKT" is used to write the GUI-SK 16 block in the PLC and was extended by command MKT2 (MWCX-, MWSX-, MWMX- and MWAX device group).
- FI command "NCM" supplies NC messages and was extended by a file name for additional information for the message text in the return value (MWCX device group).
- FI command "NVS" was set during writing NC variables to data type "LONG" or "Doublereal" (MWCX device group).
- FI command "PVM" supplies ProVi messages and was enlarged in return value by the criteria analysis and a file name for additional information for the message type or the extended text (MWCX-, MWSX-, MWMX-, and MWAX device group).
- FI command "ZOD" was extended by measuring unit [mm/inch] during reading an offset page (MWCX device group).

Authorized Errors

- FI command "DIS1" supplies "--" in all result columns if no valid parameter set is in the selected device (MWCX device group).

2.3 Version 06V00

- General Information**
- Chapter 1 has been extended to provide information on safety under the heading “Protection against dangerous movements”.
 - The section in the chapter entitled “Programming” concerning SYS messages has been revised.
 - A separate chapter entitled “Literature” has been appended.
 - Documentation of previously undocumented and new commands for the software standard 05-21V00 WIN-HMI.
 - Box 19 in the table on basic value range data is used to classify tools. The user can no longer edit (MWCX device group).
 - The MTCX device group has been almost entirely converted to the MWCX device group. This has created a designation for the newly introduced WinPCL.
 - Only the MTVNC remains in the MTCX device group. Its instruction set is a subset of the MWCX device group. Individual commands are listed only in table form with detailed reference to explanations in the MWCX device group.
 - The MISX device group has been converted to the MWSX device group. This has created a designation for the newly introduced WinPCL.
 - The MTAX device group has been converted to the MWAX device group. This has created a designation for the newly introduced WinPCL.
 - From IF Version 06 the FI commands "IPP", "NPC", "NPD", "NPI", "PPD" and "PPN" are no longer valid and have been removed from the description (MWCX device group).
 - The list of error codes has been extended.
- New FI Commands**
- The FI command "ADM" supplies all messages from the Andron NC (MWAX device group).
 - The FI command "ARF" indicates the reference flags of an axis for a process (MWCX device group).
 - The FI command "ART" returns the complete axis reference tables of a system (MWCX device group).
 - The FI command "ATR" returns the complete basic data and cutter data of the current processing tool (MWCX device group).
 - The FI command "ATU" causes the NC to accept the data record of the current tool changed after editing (MWCX device group).
 - The FI command "BFJ" interrupts FI jobs (MPCX device group).
 - The FI command "CCA" triggers the downloading of NC cycles by means of the download file (MWCX device group).
 - The FI command "CPR" starts a Win32 application (MPCX device group).
 - FI command "DCA" triggers the downloading of NC D corrections by means of the download file (MWCX device group).
 - The FI command "DPR" exits a Win32 application (MPCX device group).
 - The FI command "DSI" returns the most important information on the status of the device (MSCX, MWCX, MWSX, and MWAX device groups).

- The FI command "DTC2" returns tool management data such as basic user data and cutter user data (MWCX device group).
- The FI command "DWD" outputs all diagnostic messages (MWCX, MWSX, and MWAX device groups).
- The FI command "EAD" returns which Andron diagnostic types are available in a module (MWAX device group).
- The FI command "EDE" returns whether or not there are errors present (MWCX, MWSX, and MWAX device groups).
- The FI command "EDW" returns all diagnostic window types (MWCX, MWSX, and MWAX device groups).
- The FI command "END" returns which NC diagnostic types are available in a module (MWCX device group).
- The FI command "EPD" returns which PLC diagnostic types are available on a control unit (MWCX and MWSX device groups).
- The FI command "EPT" returns all ProVi types (MWCX, MWSX, and MWAX device groups).
- The FI command "EST" queries the error state of a variable (MWCX, MWSX and MWAX device groups).
- The FI command "EXD" shows the extent to which a step or action can be executed (MWCX, MWSX and MWAX device groups).
- The FI command "GDB" writes/reads the data for the general FI data buffer (MPCX device group).
- The FI command "MDA" uploads or downloads machine data records (MWCX device group).
- The FI command "MDS" is for writing and reading machine data (MWCX device group).
- The FI command "MFD" loads the message text into the device indicated (MWCX, MWSX, and MWAX device groups).
- The FI command "MKS" returns information on the machine buttons (MWCX, MWSX, and MWAX device groups).
- The FI command "MKT" writes the GUI-SK16 block in the PLC (MWCX, MWSX and MWAX device groups).
- The FI command "NCA" triggers the downloading of NC programs by means of the download file (MWCX device group).
- The FI command "NCM" returns all messages from the Bosch Rexroth NC (MWCX device group).
- The FI command "NEA" triggers the downloading of NC events by means of the download file (MWCX device group).
- The FI command "NPA5" returns a list of a maximum of 10 parameters of the same parameter type (MWCX device groups).
- The FI command "NST" brings the operating system to a stop (MPCX device group).
- The FI command "NUA" triggers the downloading of NC zero points by means of the download file (MWCX device group).
- The FI command "NVA" triggers the downloading of NC variables by means of the download file (MWCX device group).
- The FI command "PAA" uploads or downloads parameter records (MWCX and MTAX device groups).
- The FI command "PDT" returns parameters from the definition table for the selected device in binary form (MWCX device group).
- The FI command "PHD" generates a physical directory with the assistance of the BDI (MPCX, MSCX, MWAX, MWSX, MWCX, and MSYX device groups).

- The FI command "PSM" initiates the most important SYS messages with regard to the PCL programming interface (MWCX, MWSX and MWAX device groups).
- The FI command "PVM" returns all ProVi messages (MWCX, MWSX, and MWAX device groups).
- The FI command "RPR" informs the caller that the program is now active (MPCX device group).
- The FI command "SDD" returns data for the step chain diagnosis (MWCX, MWSX, and MWAX device groups).
- The FI command "SFD" returns data for a step chain (MWCX, MWSX, and MWAX device groups).
- The FI command "SFE" returns all the step chain messages for a module (MWCX, MWSX, and MWAX device groups).
- The FI command "SFM" returns the operating mode of a step chain (MWCX, MWSX, and MWAX device groups).
- The FI command "SFW" focuses the screen (MPCX device group).
- The FI command "SSM" is for issuing SYS messages (MPCX device group).
- The FI command "WLA" requests free watch list allocations (MWCX-device group).
- The FI command "WLF" frees up requested watch list allocations (MWCX-device group).

Modifications of FI Commands

- The FI command "AAS" returns information about the current axis speed and, in return, has been extended by an additional unit of measurement [inch/min] (MWCX device group).
- The FI command "AFR" returns information about the current feed velocity and, in return, has been extended by an additional unit of measurement [inch/min] (MWCX device group).
- The FI command "APO" returns the current axis position value and, in return, has been extended by an additional unit of measurement [inch] (MWCX device group).
- The FI command "CPO" returns the current axis position command value and, in return, has been extended by an additional unit of measurement [inch] (MWCX device group).
- The FI command "DTG" returns the distance to go of an axis and, in return, has been extended by an additional unit of measurement [inch]" (MWCX device group).
- The FI command "EPO" returns the end point of an axis movement and, in return, has been extended by an additional unit of measurement [inch]" (MWCX device group).
- The FI command "MFR" returns the maximum feedrate and, in return, has been extended by an additional unit of measurement [inch/min]" (MWCX device group).
- The FI command "OPD" returns the optimum position distance of an axis and, in return, has been extended by an additional unit of measurement [inch]" (MWCX device group).
- The FI command "PFR" returns the value for the programmed feedrate and, in return, has been extended by an additional unit of measurement [inch/min]" (MWCX device group).
- The FI command "SLA" returns the current servo lag of an axis and, in return, has been extended by an additional unit of measurement [inch]" (MWCX device group).

- The FI command "TDR" returns the complete basic data and tool edge data of a tool and has been extended by information concerning optional data elements (MTCX device group).
- The FI command "TLD" returns elements of the basic data and cutter data of a tool and has been extended by information concerning optional data elements (MTCX device group).

2.4 Version 05V00

- | | |
|--|---|
| General Information | <ul style="list-style-type: none"> • Documentation of previously undocumented and new commands for the software standard 05-19V00 WIN-HMI. • The chapter entitled "Practical Use of Tool Commands" has been included to better explain the tool commands (MTCX device group). |
| FI Commands Changes / Additions | <ul style="list-style-type: none"> • The FI command "ATP" returns information on the current tool location (MTCX device group). • The FI command "ERI1" returns the error text and the additional text of an FI error code or a NACK error number (MPCX device group). • The FI command "MAR" reads the reference names of a PLC variable (MTCX device group): • FI command "MTD1" for reading and writing machine user data has been expanded (MTCX device group). The FI command "MTD" is no longer used for reading! • The FI command "PVF" is for the formatted reading and writing of PLC variables, arrays and structures (MTCX device group). • The FI command "PVS" is for reading and writing PLC variables, arrays and structures (MTCX device group). • The FI command "PVT" reads the declaration of PLC variables, including structures and arrays (MTCX device group). • The FI command "TDR" returns the complete basic data and tool edge data of a tool (MTCX device group). • The FI command "TLB" returns the basic data of the tool list (MTCX device group). • The FI command "TLD" returns elements of the basic data or cutter data of a tool in the tool memory (MTCX device group). • The FI command "TLE" returns the cutter data of the tool list (MTCX device group). • The FI command "TII" initiates a tool replacement (MTCX device group). • The FI command "TMV" moves an entire tool data record comprising the basic data and defined cutter data (MTCX device group). • The FI command "TRS" resets the percentage life time of a tool to 100% (MTCX device group). • The FI command "TIF" terminates a tool replacement (MTCX device group). • The result [*X] of an FI command with invalid parameters has been changed into [--] (2 hyphens). This concerns the following commands of the MTCX device group:
AAS1, AAS2, APO1, APO2, ASO1, ASS, CPO1, CPO2, DTG1, DTG2, EPO1, EPO2, MSS, OPD1, OPD2, PSS, SLA1, SLA2, TQE1, TQE2. • Chapter 05 contains the table "Logical Linking of FI Commands" with direct links to the individual commands in the help file. |

- The table "Availability of FI Commands" in Chapter 05 has been split into separate sections for each device group. It now contains direct links to the individual commands in the help file.
- The new device group MSYX (SYNTAX200-P, SYNTAX200-R) has been incorporated into the documentation.

2.5 Version 04V03

- | | |
|--|--|
| General Information | <ul style="list-style-type: none"> • Documents previously undocumented and new commands for the software standard 05-18V06 WIN-HMI. • Includes a table with logical links for the FI commands in Chapter 5. • Inclusion of a table above the command times in chapter 05. • Includes the component types for the NC and PLC hardware in the file "IND_DEV.INI". Expansion of the FI command "DTY" by the output of the component types "Componenttype1=" and "Componenttype2=". |
| Identification of versions, Bosch Rexroth software components | <p>Entries in the "C:\IND_BASE\INDDRAMAT.INI" file:</p> <ul style="list-style-type: none"> • IfDIIMode = 04.20 • IfVersion = 04V03 <p>Software components contained normally within the function interface :</p> <ul style="list-style-type: none"> • all Bosch Rexroth System 200 user interfaces for Version 18V06. |
| FI Commands Changes / Additions | <ul style="list-style-type: none"> • The FI command "AMM7" for outputting active mechanism messages and errors (MTAX device group). • FI command "DCD1": The values of a D-correction register are read out (MTCX device group). • The FI command "DTC1" returns the most important system parameter data of the tool management (MTCX device group). • The FI command "FIT1" returns the additional text of an FI error code or a NACK error number (MPCX device group). • The FI command "PTC1" returns the tool management data of all defined NC processes. The FI command "PTC2" returns the tool management data of a defined NC process. Only for the MTCX device group. • The FI command "PPD" reads an NC program directory (MTCX device group). • The FI command "PPN" converts an NC program from the NC program directory into an ASCII file and vice versa (MTCX device group). • The FI command "PPP" changes the designation of an NC program package (MTCX device group). • The FI command "PVF" handles the formatted reading and writing of PLC variables, arrays and structures (MTCX, MISX and MTAX device groups). • The FI command "PVT" reads the type of PLC variables, including structures and arrays with their elements (MTCX, MISX and MTAX device groups). • The FI command "SID1" returns information regarding the installation. This information includes installation paths, the software version used, DLL mode, plus service pack and release information. Valid for all device groups. |

- The FI command "SLI" returns the single data from the SPS long ID such as the number, name and length of the program, the compiling date and more (MTCX, MISX and MTAX device groups).
- The FI command "TII" initiates a tool replacement (MTCX device group).
- The FI command "TIF" initiates the end of a tool replacement (MTCX device group).
- The FI command "TLB1", or "TLB2" returns basic data of the tool list such as tool memory, designation, correction type and more (MTCX device group).
- The FI command "TLE1", or "TLE2" returns cutter data of the tool list such as tool memory, location number, tool status and more (MTCX device group).
- The FI command "DCR1" has been implemented for reading and writing the D-correction register with the newly formatted output. (MTCX device group). The FI command "DCR" is no longer used!
- The FI command "DTY1" for outputting the device type has been expanded by the corresponding components. (MTCX, MSCX, MISX, and MTAX device groups). The FI command "DTY" is no longer used!
- The FI command "ZOD" for reading and writing data from the zero offset table has been expanded to include the FI commands "ZOD1" and "ZOD2" (MTCX and device group). The FI command "ZOD" is no longer used!
- New, speed-optimized FI command "GPP" for reading out the global process parameters. (MTCX device group).
- New, speed-optimized FI command "NPD3 and "NPD4" for the NC download of small NC part programs. (MTCX device group).
- "NPD1" and "NPD2" commands for the NC download expanded by the value to be written "Initialization" (MTCX device group).
- Expansion of the FI command "CCP" by the output of the component types "Componenttype1=" and "Componenttype2=" (MPCX device group).
- New FI command "CCP5" for outputting the configuration data of the device that is addressed via the indicated device address (MPCX device group).
- New FI commands "CMA, CMF and CMI" for reading and writing of CMOS RAM, ASCII, floating point and integer parameters. (MTAX Device Group)
- New FI command "CRT" for triggering a control reset for the selected device (MTAX device group).
- New FI command "NPS" for preselecting the NC program located in the NC memory for processing (MTCX device group).
- New FI command "NMM" for selecting the NC memory for the NC program processing (MTCX device group).
- Addition to the documentation of the FI command "SPA" of the Ident. Number formats. The error return in the event of a form error in the value to be written has also been improved (MTCX, and MSCX device group).
- New FI command "TDax, TMV and TRM" for editing complete tool data records (MTCX device group).

2.6 Version 04V02

- | | |
|--|---|
| General Information | <ul style="list-style-type: none"> • New chapter in the documentation: "Answers to Frequently Asked Questions (FAQ)". • Most resource wastage has been eliminated in Service Pack 2 of the user interface GUI 18V05. |
| Identification of versions, Bosch Rexroth software components | <p>Entries in the "C:\IND_BASE\INDRAMAT.INI" file:</p> <ul style="list-style-type: none"> • IfDIIMode = 04.10 • IfVersion = 04V02 <p>Software components contained normally within the function interface :</p> <ul style="list-style-type: none"> • all Bosch Rexroth System 200 user interfaces for Version 18V05 with Service Pack 3. |
| FI Commands Changes / Additions | <ul style="list-style-type: none"> • Module commands MCD1, MCM1 and MCS1 enabled for the MISX device group. • Module commands MCD1, MCM1, MCP1, MCS1, MAP1 enabled for the MTAX device group. • Addition of CR_APO2, CR_DTG2, CR_CMA, CW_CMA, CR_CMI, CW_CMI, CR_CMF and CW_CMF for the MTAX device group. |
| Basic processes Changes / Additions | <ul style="list-style-type: none"> • Waste of resources in logic process resolved. • Expansion from 15 to a max. of 255 group requests during cyclic requests (see "Routines for Cyclic Reading via Pipes"). |

2.7 Version 04V01

- | | |
|--|---|
| General Information | <ul style="list-style-type: none"> • Inclusion of the PRO-VERSION as a software option in the installation program. |
| Identification of versions, Bosch Rexroth software components | <p>Entries in the "C:\IND_BASE\INDRAMAT.INI" file:</p> <ul style="list-style-type: none"> • IfDIIMode = 04.10 • IfVersion = 04V01 <p>Software components contained normally within the function interface :</p> <ul style="list-style-type: none"> • all Bosch Rexroth System 200 user interfaces for Version 18V05. |
| FI Commands Changes / Additions | <ul style="list-style-type: none"> • Expansions to the device-independent access functions. • New FI command "CRT" for triggering a control reset. (MTCX and MISX device groups). |
| Basic processes Changes / Additions | <ul style="list-style-type: none"> • Error correction of the telegram optimizer (correction of timeout recognition). • New SYS message "MSG_PC__ALIVE" in PC network. |

2.8 Version 04V00

In contrast to the previous 03VRS versions, fundamental changes have been made in this version:

- | | |
|--|---|
| General Information | <ul style="list-style-type: none"> • Delivery of a Visual Basic example connection to the function interface (application including source codes). • Delivery of printed documentation as online help in Windows NT/95 help format. • Provision of an installation program for the function interface. • New! FI commands for an NC download. (MTCX device group) |
| Identification of versions, Bosch Rexroth software components | <p>Entries in the "C:\IND_BASE\INDRAMAT.INI" file:</p> <ul style="list-style-type: none"> • IfDIIMode = 04.00 • IfVersion = 04V00 <p>Software components contained normally within the function interface :</p> <ul style="list-style-type: none"> • all Bosch Rexroth System 200 user interfaces for Version 18V04 with Service Pack 2. |
| FI Commands Changes / Additions | <ul style="list-style-type: none"> • FI command "XYZ" are implemented as "XYZ1" with re-formatted output: AAC1, AAS1, ADN1, AFO1, APO1, ARO1, ASO1, AZB1, MFO1, MRO1, MSO1. The FI command "XYZ" should no longer be used! • The FI command "ABN" has been replaced by the FI commands "ASM" and "AMM". • New functions for the BOF/GB0 for calling up WIN32 applications. • New functions for a WIN32 application at the function interface for calling up the BOF/GB0. • Expansion of the data structures for the BOF/GB0. Support for Bosch Rexroth TRANS 200 device types. |
-
- Note:** The Bosch Rexroth devices TRANS 200-P and TRANS 200-R are still at the development stage and therefore cannot yet be ordered.
-
- | | |
|--|---|
| Basic processes Changes / Additions | <ul style="list-style-type: none"> • Expanded function calls for the device configuration. • Message for activating or deactivating a PC in the PC network. • Expansions to the device-independent access functions. • Error correction of the telegram optimizer (correction of timeout recognition). • Expansion in the PLC data optimizer. • Enlarged input buffer for the telegram optimizer. • Reworking of the internal interface. • Error correction in data provision by means of the "ReadGroupItem" routine. • Error correction of the communication channel. • Error corrections in the internal DLL interfaces. • Correction of the INDIF200.DLL (correction of the binary result for spindle data). |
|--|---|

- Changes in LogOutf(), with regard to the selective KILLTASK
- Reworking of the COMVIEW interface for WIN200.
- Moving of the new SYS-Message interface into the file "INDIF000.H".

2.9 The Data Interface Newsletter

We will be informing you by email of new developments and updates to the Bosch Rexroth Products MPI and Function Interface.

Please send an email request with the message **subscribe** to:

owner-ml_datainterface@proxy.indramat.de

To unsubscribe, please proceed identically, but instead write

Message: **unsubscribe**.

Note: Your email address will be kept confidential and not passed on to third parties.

3 General

3.1 Introduction

The Bosch Rexroth Function Interface is a unified data interface produced by Bosch Rexroth for application programs (sometimes referred to as clients) based on the Windows NT platform.

Requirements To obtain free access to data on existing NC and PLC data, it is necessary to provide a data interface that is as open, reasonably priced and as simple to handle as possible. The main requirements are to be able to access CNC/PLC data with a large range of functions and rapid access and reaction speeds. Several clients can access the data.

Objective The Bosch Rexroth Function Interface aims to do exactly that, i.e. it allows access to all required control data via a compact, functional interface. This therefore allows the customer to completely create his own user interface in the programming languages Visual C++ or Visual Basic. The user is thus provided with a powerful interface with which he can communicate with Bosch Rexroth devices and user interfaces using mnemonic function calls. The Function Interface is therefore a universal solution for data communication.

Availability This description is valid for the following versions:

WinHMI:	23Vxx
Function Interface:	08Vxx
Windows NT Workstation/Server:	4.0
Visual C++:	5.0 + 6.0
Visual Basic:	5.0 + 6.0

3.2 The Function Interface from the User's Point of View

The Function Interface is a client (service requester) – server (service provider) interface and provides the user with a library (DLL) for communication services. The services, i.e. the functions of the DLL, fulfill the communication tasks that are required for reading individual data, the cyclic reading of data, the cyclic reading of data groups, the writing of data and for processor communication with Bosch Rexroth user interfaces and devices.

The Function Interface can communicate with a maximum of ten independent user programs (clients). This means a user program can, for example, be a customized user interface, a Bosch Rexroth OPC-Server or a communication driver to another data interface.

Up to ten parallel communication channels, one for each client, are supported between the Function Interface and the device. One communication channel can connect with a maximum of 64 data terminal devices.

The physical communication address in this case can be a serial interface (RS232/RS485), a Dual-Port-RAM or a Shared Memory area.

3.3 Protection Against Dangerous Movements

Dangerous movements can be caused by the faulty control of connected motors. The reasons can be extremely varied:

- careless or faulty wiring or cabling,
- errors in operating the components,
- faults in the measured-value and signal transmitters,
- defective components, and/or
- errors in the software.

These faults can occur immediately after switching on or at any time during operation.

The Bosch Rexroth Function Interface is communication software which can be used to change the values of variables in the control unit.

As far as possible, monitoring in the drive components precludes faults in the connected drives. Where personnel safety is concerned, particularly where there is a risk of physical injury and/or damage to property, this fact should not be relied on exclusively. Until the built-in monitoring systems become active faulty drive movement is always to be expected; the degree of movement depends on the type of control unit and the operating status.

**DANGER****Dangerous movements! Risk of death, injury, severe physical injury or damage to property!**

⇒ For the reasons given above, protection of personnel is to be guaranteed by means of monitoring or other higher-ranking measures within the system.

For this purpose risk and fault analysis are to be provided for by the system designer according to the specific conditions within the system. The safety regulations applicable for the system are also to be taken into consideration. Arbitrary movements in the machine or other erratic functions can occur if safety devices are switched off, bypassed or activated wrongly.

To avoid accidents, physical injury and/or damage to property:

⇒ Do not stay within the motional range of the machine or machine parts. Possible measures to prevent personnel accidentally accessing the machine:

- protective fencing
- protective grid
- protective cover
- light barrier

⇒ Fencing and covers must be adequately secured against the maximum possible force of movement.

⇒ Position emergency stop switches within the immediate vicinity and so that they are easily accessible. Check that the emergency stop equipment is functioning before start-up. Do not operate the device if the emergency stop switch is not functioning correctly.

⇒ Protect against the device starting unintentionally by providing safety isolation for the drive's power connection by means of an emergency stop circuit or by using a safe starting lockout function.

⇒ Before accessing or entering the danger area bring the drives safely to a standstill.

⇒ Secure vertical axes against falling or slipping after switching off the motor power by, for example:

- mechanically locking the vertical axis,
- providing external brake/catching/clamping mechanisms or
- adequately counterbalancing the axis.

The standard motor holding brake provided or an external motor holding brake controlled directly by the drive controller are not sufficient on their own to guarantee the safety of personnel!

⇒ De-energize electrical equipment by means of the main switch and secure against reconnection during:

- maintenance and repair work
- cleaning work
- lengthy breaks in operation

⇒ Avoid operating high-frequency, remote controlled and radio devices in the vicinity of the device electronics and their power supply cables. If the use of these devices cannot be avoided, check the system and installation for possible faults in all working areas before switching on the system. If necessary, the system will require a

special EMC test.

4 Structure and Configuration Examples

4.1 The Structure of the Function Interface

Viewed as a complete component, the function interface consists of the following three basic processes:

- Logic process
- Communication process and
- Management process

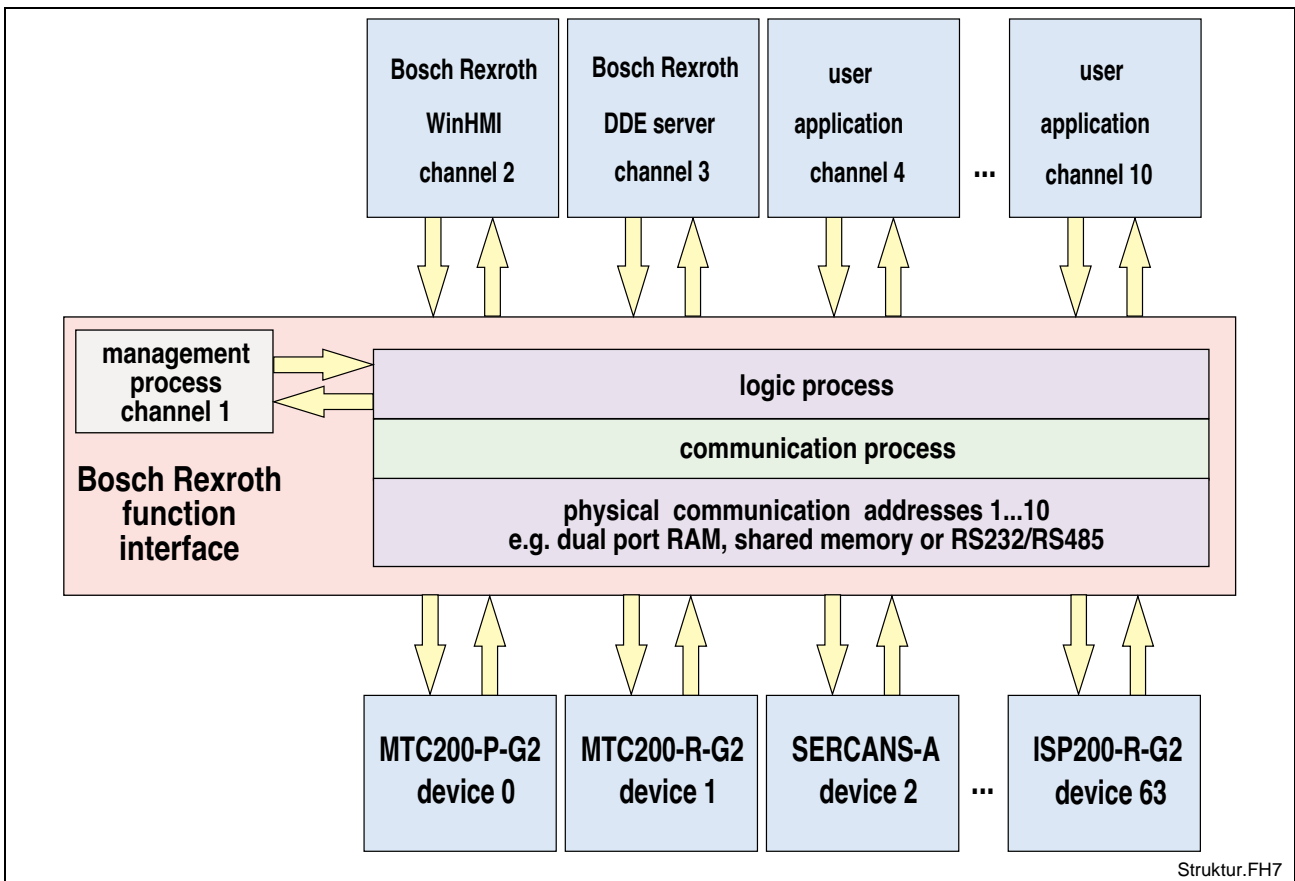


Fig. 4-1: Structural overview of the function interface

Logic Process

The logic process provides the user program (client) with the actual data interface along with the services described in the previous chapter. To do this, it opens a logic channel (LOG channel) for every connected client. The number of active LOG channels therefore directly depends on the number of the connected clients. Furthermore, the logic process is a data interface to all defined devices and to the management and status terminal data that are monitored by the management process. As far as the user program (client) is concerned, the logic process is the server. On the other hand, the logic process provides the connection to the communication process via a shared memory. Data is distributed to the individual logic channels via this connection.

Note: The maximum number of LOG channels available to function interface applications is administered dynamically. If a function interface application exceeds this limit then an error message is issued. The chapter 4 "Programming" describes how the data interface to the logic process is to be handled and how to allow data access from the client to the function interface.

Communication Process

The communication process executes the requirements of the various logic channels, generates communication to the devices together with the time allocations and initializes all devices configured on starting. The communication process and the logic process thereby allow data access to the respective Bosch Rexroth devices (MTC200-P-G2, ISP200-P-G2, SERCANS etc.).

On the one hand, it exchanges Bosch Rexroth telegrams with the logic process via the shared memory. On the other hand, it exchanges internal telegrams with the configured Bosch Rexroth devices via the dual port RAM or via a serial interface. The communication process opens a communication channel (thread) for each of these configured devices. It thereby allows simultaneous communication via various communication methods and via several parallel interfaces.

Note: The configuration of the communication addresses as well as the processing options of the individual Bosch Rexroth devices are carried out by the Bosch Rexroth system configurator and stored in the "IND_DEV.INI" file (see Chapter 5.4 "Directory and File Structure of the Function Interface").

Management Process

The management process is designed as an internal user program and uses the first LOG channel for communication with the logic process. It provides static and dynamic configuration data, delivers the more valuable functions and creates the corresponding data structure for each configured device. The management process thereby collects, for example, MTC200-P-G2 control data together with data from the PC hard drive which a client can then access. The management process thereby fulfills administrative tasks.

Note: The chapter "Function Interface Commands" chapter describes how to access data from the Bosch Rexroth devices and the PC hard drive.

4.2 Configuration Examples and Connection Options

MPI Connection with Profibus FMS

The following figure shows the connection of the Bosch Rexroth MPI (Multi-Protocol-Interface) with Profibus FMS design-type and additional clients to the function interface.

The first LOG channel (logic channel 1) of the function interface is used by a user program (client), e.g. a customized user interface. The Bosch Rexroth user interface (**WinHMI = Windows Human Machine Interface**) runs under Windows NT. The MPI connection to the function interface is made via the second LOG channel (logic channel 2).

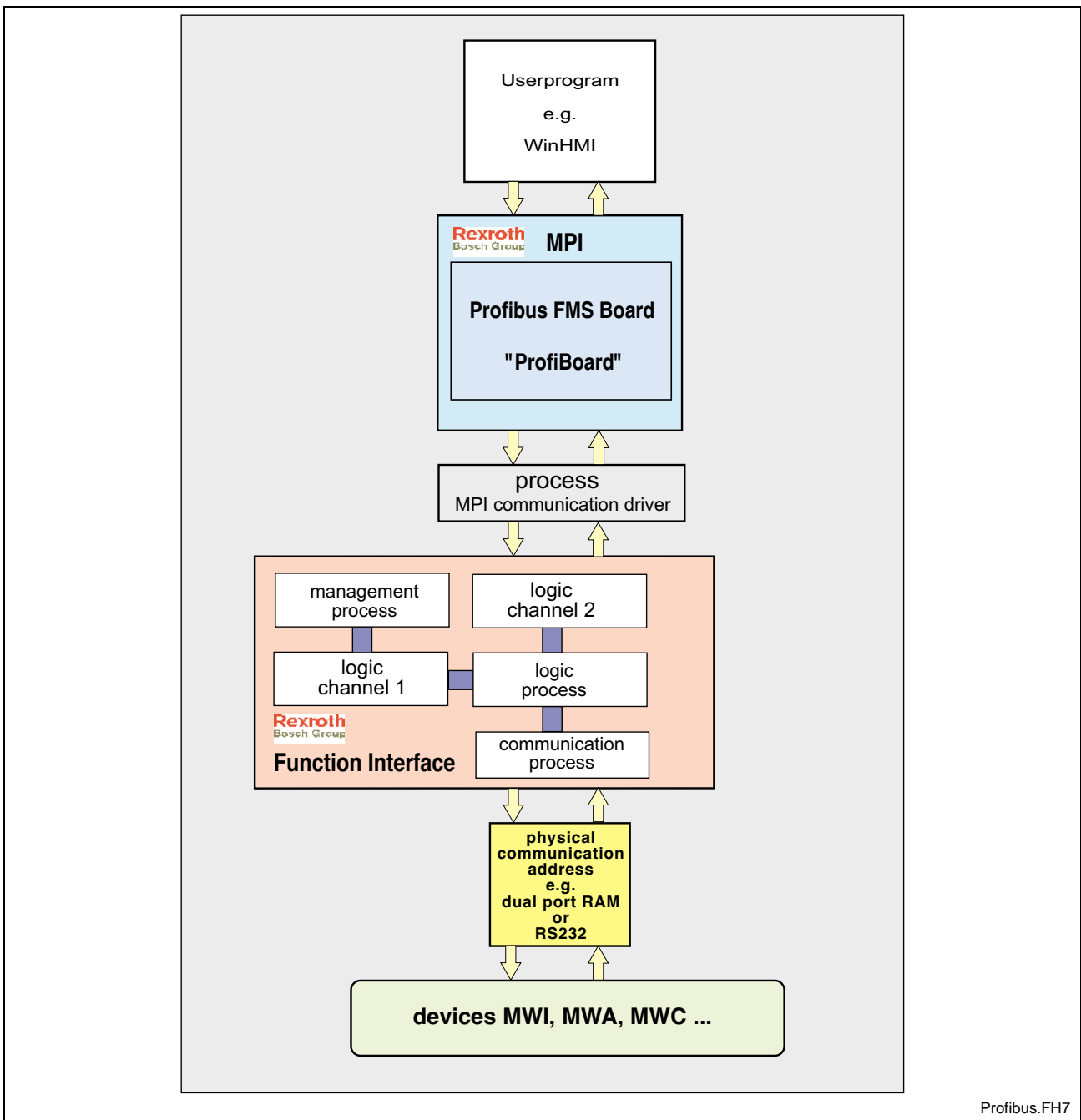


Fig. 4-2: MPI with Profibus FMS connection

Profibus.FH7

Bosch Rexroth GUI and Server

The following figure shows the software structure with the 21VRS Bosch Rexroth GUI (WinHMI) as well as when using the Bosch Rexroth DDE server. It also shows the connection of an OPC server.

The components "WinMTC" and "WinHMI" are component parts of the Bosch Rexroth GUI WIN200. The DDE server allows connection via standard communication mechanisms to external program packages such as WONDERWARE "InTouch". Furthermore, using the NetDDE option, the DDE server allows a connection to be made via a network.

OPC™ stands for **O**LE for **P**rocess **C**ontrol. OLE (**O**bject **L**inking and **E**mbedding) was originally introduced by Microsoft for communication between software components. Today, we refer to the terms COM (**C**omponent **O**bject **M**odel) or DCOM.

The goal of OPC is to create a unified communication interface for process data from any sources such as PLC and NC controls.

The user (developer of OPC client programs) therefore has the following advantages:

- Only minimum knowledge of the controls is required in order to communicate with the control software.
- No adjustment has to be made if an application has to communicate with different makes of control.

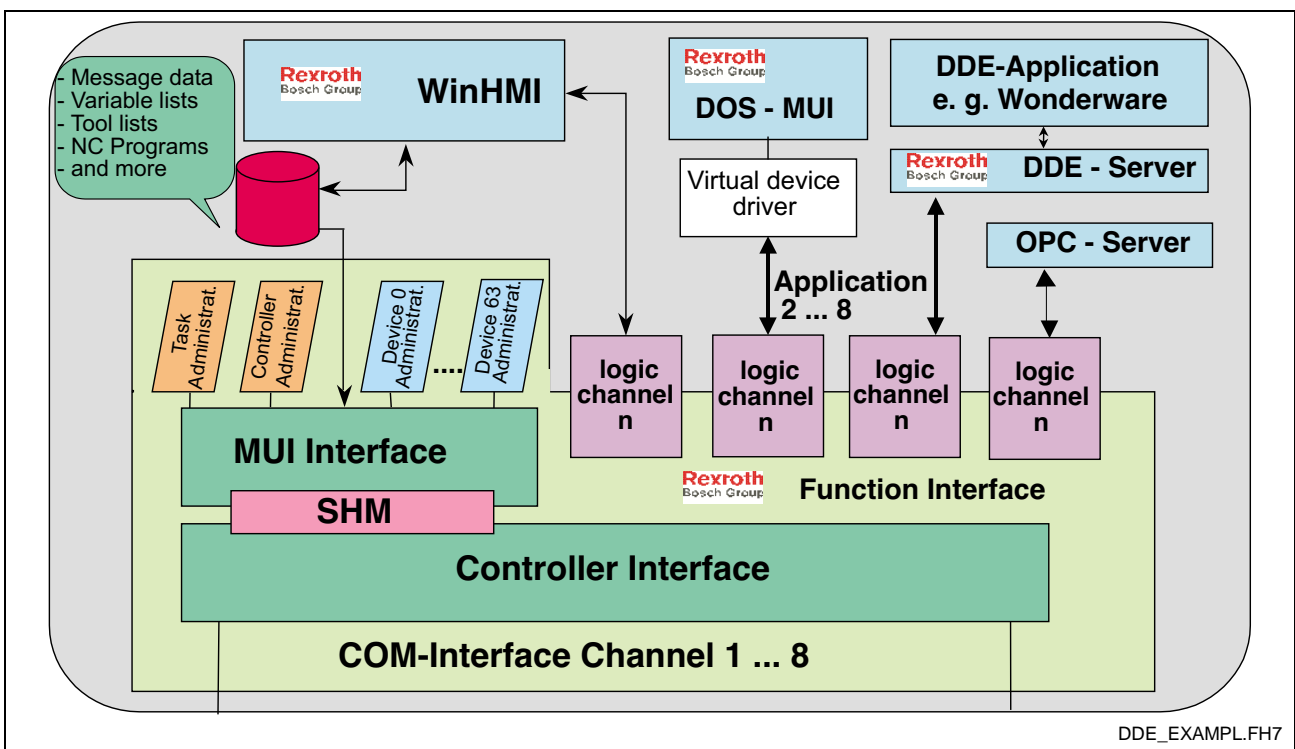


Fig. 4-3: Software structure: Bosch Rexroth user interface and DDE server

Connection to the Function Interface

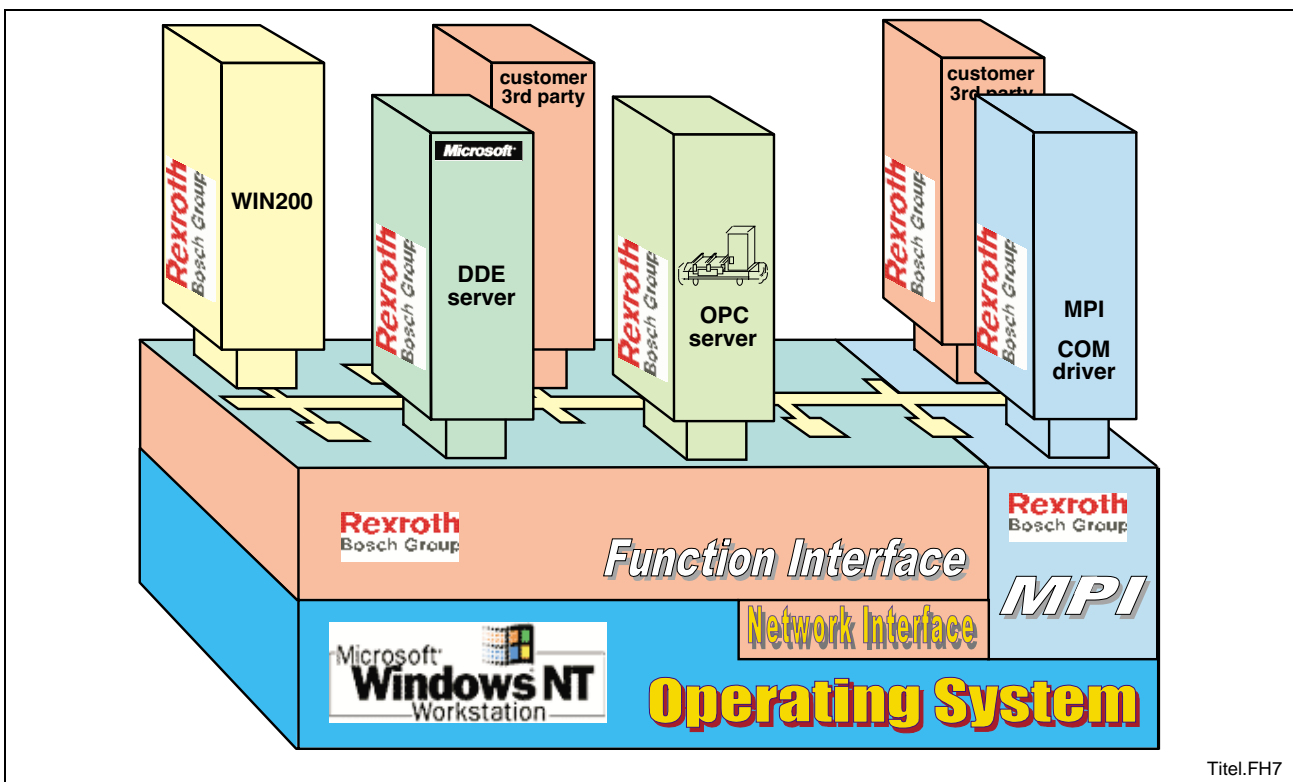
The following illustration shows the various options for connecting an application to the function interface.

Direct connection can be achieved via:

- Bosch Rexroth's own GUI, WinHMI.
- programs written by the user in Visual C++ or Visual Basic (customer 3rd party).

The following are examples of indirect connection:

- DDE server,
- OPC server, and
- MPI Com Driver.



Titel.FH7

Fig. 4-4: Overview of the connection options

Communication Between a Client and Bosch Rexroth Devices

The following figure shows the process of communication of a client on a Bosch Rexroth PC (BTV30) with two Bosch Rexroth devices (MTC200-R-G2 and MTC200-P-G2). On the one hand, the device 00 (MTC200-R-G2) communicates with the communication process via the serial interface (COM1), while device 01 (MTC200-P) communicates via a dual port RAM. The communication process opens a thread for each communication channel that has been configured. The client shown can access data from both devices. To do this, the appropriate device address is specified in the function interface command (FI command) (see Chapter "Design and Availability of the FI Command").

Note: Several cyclic requests (FI commands) can easily be combined at both devices (See chapter 5.3 "Data Transfer and Result Evaluation Routines").

During the initialization phase of the function interface, the configuration data of Bosch Rexroth devices is compared to the actual status. FI commands that have been requested are thereby checked as to their validity for the configured device group. Any errors in command mnemonics can then already be intercepted at the top level.

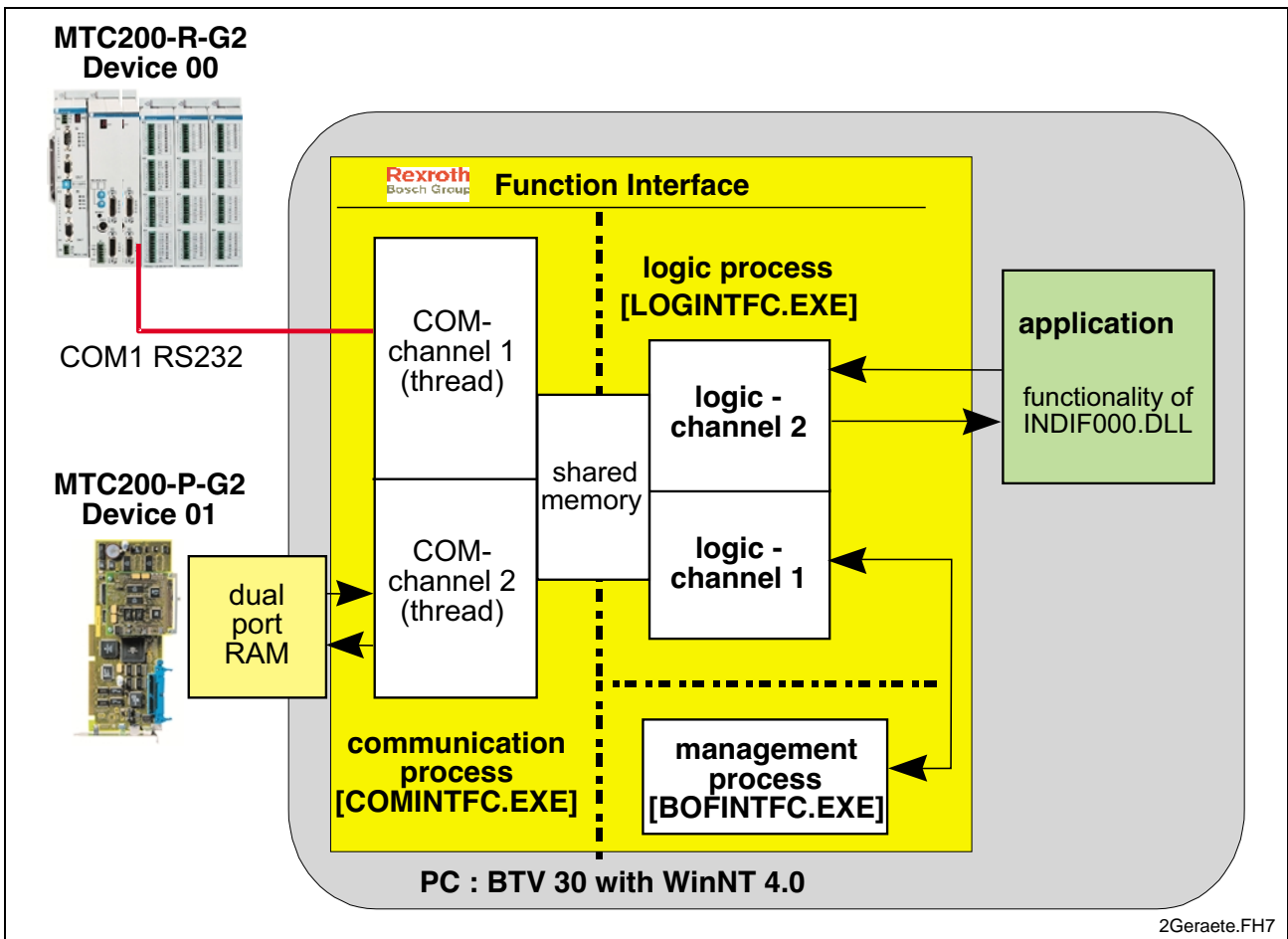


Fig. 4-5: Communication between a Client and Bosch Rexroth Devices

Communication Between Several Clients and Bosch Rexroth Devices

The following figure shows the software structure of the function interface when communicating with several devices during the operation of several clients.

Note: Combining the decentralized MTC200-R-G2 with the integrated MTC200-P-G2 is a practical configuration, for example, for a rotary transfer machine.

Here, the function interface allows parallel communication via various interfaces. In the following example, four programs are connected to the function interface in the direction of the clients. Every client can communicate with every device, independently of the other clients. When operating with several devices and several clients, the function interface works like a two-stage, buffered multiplexer. The communication process comprises a multiplexer in the direction of the device and the logic process comprises a multiplexer in the direction of the clients.

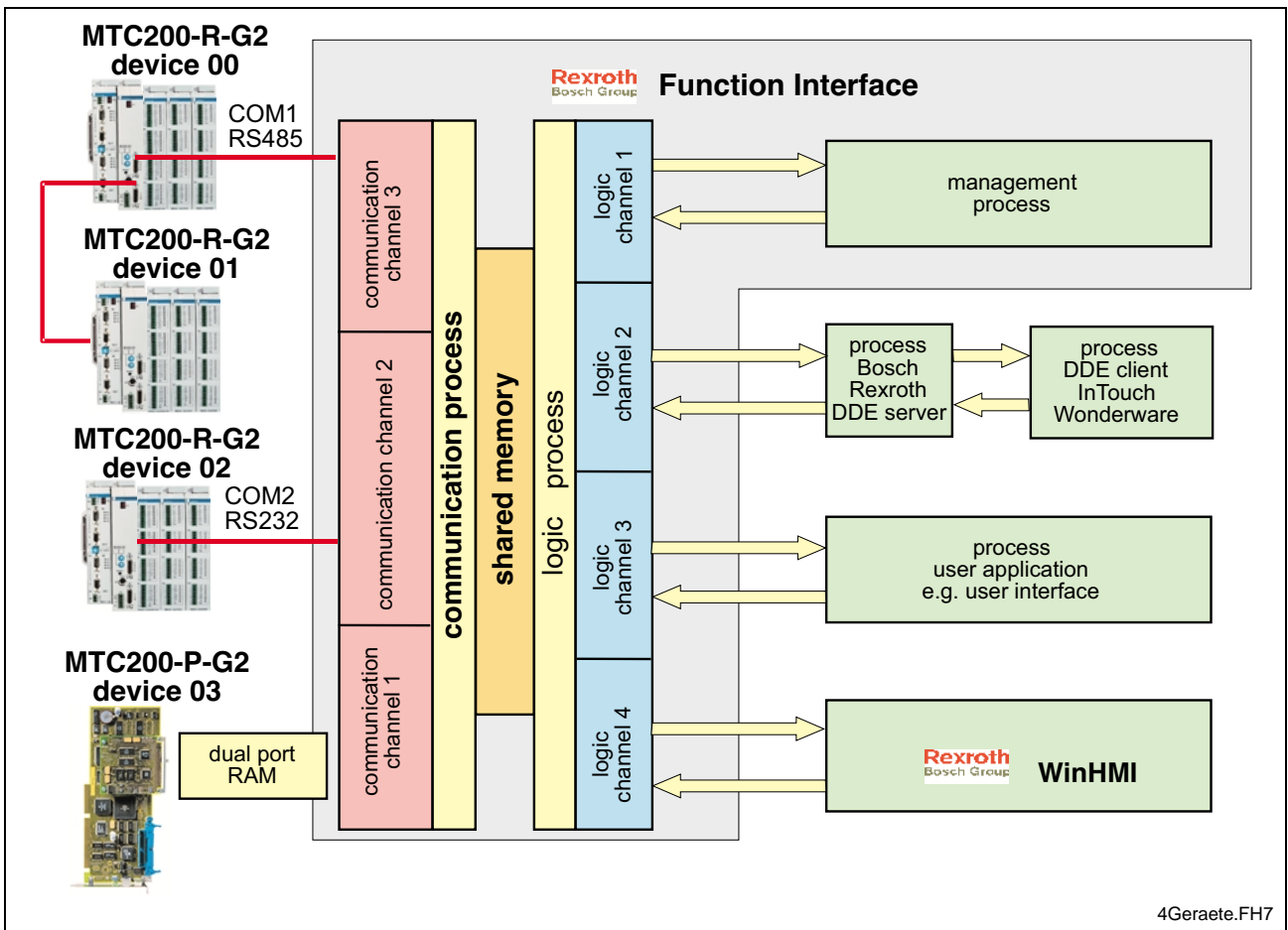


Fig. 4-6: Communication between several Client and Bosch Rexroth Devices

5 Programming

5.1 Guidelines

All user software (clients) that wants to access the function interface must be created in one of the following program languages:

- Visual C/C++ (32 bit version), or
- Visual Basic Version 5.0 and above.

The following should be observed when programming:

- the computer should be a Pentium Processor running at a min. of 200 MHz and with a RAM of at least 64 MB.

Note: Parts of the Bosch Rexroth function interface require the highest priority as a Windows NT process.

- Absolute paths should be avoided in the application as any later changes in the drive path (e.g. from C:\ to D:\) or in the directory structure are not supported.

Note: The system directory as well as the Windows NT disk drive can also be freely selected.

The following conditions and statuses of the controls or devices must be considered when programming:

- During a PLC program and/or parameter download from the Bosch Rexroth GUI, other applications must not read or write control data. The system messages (SYS-MSGs) from the call interface are used in evaluating this status. The system messages for the PLC program and/or parameter download are to be considered in the logic of the client.
- Reading and writing of PLC data is limited. Using the FI command "PVS" (see Chapter "Function Interface Commands"), PLC variables with a maximum length of 240 bytes can be read and written. PLC structures and arrays can have a dynamic length. Extremely precise planning is required for communication with the PLC.
- In principle, any PLC variable can be written using the function interface. However, only those PLC variables that are also found in the PLC program should be written in the application.

Note. Write-access to non-declared PLC variables should be avoided.

- Signals from the process/axis interface should never be directly changed by the application. Use a read/write buffer in the PLC.

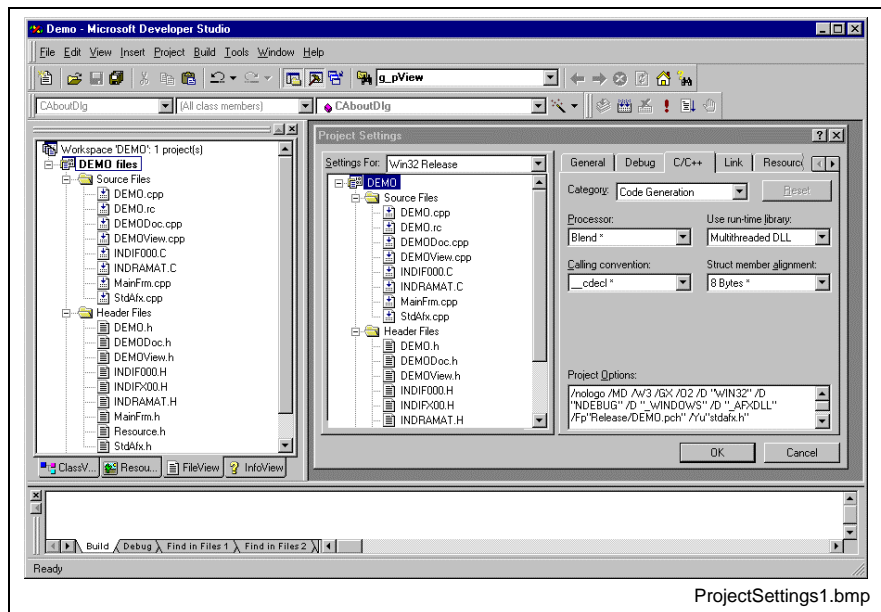


Fig. 5-2: Project settings "For Win32 Release": Multithreaded DLL

In project settings "For Win32 Debug", select the "Debug Multithreaded DLL" entry in the "Use run-time library" combo box under the category "Code-Generation" in the "C/C++" tab page.

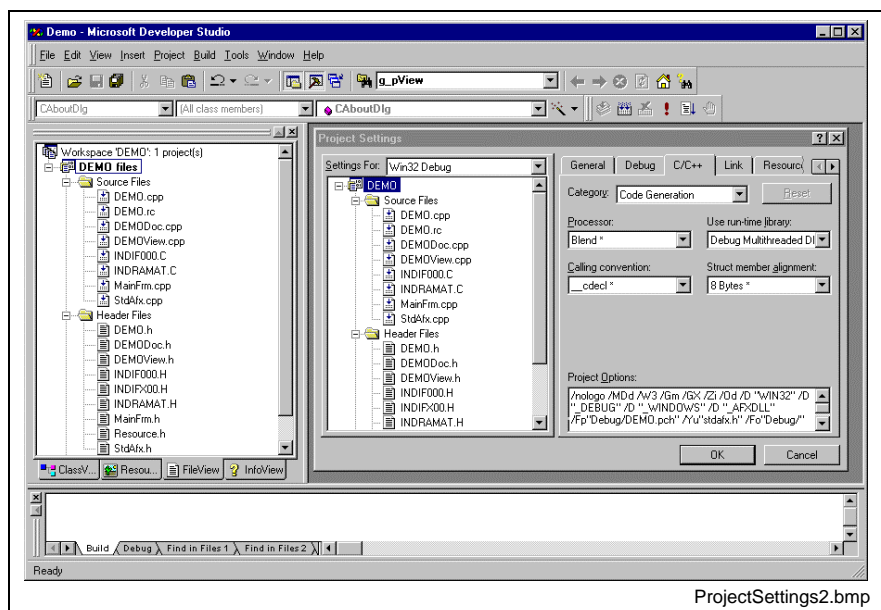


Fig. 5-3: Project settings "For Win32 Debug": Debug Multithreaded DLL

In addition, select the setting "Not using precompiled Headers" in the "For All Configurations" project settings in the "C/C++" tab page under the category "Precompiled Headers" for the following C sources:

- INDIF000.C and
- INDRAMAT.C.

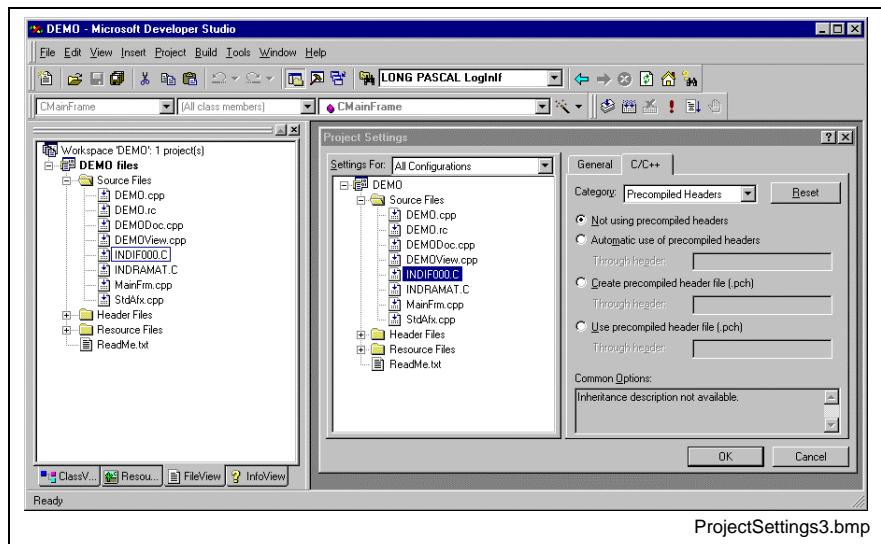


Fig. 5-4: "For All Configurations" project settings

5.2 Routines for Logging in and Logging Out

Before being able to use the access functions described in the following chapter, the login routine "LogInIf" must always be called up first. Once work with the function interface has been completed, then the logout routine "LogOutIf" should be called.

"LogInIf" Login Routine

Explanation A client connects to the management structure of the function interface via the "LogInIf" routine.

Syntax **LONG PASCAL LogInIf (** CHAR *IpcTaskName,
 CHAR *IpcCommandLine,
 CHAR *IpcParentWinName,
 HANDLE *IhTerminateEvent,
 UCHAR IuclfChannel,
 UCHAR IuclfChannelGrp,
 HANDLE *IhSysMsgEvent,
 UCHAR *IucTaskId,
 DWORD *IdwIFChannelId);

Pass Parameters

Parameters	Explanation
[IN] IpcTaskName	Pointer to the name of the client
[IN] IpcCommandLine	Pointer to the command row for the management and logic process. As a rule, the pass parameters are passed on to the client here. By this means the function interface can be switched to diagnostics mode via the command row of the client.
[IN] IpcParentWinName	Pointer to the name of the parent window of the process. Max. length = MAX_PARENT_WIN_NAME_LEN. (See file "INDIF000.H" or "INDIF000.BAS") NULL = no parent window (normal case)

Parameters	Explanation
[OUT] lhTerminateEvent	HANDLE to the termination event of a process.
[IN] luclfChannel	Decides whether or not the process requests a LOG channel 0= no LOG channel request >0 = LOG channel is requested (normal case).
[IN] luclfChannelGrp	Maximum number of function calls within a group request [1...MAXGRP]. Default 10 (refer to entries in the "INDIFX00.H" file)
[OUT] lhSysMsgEvent	HANDLE on the SYS-Msg-Event.
[OUT] lucTaskId	TaskID, that is assigned to a client on logging in for management reasons [1..MAX_TASK_ANZAHL] (see entries in the "INDIFX00.H" file).
[OUT] ldwIFChannelId	Assigned ID of the Communication Channel [2 to 8]

Return Values 0: Request successful.
1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter "General Error Result Lines".

LogInIf - Example (Visual Basic: VBDEMO.FRM)

```
Private Sub Form_Load()

'INPUT-Values of the LogInIf-routine
'*****
Dim TaskName As String           'Application's name
Dim CommandLine As String       'Command for starting conditions, e.g. "/C=t /B=w"
Dim ParentWinName As String     'Titlebar's (Window's )name
Dim IfChannel As Byte           'Function Interface Channel
Dim IfChannelGrp As Byte        'Value for group request
Dim ResBuf As String * 32768    'Resultbuffer

'Return-Values of the LogInIf-routine
'*****
Dim TaskId As Byte
Dim IfChannelId As Long

'General declarations
'*****
Dim lRet As Long                'Routine's returnvalue
Dim ErrMsg As String           'Error message string
Dim nHookList(0 To 4) As Integer 'Number of FI-System Messages (FI-SYS-MSGs)
Dim lpThreadId As Long

'Timer interval initialisation
'*****
TimerInterval.Caption = CyclicOutputTimer.Interval
CycleTime.Value = CyclicOutputTimer.Interval

TaskName = "VBDemo.exe"        'Application's name
CommandLine = Command         'Command for starting conditions, e.g. /C=t /B=w
ParentWinName = "VBDemo"      'Titlebar's (Window's )name
IfChannel = 1                  'Function-Interface Channel 1 requested
IfChannelGrp = 10             'Max. value for group request
lRet = 1                       'Default Returnvalue = 1 for error handling

'Call LogInIf-Routine (Start Interface)
'*****
```

```

lRet = LogInIf(TaskName, CommandLine, ParentWinName, SysThread.hTerminateEvent,
IfChannel, IfChannelGrp, SysThread.hSysMsgEvent, TaskId, IfChannelId)

'Error handling & Function-interface channel identification output
'*****
If lRet Then 'error handling
    VBDemoStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
    ErrMsg = "LogIn Error code: " + CStr(lRet)
    VBDemoStatus.Caption = ErrMsg
Else 'Function-interface channel identification output
    VBDemoStatus.BackColor = QBColor(10) 'set BackgroundColor to bright green
    VBDemoStatus.Caption = "Login succeeded on FI-Channel " & IfChannelId
End If

'Creating Function-Interface-System-Message-List (FI-SysMsg)
'*****
nHookList(0) = 4 'Number of FI-SYS-MSGs
nHookList(1) = MSG_PCLUPDBEG 'PLC Download Begin
nHookList(2) = MSG_PCLUPDEND 'PLC Download End
nHookList(3) = MSG_PARUPDBEG 'Parameter Download Begin
nHookList(4) = MSG_PARUPDEND 'Parameter Download End

lRet = HookIfMsgList(nHookList(0)) 'Ptr-Handed over in Basic is equal to C

If lRet Then 'error handling
    SYS_Messages.BackColor = QBColor(12) 'set BackgroundColor to bright red
    ErrMsg = "HookIfMsgList terminated with error code: " + CStr(lRet)
    SYS_Messages.Caption = ErrMsg
End If

'Starting FI-SYS-Msg Thread
'*****
hThread = CreateThread(0, _
    0, _
    AddressOf SysThread.SysMsgThreadProc, _
    0, _
    0, _
    lpThreadId)

If hThread = 0 Then 'error handling
    SYS_Messages.BackColor = QBColor(12) 'set BackgroundColor to bright red
    ErrMsg = "Thread couldn't be created" & Err.LastDllError
    SYS_Messages.Caption = ErrMsg
End If

'Process verification for the Function-Interface
'*****
lRet = DataTransfer("XX_BW_RPR1", 0, 0, 1, ResBuf, 32768, 1)

End Sub

```

LogInIf - Example (Visual C++)

```

// General Declarations
//*****
LONG lRet;
CHAR acErrMsg[80];
// Input parameters of the LogInIf routine
//*****
HANDLE ghTerminateEv = ZERO;
HANDLE ghSysMsgEv = ZERO;
UCHAR gucTaskId = 0;
DWORD gdwlFChannelId = 0;
// LogInIf routine (Start Interface)
//*****
lRet = LogInIf("VCDemo.exe", // Name of user program,
m_lpCmdLine, // Command, e.g. "/C=t",
"Demo", // Window's Name,
&ghTerminateEv, // HANDLE on TerminateEvent,
1, // Interface channel requested,
10, // Max. number of function requests in group,
&ghSysMsgEv, // HANDLE on SYS-Msg-Event,
&gucTaskId, // Task-ID,
&gdwlFChannelId); // Communication channel - ID

```

```
// Error Handling
//*****
if (lRet)
{
    sprintf(acErrMsg,"Function-Interface LogInIf ErrorCode:%ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
}
```

"LogOutIf" Log out Routine

Explanation	A client logs out from the management structure of the function interface via the "LogOutIf" routine.
Syntax	LONG PASCAL LogOutIf ();
Pass Parameters	None
Return Values	0: Request successful. 1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter "General Error Result Lines".

LogOutIf - Example (Visual Basic: VBDEMO.FRM)

```
Public Sub Form_Terminate()
'IN-/Output Values
'*****
Dim lRet As Long           'Routine's returnvalue
Dim ErrMsg As String      'Error message string

'Closing Function-Interface Channel
'*****
lRet = LogOutIf()         'Stop Function-Interface

If lRet Then 'error handling
    VBDemoStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
    ErrMsg = "LogOut termination with error code: " + CStr(lRet)
    VBDemoStatus.Caption = ErrMsg
End If

CloseHandle (hThread)    'Thread clearance

End Sub
```

LogOutIf - Example (Visual C++)

```
// General Declarations
//*****
LONG lRet;
CHAR acErrMsg[80];
//LogOutIf-Routine (Stop Interface)
//*****
lRet = LogOutIf();
// Error Handling
//*****
if (lRet)
{
    sprintf(acErrMsg,"Function-Interface LogOutIf ErrorCode:
```

5.3 Data Transfer and Result Evaluation Routines

The "DataTransfer" routine is used for reading and writing data to and from Bosch Rexroth devices. The data delivered in the result buffer is structured as follows:

Single Request

Group Element 1	Line 1	Column 1	...	Column j
	:	:	:	:
	Line m	Column 1	...	Column j

Notes: In case of an error, (return value $\neq 0$), the result buffer contains a general error result line that may have to be evaluated in a separate routine (see Chapter 7 "Error Codes").

As only one command row and **no** group of command rows (also called group request) can be processed via the "DataTransfer" routine, it only has one results group. The data in the result buffer can be evaluated using the "ReadGroupItem" routine.

The "GetNumberOfGroups" returns the number of groups in the result. The "GetNumberOfRows" routine determines the number of lines (rows) for a group, and the "GetNumberOfItems" routine determines the number of columns in the rows.

Group request (ONLY for cyclic requests)

During a group request, several command rows (single requests) are requested simultaneously. The command rows of a group request are separated by spaces. Exactly one group element is delivered in the result for each of these command rows in a group request. The data returned in the result buffer is therefore structured as follows:

Group Element 1	Line 1	Column 1	...	Column j
	:	:	:	:
	Line i	Column 1	...	Column j
:	:	:	:	:
Group Element n	Line 1	Column 1	...	Column j
	:	:	:	:
	Line i	Column 1	...	Column j

Example of a group request

During a group request (BR_NPS... BR_ABN... BR_AGF...), the single group elements can be accessed with $[bGroup]$. The meaning of the elements is as follows:

- Group Element ($bGroup = 1$): BR_NPS...
- Group Element ($bGroup = 2$): BR_ABN...
- Group Element ($bGroup = 3$): BR_AGF...

Note: A maximum of 256 command rows (FI commands) can be abstracted as a group request.

"DataTransfer" Routine

Explanation Data is read or written in accordance to the configured functions using the "DataTransfer" routine (see Function Interface 08VRS - Reference, Chapter "Function Interface Commands").

Syntax **LONG PASCAL SetIfMsgConf (** **CHAR *pszFunction,**
CHAR acValue[],
LONG ValLen,
LONG ValType,
CHAR acResBuf[],
LONG IMaxResLen,
LONG IResBufType);

Pass Parameters

Parameters	Explanation
[IN] pszFunction	Command row
[IN] acValue	Value to be written
[IN] ValLen	Length of value to be written
[IN] ValType	Data code of the value to be written (see chapter "Design and Availability of the FI Command", Data Code)
[OUT] acResBuf	Result buffer
[IN] IMaxResLen	Length of the result buffer depending on the requested data. The RESULT_BUF_SIZE constant from the INDIF000.h file can be taken as the guide value.
[IN] IResBufType	Data code of results data (see Chapter "Design and Availability of the FI Command", Data code)

Note: The data delivered in the result buffer is coded. To access the single elements, the content of the result buffer must be processed using the "ReadGroupItem" routine:

Return Values 0: Request successful.
 1 ...n: Request unsuccessful (see chapter 7 "Error Codes").

Note: Additionally, an error can be queried in more detail with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter 7.1 "General Error Result Line".

DataTransfer - Example (Visual Basic: VBDEMO.FRM)

```

Private Sub DataTransferFunc ()

'Read/Write Data from/to the various devices via the function-interface
'*****
Dim ResBuf As String * 32768      'Resultbuffer
Dim lRet As Long                 'Routine's returnvalue
Dim lLen As Long                 'Value's length
Dim pszFunction As String        'FI-command
Dim ErrMsg As String            'Error message string
Dim szBuf As String * 32768     'Buffer for controller data
Dim DataValidation As Boolean    'Flag for data validation
Dim szVal As String             'Writevalue

pszFunction = SingleRequest.Text 'Hand over FI-command from Editbox
szVal = WriteValue.Text         'Hand over WriteValue from Editbox

'DataTransfer to function-interface
'*****
lRet = DataTransfer(pszFunction, szVal, Len(szVal), 1, ResBuf, 32768, 1)

If lRet Then 'error handling
    ErrMsg = "DataTransfer terminated with error code: " + CStr(lRet)
    SingleRequestStatus.Caption = ErrMsg
    SingleRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
    lRet = ReadGroupItem(ResBuf, 1, -1, -1, szBuf, 32768, lLen, DataValidation)
    Output.Text = szBuf
Else 'Valid reply
    SingleRequestStatus.BackColor = QBColor(10) 'set BackgroundColor to bright green
    SingleRequestStatus.Caption = "DataTransfer command was succesfully completed"
    lRet = ReadGroupItem(ResBuf, 1, -1, -1, szBuf, 32768, lLen, DataValidation)
    Output.Text = szBuf
End If
End Sub

```

DataTransfer - Example (Visual C++)

```

// General Declarations
//*****
LONG lRet;
CHAR acErrMsg[80];
int i,j;

// Starting Parameters of the DataTransfer-Routine
//*****
CHAR * szValue = "";
CHAR * szFunction = "02_CR_CCP4";
CHAR acResultbuf[RESULT_BUF_SIZE];

// Starting Parameters of the GetNumberOfRows-Routine
//*****
LONG lNumOfRows;

// Starting Parameters of the GetNumberOfItems-Routine
//*****
LONG lNumOfItems;

// Starting Parameters of the ReadGroupItem-Routine
//*****
LONG lItemLen;
CHAR acItembuf[50];
BOOL boItemValid;

// Access to Function Interface
//*****

lRet = DataTransfer (szFunction, // Command row,
szValue, // Value,
strlen(szValue), // Length of value,
1, // Data code of value,
acResultbuf, // Result buffer,
RESULT_BUF_SIZE, // Length of result buffer,
1); // Data code of result data

// Error Handling

```

```

if (lRet)
{
    sprintf(acErrMsg,"Function-Interface DataTransfer ErrorCode:%ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}

// Get number of rowa
//*****
lRet = GetNumberOfRows(acResultbuf, // Result data,
1, // Group element,
&lNumOfRows); // Number of rows

// Error Handling
if (lRet)
{
    sprintf(acErrMsg,"Function-Interface GetNumberOfRows ErrorCode:%ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}

// Result Evaluation
//*****
for (i=1; i<=lNumOfRows; i++)
{
    // Determine Number of Elements
    //*****
    lRet = GetNumberOfItems(acResultbuf,
1, // Group element,
i, // Row,
&lNumOfItems); // Number of elements ?

    // Error Handling
    if (lRet)
    {
        sprintf(acErrMsg,"Function-Interface GetNumberOfItems ErrorCode:%ld ",lRet);
        MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
    }

    // Evaluate all Lines
    //*****
    for (j=1; j<=lNumOfItems; j++)
    {
        // Evaluate all Results of a Line
        lRet=ReadGroupItem(acResultbuf, // Result data,
1, // Group element,
i, // Row,
j, // Column,
acItembuf, // Individual result,
50, // Length of individual result buffer,
&lItemLen, // Length of result,
&boItemValid); // Valid value ?

        // Error Handling
        if (lRet)
        {
            sprintf(acErrMsg,"Function-Interface ReadGroupItem ErrorCode:%ld ",lRet);
            MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
        }
    }
}
}

```

"ReadGroupItem" Routine

Explanation This routine allows a single result, an entire row or a table of a single or group request to be read out. All results data must be evaluated using this routine.

Syntax **LONG PASCAL** ReadGroupItem (**CHAR acResBuf[]**
BYTE bGroup,
LONG IRow,
LONG IItem,
CHAR acItemBuf[],
LONG IItemBufLen,
LONG *pIItemLen,
BOOL *pbolItemValid);

Pass Parameters

Parameters	Explanation and Value Range
[IN] CHAR acResBuf[]	Buffer for the entire result
[IN] BYTE bGroup	Details of group element [1 to n]
[IN] LONG IRow	-1: Output of a complete table, i.e. all rows of a request [1 to n]: the respective result line
[IN] LONG IItem	-1: Output of a row 0: Output of the requested command with management information [1...n]: Individual result (Element of a row)
[OUT] CHAR acItemBuf[]	Buffer for requested partial result
[IN] LONG IItemBufLen	Length of buffer for partial result
[OUT] LONG *pIItemLen	Length of partial result
[OUT] BOOL *pbolItemValid	TRUE: with valid value of the partial result

Return Values 0: Request successful.
 1 ...n: Request unsuccessful (see chapter "Error Codes").

Example of "ReadGroupItem" Routine

The following example assumes that a single request (*bGroup* = 1) has been requested:

Line	Column 1	Column 2	Column 3	Column 4
1	E1	E2	E3	
2	E4	E5		
3	E6	E7	E8	E9

Example of Syntax	Result
ReadGroupItem(acResBuf, 1, 1, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	E1
ReadGroupItem(acResBuf, 1, 2, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	E2
ReadGroupItem(acResBuf, 2, 2, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	E5
ReadGroupItem(acResBuf, 3, 4, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	E9
ReadGroupItem(acResBuf, 2, 3, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	Error code
ReadGroupItem(acResBuf, 1, -1, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	E1 E2 E3
ReadGroupItem(acResBuf, 2, -1, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	E4 E5
ReadGroupItem(acResBuf, 3, -1, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	E6 E7 E8 E9
ReadGroupItem(acResBuf, -1, -1, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	E1 E2 E3 E4 E5 E6 E7 E8 E9
ReadGroupItem(acResBuf, 1, 0, 1, acItemBuf, lLenBuf, &ItemLen, &boltemValid)	supplies e.g.: 001234567800_CC_

Example of Visual Basic/ C++ (see "DataTransfer" routine)

"GetNumberOfGroups" Routine

Explanation The "GetNumberOfGroups" routine returns the number of group elements.

Syntax LONG PASCAL ReadGroupItem (CHAR *pszValBuf, LONG *plGroupSize);

Pass Parameters

Parameters	Explanation
[IN] CHAR *pszValBuf	Buffer for the entire result
[OUT] LONG *plGroupSize	Number of group elements

Return Values

0: Request successful.
1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried in more detail with the "ReadGroupItem" routine in the form of a general error result line. For more detailed information, please refer to the chapter "Error Code".

Example of Visual Basic/ C++ (see "DataTransfer" routine)

"GetNumberOfRows" Routine

Explanation The "GetNumberOfRows" routine determines the number of rows of the indicated group element.

Syntax **LONG PASCAL** GetNumberOfRows (**CHAR *pszValBuf,**
BYTE bGroupIndex,
LONG *pINumberOfRow);

Pass Parameters

Parameters	Explanation
[IN] CHAR *pszValBuf	Buffer for the entire result
[IN] BYTE bGroupIndex	Number of group elements
[OUT] LONG *pINumberOfRow	Number of rows of a group element

Return Values 0: Request successful.
1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried in more detail with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter "General Error Result Lines".

Example of Visual Basic/ C++ (see "DataTransfer" routine)

"GetNumberOfItems" Routine

Explanation The "GetNumberOfItems" routine determines the number of partial results, depending on the row number as well as the number of the group element.

Syntax **LONG PASCAL** GetNumberOfItems (**CHAR *pszValBuf,**
BYTE bGroupIndex,
BYTE bRowIndex,
LONG *pINumberOfItems);

Pass Parameters

Parameters	Explanation
[IN] CHAR *pszValBuf	Buffer for the entire result
[IN] BYTE bGroupIndex	Number of group elements
[IN] BYTE bRowIndex	Row index 0: number of all partial results
[OUT] LONG pINumberOfItems	Number of partial results for a particular row.

Return Values 0: Request successful.
1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried in more detail with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter "General Error Result Lines".

Example of Visual Basic/ C++ (see "DataTransfer" routine)

5.4 Routine for Cyclical Reading via Pipes

The pipe access functions are used for cyclical reading of device data via the function interface. Several command rows can be passed simultaneously via a group request. The command rows of a group request are separated by a space (refer here also to the "ReadGroupItem" routine).

Note: A maximum of 256 command rows (FI commands) can be abstracted as a group request.

A pipe is started by the "StartCyclicPipe" routine and then provides itself continually with updated data. Asynchronous to this, access to this data is now made via the "ReadCyclicPipe" routine. The cyclical request is stopped by the "StopCyclicPipe" routine.

"StartCyclicPipe" Routine

Explanation The "StartCyclicPipe" routine starts a pipe for cyclical reading of the data.

Syntax **LONG PASCAL StartCyclicPipe (** **WORD wPipe,**
CHAR *pszFunktion,
LONG IBufSize,
LONG IGroupSize,
DWORD dwSleep);

Pass Parameters

Parameters	Explanation
[IN] wPipe	Pipe number [1...1000]
[IN] *pszFunktion	Group of strings according to the defined function requests.
[IN] IBufSize	Size of result buffer [Byte]
[IN] IGroupSize	Number of group elements [1 to n]
[IN] dwSleep	Read delay time [ms]

Return Values 0: Request successful.
 1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter "General Error Result Lines".

StartCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```

Public Sub StartCyclicFunc()
'Start of a cyclic request
'*****
Dim lRet As Long           'Routine's returnvalue
Dim ErrMsg As String      'Error message string
Dim pszFunction As String 'FI-command
pszFunction = CyclicRequest.Text 'Hand over FI-Command from Editbox
If Not CyclicRun Then 'in case of a cyclic request has NOT been started
    lRet = StartCyclicPipe(1, pszFunction, 32768, 2, 250)
    If lRet Then 'error handling
        CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
        ErrMsg = "StartCyclicPipe terminated with error code:" + CStr(lRet)
        CyclicRequestStatus.Caption = ErrMsg
        Exit Sub 'in case of an error has occurred

    End If
    CyclicRun = True 'Flag for a cyclic request is started
    CyclicOutputTimer.Enabled = True 'Timer output is started
End If
End Sub

```

StartCyclicPipe - Example (Visual C++)

```

// General Declarations
//*****
LONG lRet;
CHAR acErrMsg[80];

// Starting Parameters of the StartCyclicPipe - Routine
//*****
CHAR * szGroupFunction = "00_CC_AGF_0 00_CC_PVS_ErrorFlg";
// Open Pipe
//*****
lRet = StartCyclicPipe(wPipeNo, // Pipe - number,
szGroupFunction, // Function call group,
RESULT_BUF_SIZE, // Size result buffer,
2, // Number group elements,
500); // Reading delay time [ms]

// Error Handling
//*****
if (lRet)
{
    sprintf(acErrMsg, "Function-Interface LogInIf ErrorCode:%ld ", lRet);
    MessageBox (GetFocus(), acErrMsg, "Function Interface Error", MB_OK);
}

```

"ReadCyclicPipe" Routine

Explanation The "ReadCyclicPipe" routine reads the data of a pipe that has been started by "StartCyclicPipe".

Syntax **LONG PASCAL** ReadCyclicPipe (**WORD** wPipe,
CHAR acResult[],
LONG lBufSize,
BYTE *pbGroupFault,
LONG *plAttr);

Pass Parameters

Parameters	Explanation
[IN] WORD wPipe	Number of the pipe
[OUT] CHAR acResult[]	Buffer for the entire result
[IN] LONG lBufSize	Buffer size of the entire result
[OUT] BYTE *pbGroupFault	Number of the group element in case of error
[OUT] LONG *plAttr	Result attribute

Return Values

0: Request successful.
 1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: The pass parameter [OUT] BYTE *pbGroupFault contains the number of the faulty group element. Further information on the meaning of the "ReadCyclicPipe" routine error code can be requested in the form of a general error result line using the "ReadGroupItem" routine. For additional information refer to the chapter "General Error Result Lines".

ReadCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```
Private Sub CyclicOutputTimer_Timer()

'IN-/Output Values
'*****
Dim lRet As Long           'Routine's returnvalue
Dim ErrMsg As String      'Error message string
Dim ResultBuffer As String * 32768
Dim lNumberOfRows As Long 'Number of Rows ->
Dim i As Long             'Index for the number of rows
Dim szBuf As String * 256 'Buffer for controller data
Dim lLen As Long          'Value's lenght
Dim DataValidation As Boolean 'Flag for data validation
Dim bGroup As Byte
Dim lAttr As Long

lRet = ReadCyclicPipe(1, ResultBuffer, 32768, bGroup, lAttr)

If lRet Then 'error handling
  CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
  ErrMsg = "ReadCyclicPipe terminated with error code: " + CStr(lRet)
  CyclicRequestStatus.Caption = ErrMsg
  Exit Sub
End If
OutputList.Clear
If lRet = 0 Then
  lRet = GetNumberOfRows(ResultBuffer, 1, lNumberOfRows)
  Rows.Text = lNumberOfRows
  If lRet Then 'error handling
    CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to
bright red
    ErrMsg = "GetNumberOfRows terminated with error code: " + CStr(lRet)
    CyclicRequestStatus.Caption = ErrMsg
  End If
  For i = 1 To lNumberOfRows
    lRet = ReadGroupItem(ResultBuffer, 1, i, -1, szBuf, 32768, lLen,
DataValidation)
    If lRet Then 'error handling
      CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to
bright red
      ErrMsg = "ReadGroupItem terminated with error code: " + CStr(lRet)
      CyclicRequestStatus.Caption = ErrMsg
    End If
    OutputList.AddItem (szBuf)
  Next
  CyclicRequestStatus.BackColor = QBColor(10) 'set BackgroundColor to bright
green
  CyclicRequestStatus.Caption = "ReadCyclicPipe command was succesfully
```

```

completed"
End If
End Sub

```

ReadCyclicPipe - Example (Visual C++)

```

// General Declarations
//*****
LONG lRet;
CHAR acErrMsg[80];
int i;
// Input parameters of the ReadCyclicPipe routine
//*****
CHAR acResultbuf[RESULT_BUF_SIZE];
UCHAR bIndexItemFault;
LONG lAttr;
// Input parameters of the GetNumberOfGroups routine
//*****
LONG lNumOfGroups;
// Read pipe
//*****
lRet = ReadCyclicPipe(wPipeNo,           // Pipe number,
acResultbuf,                          // Result buffer,
RESULT_BUF_SIZE,                      // Length result buffer,
&bIndexItemFault,                    // Index of the group
                                       // element with error,
                                       // result attribute
&lAttr);
// Error handling
if (lRet)
{
    sprintf(acErrMsg,"Function-Interface ReadCyclicPipe ErrorCode: %ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
// Determine number of groups
//*****
lRet = GetNumberOfGroups(acResultbuf,    // Result buffer,
&lNumOfGroups); // Number of groups,
// Error handling
if (lRet)
{
    sprintf(acErrMsg,"Function interface GetNumberOfGroups ErrorCode: %ld",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
// Evaluation of results
//*****
for (i=1; i<=lNumOfGroups; i++)
{
    // Results evaluation for each group result
    // e.g. LONG lItemLen;
    //     CHAR acItembuf[50];
    //     int iItemValid;
    //
    lRet=ReadGroupItem(acResultbuf,      // Result buffer,
i,                                     // Group element,
1,                                     // Line,
1,                                     // Element,
acItembuf,                            // Individual result buffer,
50,                                    // Length of the individual result buffer,
&lItemLen,                             // Length of the individual result,
&iItemValid);                          // Individual result valid?
// Error handling
if (lRet)
{
    sprintf(acErrMsg,"Function-Interface ReadGroupItem ErrorCode:%ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
}

```

"StopCyclicPipe" Routine

Explanation The "StopCyclicPipe" routine stops the data request of a pipe that has been started by "StartCyclicPipe".

Syntax **LONG PASCAL StopCyclicPipe (WORD wPipe);**

Pass Parameters

Parameters	Explanation
[IN] WORD wPipe	Pipe number

Return Values

0: Request successful.
1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter "General Error Result Lines".

StopCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```
Public Sub StopCyclicFunc()
'Stop of a cyclic request
'*****

'IN-/Output Values
'*****
Dim lRet As Long           'Routine's returnvalue
Dim ErrMsg As String      'Error message string

'Cyclic request termination
'*****
If CyclicRun Then 'in case of a cyclic request has been started
    CyclicOutputTimer.Enabled = False      'Timer output is stopped
    lRet = StopCyclicPipe(1)

    If lRet Then 'error handling
        CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
        ErrMsg = "StopCyclicPipe terminated with error code:" + CStr(lRet)
        CyclicRequestStatus.Caption = ErrMsg
    End If

    CyclicRun = False 'Flag for a cyclic request is stopped

End If
End Sub
```

StopCyclicPipe - Example (Visual C++)

```
// General Declarations
//*****
LONG lRet;
CHAR acErrMsg[80];

// Close Pipe
//*****
lRet = StopCyclicPipe(wPipeNo);           // Pipe number
// Error handling
//*****

if (lRet)
{
    sprintf(acErrMsg, "Function-Interface ErrorCode:%ld ", lRet);
    MessageBox (GetFocus(), acErrMsg, "Function Interface Error", MB_OK);
}
```

"SuspendCyclicPipe" Routine

Explanation The "SuspendCyclicPipe" routine sets the data request of a pipe that has been started by "StartCyclicPipe" into standby mode. It is used to stop communication while at the same time maintaining the management structure of the function interface established by the "StartCyclicPipe" routine (see "ResumeCyclicPipe" routine).

Syntax **LONG PASCAL SuspendCyclicPipe (WORD wPipe);**

Pass Parameters

Parameters	Explanation
[IN] WORD wPipe	Number of the pipe

Return Values

0: Request successful.
1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter "General Error Result Lines".

SuspendCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```
Public Sub SuspendCyclicFunc()
'Stand-by-Modus for a cyclic request
'*****

'IN-/Output Values
'*****
Dim lRet As Long           'Routine's returnvalue
Dim ErrMsg As String      'Error message string

If CyclicRun Then 'in case of a cyclic request has been started
    CyclicOutputTimer.Enabled = False      'Timer output is stopped
    lRet = SuspendCyclicPipe(1)

    If lRet Then 'error handling
        CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
        ErrMsg = "SuspendCyclicPipe terminated with error code: " + CStr(lRet)
        CyclicRequestStatus.Caption = ErrMsg
    End If

End If
End Sub
```

SuspendCyclicPipe - Example (Visual C++)

```
// General Declarations
//*****
LONG lRet;
CHAR acErrMsg[80];

// Suspend Pipe
//*****
lRet = SuspendCyclicPipe(wPipeNo); // Pipe number

// Error handling
//*****
if (lRet)
{
    sprintf(acErrMsg,"Function-Interface SuspendCyclicPipe ErrorCode:%ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
}
```


"ResumeCyclicPipe" Routine

Explanation The "ResumeCyclicPipe" routine reactivates the data request of a pipe that has been set to standby mode by the "SuspendCyclicPipe" routine.

Syntax LONG PASCAL SuspendCyclicPipe (WORD wPipe);

Pass Parameters	Parameters	Explanation
	[IN] WORD wPipe	Number of the pipe

Return Values 0: Request successful.
1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried with the "ReadGroupItem" routine in the form of a general error result line. For additional information refer to the chapter "General Error Result Lines".

ResumeCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```
Public Sub ResumeCyclicFunc()
'Activates a suspended cyclic Pipe
'*****

'IN-/Output Values
'*****
Dim lRet As Long           'Routine's returnvalue
Dim ErrMsg As String      'Error message string

'Cyclic request termination
'*****
If CyclicRun Then 'in case of a cyclic request has been started
    CyclicOutputTimer.Enabled = True           'Timer output is started
    lRet = ResumeCyclicPipe(1)

    If lRet Then 'error handling
        CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
        ErrMsg = "ResumeCyclicPipe terminated with error code: " + CStr(lRet)
        CyclicRequestStatus.Caption = ErrMsg
    End If

End If
End Sub
```

ResumeCyclicPipe - Example (Visual C++)

```
// General Declarations
//*****
LONG lRet;
CHAR acErrMsg[80];

// Resume Pipe
//*****

lRet = ResumeCyclicPipe(wPipeNo);           // Pipe number
// Error Handling
//*****

if (lRet)
{
    sprintf(acErrMsg,"Function-Interface ResumeCyclicPipe ErrorCode:%ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
```

5.5 System Messages in the Network

SYS-Messages are used to report particular events in the system to the applications. Generally speaking the application is synchronized to the changed control system data.

Examples of this are PLC program download, parameter download, system error.

Example PLC program download

Before the PLC GUI downloads a PLC program the SysMessage MSG_PCLUPDBEG is reported.

The PLC GUI reports the end of the download with the SYS-Message MSG_PCLUPDEND.

After that, the applications will re-synchronize access to the PLC data.

PLC user interface	SYS Message	Application status
Translate program		PLC available (1)
	→ MSG_PCLUPDBEG →	PLC available
		PLC blocked
	← MSG_PCLUPDBEG_Q ←	PLC blocked
Download		PLC blocked
	→ MSG_PCLUPDEND →	PLC blocked
		PLC blocked
	← MSG_PCLUPDEND_Q ←	PLC available
!! Download ended!!		PLC available

Fig. 5-5: Synchronization via SYS-Messages

- Transparent across the network.
- Messages logged in to a particular device.
- Multiple logins in one application to support different process packages running in a process.
- Dynamically expandable message type for messages relating to the application such as for updating global network data.
- Failure of the network connection: The failure of the connection to a device is acknowledged for each requested message.

Data Types

TyVMsgCallback

The following callback has been defined for receiving network messages:

```

Declaration
LONG          VOID          *pParam,
(TyVMsgCall  LONG          lMsgId,
back*) (      CONST CHAR    *pszDevice
             CONST CHAR    *pcData
             LONG          lDataLen);

```

Parameters

- lMsgId** Numerical identification of the SYS-Message.
 Gives the system area in which the control component messages are written.
 Gives the user area in which the applications can be assigned message IDs.
- pszDevice** The device for which a message is generated or received.
- pcData** User data of a message.

Programming Interface

Constants

- **SYSMSG_ALLPC** Login for all PCs in the network
- **SYSMSG_ALLDEVICE** Login for all devices
- **SYSMSG_LOCALPC** Login for all devices of the local PC

Flags for SysMsgHookCreate

- **SYSMSG_MANUAL_ACK** Manual acknowledgement

This flag must be set if the acknowledgement of the system message is not to be carried out automatically after the callback function.

!! In this case, the application must call up

- **SYSMSG_LOGOUT_CALLBACK** even when logging out.

TySysMsgCallbackInfo Data Type

This data type is used to provide the callback function with data from the system message.

```

typedef struct {
    VOID          *pUserParam;
    LONG          lDevice;
    CONST CHAR    *pcData;
    LONG          lDataLen;
    LONG          lStatus;
    LONG          lHookId;
    LONG          lMsgId;
    LONG          lPc;
    LONG          lFarDevice;
} TySysMsgCallbackInfo;

```

Name	Description
pUserParam	User-defined parameter. Refer also to SysMsgHookCreate
IDevice	The local device address of the system message, or SYSMSG_ALLDEVICE, if the system message is not issued specifically for the device.
pcData	Message data
ldataLen	Length of data
IStatus	Callback status 0: OK Otherwise, logout or error
IHookId	ID of the connected hook
IMsgId	The message ID
IPc	The PC from which the system message originates. SYSMSG_LOCALPC, if the system message has been requested from the local PC.
IFarDevice	The device address in the network. If the network has not been activated then the local device address is accepted. SYSMSG_ALLDEVICE, if the system message has not been issued for a specific device.

TySysMsgCallback Data Type

```
typedef VOID ( __stdcall
*TySysMsgCallback) (TySysMsgCallbackInfo *pCbInfo);
```

The function pointer data type for the callback.

SysMsgHookCreate

The callback function is called up in a try catch block.

The callback is activated in a separate thread.

Note: No further system messages are handled while the function is active.

The transmitted data is generated on the stack and becomes invalid at the end of the return of the function.

Declaration	LONG SysMsgHookCreate(LONG LONG LONG LONG TySysMsgCallback VOID LONG	*plHookId, lDevice, lPC, lMsgId, CallbackFunc, *pParam, lFlags);
Parameters:	plHookId	Output Parameter: Handle on the hook This value is needed to deactivate the callback or to make a manual acknowledgement. Refer also to SysMsgHookDelete() and SysMsgHookAcknowledge()	
	IDevice	Device number for which this callback is to be activated. A valid local or Far Device number. To log on the system message for all devices on a PC, SYSMSG_ALLDEVICE can also be transferred. In this case IPC defines the PC for which the callback is activated.	

Declaration	LONG SysMsgHookCreate(LONG LONG LONG LONG TySysMsgCallback VOID LONG	*pIHookId, lDevice, lPC, lMsgId, CallbackFunc, *pParam, lFlags);
	IPC	<p>Defines the PC for which the callback is to be activated.</p> <p>!! This parameter is only effective if lDevice is transferred with SYSMMSG_ALLDEVICE.</p> <p>SysMsgHookCreateAll() must be used to activate the callback for all devices of all PCs in a network.</p>	
	lMsgId	The message ID	
	CallbackFunc	<p>Callback user function.</p> <p>!! This function is activated in a separate thread.</p> <p>!! No further system messages are handled while the function is active.</p> <p>The transmitted data is generated on the stack and becomes invalid at the end of the return of the function.</p>	
	pParam	User Parameter for the callback function.	
	lFlags	<p>The value of this parameter switches on certain hook options. The options can be abstracted by means of a logical OR operation ' '. The system message is not acknowledged automatically when the callback function is exited. The application must acknowledge receipt of the system message by means of the SysMsgHookAcknowledge() function.</p> <p>SYSMMSG_LOGOUT_CALLBACK: On LogoutIF(), the callback is activated one last time.</p> <p>If necessary, the application can delete user data allocated to the hook.</p>	
Return value:	0	OK	
	NET_ETIMEOUT:	Time for making a connection to a remote device has been exceeded.	
	NET_EINVPARAM	Invalid parameter	
	NET_EFALSE	LoginIf must be called up before system messages can be logged in.	

SysMsgHookAcknowledge

Acknowledgement of a system message in the network.

Declaration	LONG SysMsgHookAcknowledge	(LONG lHookId);
Parameters:	lHookId	HookId of the system message that is to be acknowledged.
Return value:	0	OK
	NET_EINVPARAM	Invalid parameter
	NET_EFALSE	LoginIf must be called up before this function can be called up.

SysMsgHookDelete

Deletion of a system message in the network.

Declaration	LONG SysMsgHookDelete (LONG lHookId);	
Parameters:	lHookId	HookId of the system message that is to be deleted.
Return value:	0	OK
	NET_EINVPARAM	Invalid parameter
	NET_EFALSE	LoginIf must be called up before this function can be called up.

SysMsgHookCreateAll

Generates multiple hooks for system messages. A hook is generated for each PC in the network and is activated for each device.

A hook ID is returned for each PC. If plHookId is 0, the hook IDs are not returned.

Declaration	LONG SysMsgHookCreateAll (LONG *plHookId, LONG lHooktabMax, LONG *plHooktabCnt, LONG lMsgId, TySysMsgCallback CallbackFunc, VOID *pParam, LONG lFlags);	
Parameters:	plHookId	Output parameter: handles on the hooks to the various PCs These values are needed to deactivate the callback or to make a manual acknowledgement. Refer also to SysMsgHookDelete() and SysMsgHookAcknowledge()
	lHooktabMax	Size of the hook table the user is administering. → The function generates as many hooks as there are PCs declared in the network.
	plHooktabCnt	Transfer of number of hooks generated.
	lMsgId	The message ID
	CallbackFunc	Callback user function. !! This function is activated in a separate thread. !! No further system messages are handled while the function is active. The transmitted data is generated on the stack and becomes invalid at the end of the return of the function.
	pParam	User Parameter for the callback function.
	lFlags	The value of this parameter switches on certain hook options. The options can be OR operated ' '. The system message is not acknowledged automatically when the callback function is exited. The application must acknowledge receipt of the system message by means of the SysMsgHookAcknowledge() function. SYSMMSG_LOGOUT_CALLBACK: On LogoutIF(), the callback is activated one last time. If necessary, the application can delete user data allocated to the hook.
Return value:	0	OK

NET_ETIMEOUT:	Time for making a connection to a remote device has been exceeded.
NET_EINVPARAM	Invalid parameter
NET_EFALSE	Loginlf must be called up before system messages can be logged in.

Example of Programming

The following example describes an application of the system message mechanism. A class is declared that works with the functions described:

Declaration

```
#include "indif000.h"    // Declaration of the FI-Routines

//trigger up to maximum of 100 PCs in the PC network(more
than //100 are currently not permitted)
#define D_nSYMSMSG_MAX_PC      100

//e.g., react to 28 messages
#define D_nSysMSG_COUNT      28

// Data structure to management (s.u.)
typedef struct _TyHookTable
{
    LONG  alHookId[D_nSYMSMSG_MAX_PC];
    LONG  lCount;
} TyHookTable;

// Class declaration
/*##{class}#####

-----
Description:
Sample Class for FI System message Handling

-----
History:
#####{end}##*/
class KSysMsgSample
{
    /**
    /** Construction
    /**
    public:
        KSysMsgSample();
        ~KSysMsgSample();

// ...

    /**
    /** Attributes
```

```

/**
private:
    // Callback Routine
    static void __stdcall
    SysMsgCallback(TySysMsgCallbackInfo* pCbInfo);

    //Help Routines
    LONG HookSysMsg (void);
    LONG UnhookSysMsg (void);

    // Data structures to SysMsg-Handling
    static LONG s_alMsgList[D_nSysMSG_COUNT];
    static LONG s_lMsgCount;

TyHookTable  c_aoHookTable[D_nSysMSG_COUNT];

};

```

Implementation

// List of SYS messages to be handled (example)

```

LONG KSysMsgSample::s_alMsgList[] =
{
    MSG_PARUPDBEG,
    MSG_PCLUPDBEG,
    MSG_LAGCHABEG,
    MSG_MECERRGEN,
    MSG_SYSERRGEN,
    MSG_FWAUPDBEG,
    MSG_MEMUPDBEG,
    MSG_ACTERRBEG,
    MSG_DEVERRBEG,
    MSG_MDLERRBEG,

    MSG_PARUPDEND,
    MSG_PCLUPDEND,
    MSG_LAGCHAEND,
    MSG_MECERRDEL,
    MSG_SYSERRDEL,
    MSG_FWAUPDEND,
    MSG_MEMUPDEND,
    MSG_ACTERREND,
    MSG_DEVERREND,
    MSG_MDLERREND,

    MSG_STRTUPCHG,
    MSG_WARNINCHG,
    MSG_SETUP_CHG,
    MSG_MESSAGCHG,
    MSG_ERROR_CHG,
    MSG_SFCERRCHG,
    MSG_SFCMODCHG,
    MSG_DMPSELCHG
}

```



```

};

// Login of system messages
LONG KSysMsgSample::HookSysMsg (void)
{
    LONG lResult = 0;

    for (INT i=0; i< D_nSysMSG_COUNT; ++i)
    {
        // Login SysMessages for all PCs in network.
        lResult = :: SysMsgHookCreateAll(
            c_aoHookTable[i].alHookId,
            D_nSYSMSG_MAX_PC,
            &(c_aoHookTable[i].lCount),
            s_alMsgList[i],
            SysMsgCallback,
            static_cast<VOID*>(this),
            0);
    }
    return lResult;
}

// Logout of system messages
LONG KSysMsgSample::UnhookSysMsg (void)
{
    for (INT i=0; i< D_nSysMSG_COUNT; ++i)
    {
        for (INT j=0; j<c_aoHookTable[i].lCount; j++)
        {
            ::SysMsgHookDelete(
                c_aoHookTable[i].alHookId[j]);
        }
    }

    return 0;
}

// Callback function
void KSysMsgSample::SysMsgCallback(
    TySysMsgCallbackInfo* pCbInfo)
{
    KSysMsgSample * pInstance =
        static_cast<KSysMsgSample*>(pCbInfo->pUserParam);

    switch (pCbInfo->lMsgId)
    {
        case MSG_PARUPDBEG:
        case MSG_PCLUPDBEG:
        case MSG_FWAUPDBEG:
            // handle begin download
            break;
        case MSG_PCLUPDEND:
        case MSG_PARUPDEND:
    }
}

```

```

        case MSG_FWAUPDEND:
            // handle end download
            break;

        //etc.
    }
}

```

What SYS Messages are There and How Should I React to Them?

The SYS messages that are most often used for a client are listed in the following table. All system messages can be found in the configuration files "INDIF000.H" and "INDIF000.BAS". The system messages always consist of a SYS-MSG and a SYS-MSG acknowledgement.

Note: The device address that has set the system message is returned as an ASCII character in the buffer "CHAR *pcSysMsgBuffer" of "GetIfMsg". The buffer may possibly contain additional SYS-MSG information such as the parameter identification string.

SYS Message	What happens?	Reaction from the Client
MSG_FWAUPDBEG	The firmware download commences, i.e. the firmware is loaded into the System200 components by the GUI.	Communication between all configured Bosch Rexroth devices is interrupted.
MSG_FWAUPDEND	End of firmware download.	Communication recommences. PLEASE NOTE: Process and axis configuration data etc., may have changed.
MSG_PARUPDBEG	The parameter download commences, i.e. the parameter set is loaded into the System200 components by the user interface.	Communication between all configured Bosch Rexroth devices is interrupted.
MSG_PARUPDEND	End of parameter download	Resumption of communication PLEASE NOTE: Process and axis configuration data etc., may have changed.
MSG_PCLUPDBEG	The PLC program download commences, i.e. the PLC program is loaded into the System200 components by the GUI.	Communication between all configured Bosch Rexroth devices is interrupted.
MSG_PCLUPDEND	End of PLC program download	Communication recommences.
MSG_MEMUPDBEG	GUI begins to delete the data memory in the System200 components.	Communication between all configured Bosch Rexroth devices is interrupted.
MSG_MEMUPDEND	GUI has deleted the data memory in the System200 components.	Communication recommences. PLEASE NOTE: Configuration data has been deleted.
MSG_SYSERRGEN	If there is a system error, this SYS-MSG is issued, i.e. the PLC cannot be accessed at the moment.	No interruption of communication to the Bosch Rexroth devices is necessary. (Used for presenting a system error from a particular Bosch Rexroth device in graphic form).
MSG_SYSERRDEL	A system error is deleted.	No interruption of communication to the Bosch Rexroth devices is necessary. (Information that the system error is no longer present at a particular Bosch Rexroth device.)

SYS Message	What happens?	Reaction from the Client
MSG_MECERRGEN	This SYS-MSG is issued if there is a fault in the mechanism.	No interruption of communication to the Bosch Rexroth devices is necessary. (Is used for presenting a system error from a particular Bosch Rexroth device in graphic form).
MSG_MECERRDEL	A mechanism error is deleted.	No interruption of communication to the Bosch Rexroth devices is necessary. (Information that the system error is no longer present at a particular Bosch Rexroth device.)
MSG_LAGCHABEG	A language switch has been initialized at the Bosch Rexroth GUI.	No interruption of communication to the Bosch Rexroth devices is necessary. (Information that the user interface language is being switched.)
MSG_LAGCHAEND	A language switch has been completed at the Bosch Rexroth GUI.	No interruption of communication to the Bosch Rexroth devices is necessary. (Information, that the user interface language has been switched.)
MSG_PC__ALIVE	A PC/device logs in/out of the PC network. The device address/FarDevice address that has set the system message is returned as an ASCII character in the buffer "CHAR *pcSysMsgBuffer" of the "GetIfMsg". The buffer also contains the information "0" for PC logged out or a "1" for PC logged in.	Depending on the application, this system message is used on the one hand purely as information regarding the logging in/logging out of PCs. On the other hand, communication to the PC/device or the re-establishment of communication with subsequent synchronization is required.

SYS-MSG Example (Visual Basic: SYSTHREAD.BAS)

```

Public Sub SysMsgThreadProc ()
'This subroutine is processing FI-SYS-Messages
'*****
Dim lWait As Long
Dim hEvList(0 To 1) As Long
'Transmission of events whose the thread is reacting
'*****
hEvList(0) = hTerminateEvent
hEvList(1) = hSysMsgEvent
Demo.SYS_Messages.BackColor = QBColor(10) 'set BackgroundColor to bright green
Demo.SYS_Messages.Caption = "Thread is running"
Do 'Threadloop
    lWait = WaitForMultipleObjects(2, hEvList(0), 0, INFINITE)
    If lWait = 0 Then 'TerminateEvent from another FI-application has occurred
        Demo.SYS_Messages.BackColor = QBColor(12) 'set BackgroundColor to bright red
        Demo.SYS_Messages.Caption = "Terminate Event has occurred"
        Demo.TerminateEvent = True
        Exit Do 'End of the threadloop
    ElseIf lWait = 1 Then
        'SysMsgs which are received from the FI
        '*****
        Dim lRet As Long 'Routine's returnvalue
        Dim nMsgNr As Integer
        Dim strMsgBuf As String * 256
        Dim strTaskName As String * 256
        Dim szMsg As String
        'Getting SYS-MSG-number
        '*****
        lRet = GetIfMsg(nMsgNr, strMsgBuf, 256, strTaskName)
        If lRet Then 'error handling
            szMsg = "GetIfMsg terminated with error code: " + CStr(lRet)
            Demo.SYS_Messages.Caption = szMsg
        Else
            Select Case nMsgNr
            Case MSG_PCLUPDBEG
                Demo.SuspendCyclicFunc 'Termination of a cyclic request
                lRet = SetIfMsgConf(MSG_PCLUPDBEG_Q) 'verification of the SYS-Message
                If lRet Then 'error handling
                    szMsg = "SetIfMsgConf terminated with error code: " +
CStr(lRet)

```

```

        Demo.SYS_Messages.Caption = szMsg
    End If
    Demo.SYS_Messages.BackColor = QBColor(11) 'set BackgroundColor to
bright blue
        Demo.SYS_Messages.Caption = "PLC Download BEGIN"
    Case MSG_PCLUPDEND
    lRet = SetIfMsgConf(MSG_PCLUPDEND_Q) 'verification of the SYS-Message
    Demo.ResumeCyclicFunc 'Start of a cyclic request
    Demo.SYS_Messages.BackColor = QBColor(6) 'set BackgroundColor to
brown
        Demo.SYS_Messages.Caption = "PLC Download END"
    If lRet Then 'error handling
        szMsg = "SetIfMsgConf terminated with error code: " +
CStr(lRet)
        Demo.SYS_Messages.Caption = szMsg
    End If
    Sleep (2000) 'Wait 2 sec.
    Demo.SYS_Messages.BackColor = QBColor(10) 'set BackgroundColor to
bright green
        Demo.SYS_Messages.Caption = "Thread is still running"
    Case MSG_PARUPDBEG
    Demo.SuspendCyclicFunc 'Termination of a cyclic request
    lRet = SetIfMsgConf(MSG_PARUPDBEG_Q) 'verification of the SYS-Message
    If lRet Then 'error handling
        szMsg = "SetIfMsgConf terminated with error code: " + CStr(lRet)
        Demo.SYS_Messages.Caption = szMsg
    End If
    Demo.SYS_Messages.BackColor = QBColor(11) 'set BackgroundColor to bright blue
    Demo.SYS_Messages.Caption = "Parameter Download BEGIN"
    Case MSG_PARUPDEND
    lRet = SetIfMsgConf(MSG_PARUPDEND_Q) 'verification of the SYS-Message
    Demo.ResumeCyclicFunc 'Start of a cyclic request
    Demo.SYS_Messages.BackColor = QBColor(6) 'set BackgroundColor to brown
    Demo.SYS_Messages.Caption = "Parameter Download END"
    If lRet Then 'error handling
        szMsg = "SetIfMsgConf terminated with error code: " + CStr(lRet)
        Demo.SYS_Messages.Caption = szMsg
    End If
    Sleep (2000) 'Wait 2 sec.
    Demo.SYS_Messages.BackColor = QBColor(10)
    Demo.SYS_Messages.Caption = "Thread is still running"
    End Select
    End If
    ElseIf lWait = 2 Then
        Exit Do 'End of Threadloop
    End If
Loop
End Sub

```

5.6 COM - Automation Interface

The function interface supports two simple COM automation interfaces:

- IFIObject, and
- IFIData.

IFIObject

Explanation IFIObject represents the following functions of a function interface (FI)

- LogInIf,
- LogOutIf, and
- DataTransfer.

IFObject::LogInIf

IDL description HRESULT LogInIf([in] BSTR bstrTaskName)

Explanation Compared to the FI function, this method has been greatly simplified. For a more detailed description, refer to the "LogInIf" FI function.

Parameters bstrTaskName: Name of the user program

Example Visual Basic Script

```
Dim oFIObject

On Error Resume Next
Set oFIObject = CreateObject("Indif000.FIObject")
if Not oFIObject is Nothing then
    oFIObject.LogInIf("fi.vbs")
end if

Set oFIObject = Nothing
```

Example JavaScript

```
var oFIObject;
try{
    oFIObject = new ActiveXObject("Indif000.FIObject");
    if (oFIObject == null)
    {
        return;
    }

    oFIObject.LogInIf("fi.js");
    oFIObject = null;
}
catch(e)
{
}
```

IFObject::LogOutIf

IDL description HRESULT LogOutIf();

Explanation Here, refer to the "LogOutIf" FI function description.

IFObject::DataTransfer

IDL description HRESULT DataTransfer([in] BSTR bstrFunction,
[in, defaultvalue(32000)] long IResSize,
[in, defaultvalue(3)] long IResType,
[in, defaultvalue("")] BSTR bstrValue,
[in, defaultvalue(3)] long IValType,
[out, retval] IFIData **ppoData);

Explanation Compared to the "DataTransfer" FI function, the IFObject DataTransfer was made easier to operate in view of the parameters. The sequence of the parameters has been changed, and some parameters have been pre-assigned default values.

Parameters

- bstrFunction:** Function interface command
- IResSize:** Length of the result buffer, default 32000 bytes
- IResType:** Data code of result data, default 3 (ANSI)
- bstrValue:** Writing value, default empty string
- IValType:** Data code of the value to be written, default 3 (ANSI)
- ppoData:** A DataTransfer request results in an IFIData object. In case of error, a zero object is returned.

Example Visual Basic Script

```
Dim oFIData
Set oFIData = oFIObject.DataTransfer("00_CR_PVF_bool0")
if Not oFIData is Nothing then
end if
```

The following source code corresponds exactly to the above example

```
Dim oFIData
Set
oFIData=oFIObject.DataTransfer("00_CR_PVF_bool0",32000,3,"",
3)
if Not oFIData is Nothing then
end if
```

Example JavaScript

```
var oFIData;
oFIData = oFIObject.DataTransfer("00_CR_PVF_bool0");
if (oFIData == null)
{
return;
}
```

The following source code corresponds exactly to the above example

```
var oFIData;
oFIData = oFIObject.DataTransfer("00_CR_PVF_bool0", 32000,
3, "", 3);
if (oFIData == null)
{
return;
}
```

IFIData

Explanation IFIObject represents the following functions of a function interface (FI)
 GetNumberOfRows,
 GetNumberOfItems, and
 ReadGroupItem.

IFIData::GetNumberOfRows

IDL description HRESULT GetNumberOfRows([out, retval] long *pIRows)

Explanation Here, refer to the "GetNumberOfRows" FI function description.

Parameters pRows: Number of rows of an FI result.

Example Visual Basic Script
 Dim IRow
 IRow = oFIData.GetNumberOfRows()

Example JavaScript
 var IRow;
 IRow = oFIData.GetNumberOfRows()

IFIData::GetNumberOfItems

IDL description HRESULT GetNumberOfItems([in] long IRow, [out, retval] long *pItem)

Explanation Here, refer to the "GetNumberOfItems" FI function description.

Parameters IRow: [1, 256]
 pItem: Number of partial results

Example Visual Basic Script
 Dim IColumn
 IColumn = oFIData.GetNumberOfItems(1)

Example JavaScript
 var IColumn;
 IColumn = oFIData.GetNumberOfItems(1);

IFIData::ReadGroupItem

IDL description HRESULT ReadGroupItem([in] long IRow, [in] long IItem, [in, defaultvalue(32000)] long IBufSize, [out, retval] BSTR *pbstrVal)

Explanation Here, refer to the "ReadGroupItem" FI function description. The only difference is the default value assignment of the IBufSize parameter.

Parameters IRow: -1: Output of a complete table, i.e. all rows of a request
 [1...256]: the respective result line
 IItem: -1: Output of a row
 0: Output of the requested command with management information
 [1...n]: Individual result (Element of a row)
 IBufSize: Length of the buffer for partial result, default 32000 bytes.
 pbstrVal: Requested partial result

Example Visual Basic Script
 Dim strText
 strText = oFIData.ReadGroupItem(-1, -1)

Example JavaScript
 var strText;
 strText = oFIData.ReadGroupItem(-1, -1);

Example for the Total Script

```

Visual Basic Script Dim oFIObject
                        Dim oFIData
                        Dim lRow
                        Dim lColumn
                        Dim strText

                        On Error Resume Next
                        Set oFIObject = CreateObject("Indif000.FIObject")
                        if Not oFIObject is Nothing then

                                oFIObject.LogInIf("fi.vbs")
                                if Err = 0 then

                                        Set oFIData =
                                        oFIObject.DataTransfer("00_CR_PVF_bool0")
                                        if Not oFIData is Nothing then
                                                lRow = oFIData.GetNumberOfRows()
                                                MsgBox "Rows: " & CStr(lRow)
                                                lColumn = oFIData.GetNumberOfItems(1)
                                                MsgBox "Columns:" & CStr(lColumn)
                                                MsgBox oFIData.ReadGroupItem(-1, -1)
                                        end if
                                        call oFIObject.LogOutIf()
                                end if
                        end if

                        Set oFIObject = Nothing

JavaScript var oFIObject;
                var oFIData;
                var lRow;
                var lColumn;
                var strText;

                function FI()
                {
                        try{
                                oFIObject = new ActiveXObject("Indif000.FIObject");
                                if (oFIObject == null)
                                {
                                        return;
                                }

                                oFIObject.LogInIf("fi.js");
                                oFIData = oFIObject.DataTransfer("00_CR_PVF_bool0");
                                if (oFIData == null)
                                {
                                        return;
                                }

                                lRow = oFIData.GetNumberOfRows();
                                strText = "Rows: " + lRow.toString();
                                WScript.Echo ( strText );
                                lColumn = oFIData.GetNumberOfItems(1);

```



```

        strText = "Columns: " + lColumn.toString();
        WScript.Echo ( strText );
        WScript.Echo (oFIData.ReadGroupItem(-1, -1));
        oFIData = null;
        oFIObject.LogOutIf();
        oFIObject = null;
    }
    catch(e)
    {
    }
}

FI();

```

5.7 Network

There is a separate documentation on working with the PC network:

It is called 'PC Network',

File name: DOK-CONTRL-PC**NET*V22-AW01-EN-P,

Mat. no.: R911296203

Z. No. 120-0400-B396-01

Its status is V22; there will no new version of this Documentation for V23.

5.8 Tips and Tricks when Working with the Interface

This chapter provides you with tips and tricks that are designed to help you to proceed faster when developing your user program (client) (see also Chapter "Installing Windows NT and the Function Interface").

Furthermore, we have discovered that using Windows NT without a mouse always tends to present difficulties and we have therefore listed the most important Windows NT key combinations in a table.

Problem	Remedy
In your application, you issue an FI command and receive: <ul style="list-style-type: none"> no response - or - an unexpected response - or - an error code (see Chapter 6 "Error Codes") 	Frequent cause: Device address has not been given or been incorrectly given! <ul style="list-style-type: none"> ⇒ Check the correct details of the FI command (see Chapter 5 "Installing Windows NT and the Function Interface" and Chapter "Function Interface Commands"). ⇒ Issue the FI command that is causing problems using the VBDemo.exe program (see Issuing FI Commands with the "VBDemo" Application).
Your client no longer reacts	⇒ See clearing the Memory using the "KILLTASK.EXE" Tool
Your client terminates "DR. WATSON" with a Windows access violation.	⇒ See clearing the Memory using the "KILLTASK.EXE" Tool ⇒ Correct the programming error and re-start your application.
The entire system (Windows NT, client and Bosch Rexroth GUI) is reacting slowly.	⇒ Check the Windows NT settings for improved performance, idling activity, swapping the core-mode driver and idling activity in accordance with Chapter "Installing Windows NT and the Function Interface".
Clearing the memory using the "KILLTASK.EXE" tool doesn't work, i.e. neither the three base processes of the function interface nor the client is removed from memory.	⇒ Start Task manager, for example, using the key combination <Ctrl>+<Shift>+<Esc> (see chapter "Windows NT Task Manager"). ⇒ Click on the "Processes" tab page. ⇒ Terminate the three basic processes of the function

	<p>interface and your user program, if applicable:</p> <ul style="list-style-type: none"> • LOGINTFC.EXE (logic process) • COMINTFC.EXE (communication process) • BOFINTFC.EXE (management process) <p>via the <Terminate process> button.</p>
<p>Your application terminates because: required files are missing - or - path entries do not exist or are incorrect.</p>	<p>⇒ Check to make sure the required files are located in their respective directories.</p> <p>⇒ Check the path entries.</p> <p>Note!</p> <ul style="list-style-type: none"> • Absolute paths should be avoided in the application as any later changes in the drive path (e.g. from C:\ to D:\) or in the directory structure are not supported. • The system directory as well as the Windows NT disk drive can also be freely selected.

Clearing the Memory Using the "KILLTASK.EXE" Tool

This tool can be used when creating software for clearing the memory. After a standard installation (see chapter "Installing Windows NT and the Function Interface") it is located in the default directory "C:\Programme\Indramat\MTGUI\Bin".

The tool provides you with the following two options for clearing the memory:

- complete reinitialization and
- selective reinitialization of the function interface.

complete reinitialization On starting the "KILLTASK.EXE" application, the following three basic processes of the function interface:

- logic process (LOGINTFC.EXE),
- communication process (COMINTFC.EXE),
- management process (BOFINTFC.EXE)

are removed from the memory, as well as all applications connected to the function interface.



CAUTION

You should first terminate all other (stable) function interface applications correctly.

Only run Killtask after doing this!

If this stipulation is not observed then all claims against Bosch Rexroth are excluded.

To completely reinitialize, proceed as follows:

- ⇒ Click on Start and then on the "Run" option.
- ⇒ Click on the <Find> button to search for the "KILLTASK.EXE" tool.

Note: After a standard installation (see Chapter "Installing Windows NT and the Function Interface") the "KILLTASK.EXE" application is located in the default directory "C:\Programme\Indramat\MTGUI\Bin".

- ⇒ Click on the <OK> button.
- All applications connected to the function interface – and the basic processes of the function interface itself – are removed from the memory.

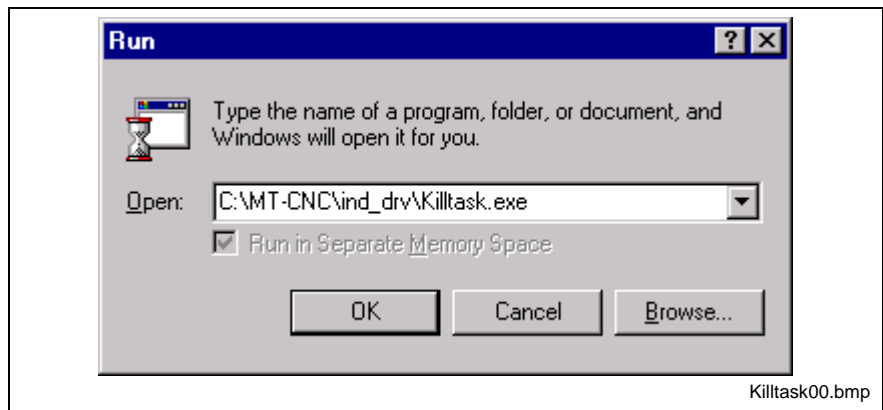


Fig. 5-6: "Run" dialog box of Windows NT: Complete re-initialization

Selective reinitialization

Only those applications that are connected to the function interface are removed from the memory and from the function interface management structure.

To selectively reinitialize, proceed as follows:

⇒ Click on Start and then on the "Run" option.

Note: You can search for the "KILLTASK.EXE" application by clicking on the "Find..." button. After a standard installation (see chapter "Installing Windows NT and the Function Interface") this file is located in the default directory "C:\Programme\Indramat\MTGUNBin".

⇒ In the text box, enter the name of the application that is to be removed from the memory and from the management structure of the function interface (here VBDemo.exe).

⇒ Then click on the <OK> button.

The client (here "VBDemo.exe") is removed from the memory and from the management structure of the function interface.

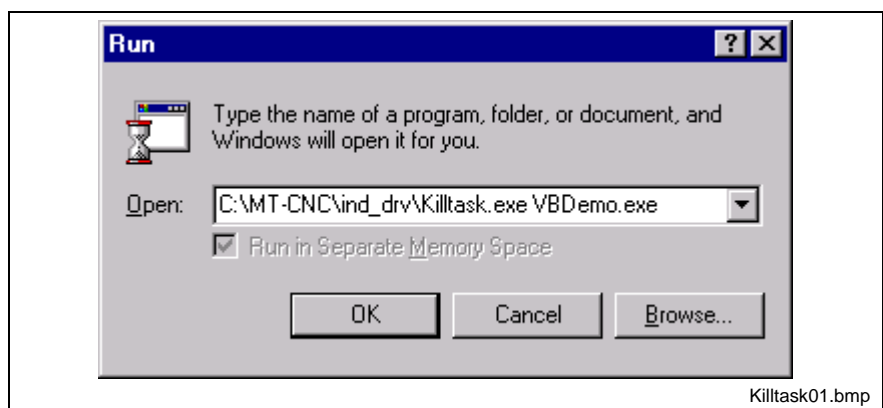


Fig. 5-7: "Run" dialog box of Windows NT: Selective re-initialization

Issuing FI Commands Using the “VBDemo” Application

Single FI commands and cyclical requests can be issued by the “VBDemo” application.

To start the application, proceed as follows:

⇒ Click on start, point to Programs, then to Bosch Rexroth and finally to FI.

⇒ Click on VBDemo.

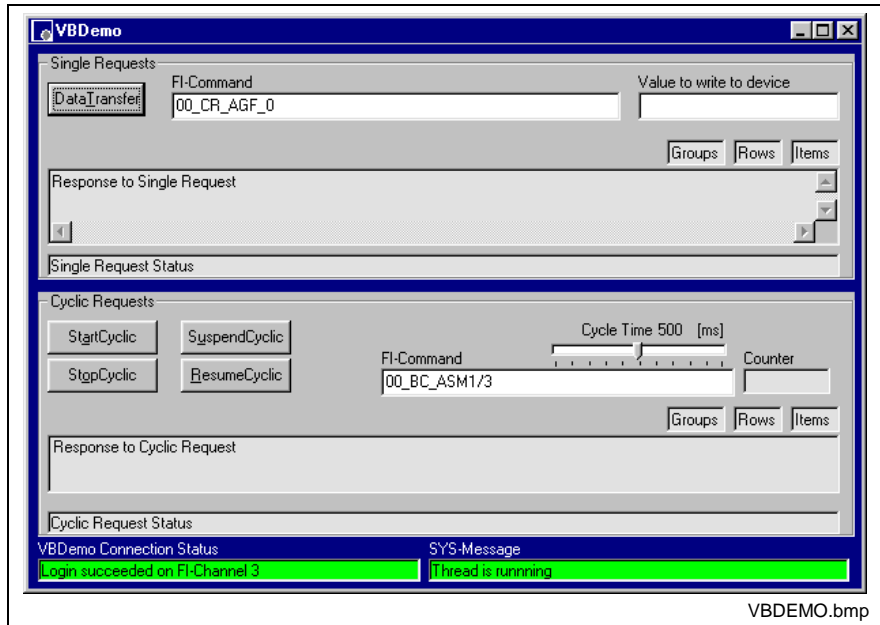


Fig. 5-8: The "VBDemo" user program

“Single Requests” Dialog Box

This dialog box allows single requests to be issued that both read and write using the “Data Transfer” routine.

To do this, enter the FI command in the “FI-Command” entry field. If a write request is made, then also enter the value that you wish to write in the “Value to write to device” box (see Chapter “Function Interface Commands”).

Then issue the FI command you have entered to the function interface by clicking on the <DataTransfer> button.

The response from the function interface is displayed in the “Response to Single Request” text box.

“Cyclic Requests” Dialog Box

This dialog box allows cyclic requests that write to be issued using the “StartCyclicPipe” routine.

To do this, enter the FI command in the entry field “FI-Command” (see chapter “Function Interface Commands”).

Then issue the FI command entered cyclically to the function interface by clicking in the <StartCyclic> button.

The response from the function interface is displayed in the “Response to Cyclic Request” text box.

Note: You can change the request time from between 10 to 100 ms using the “Cyclic Time” thumb switch.

To stop the cyclic request, click on the <StopCyclic> button. This will cause the “StopCyclicPipe” routine to be executed.

Note: Clicking on the <SuspendCyclic> button processes the “suspendCyclicPipe” routine and sets the cyclic request to standby mode. To reactivate the cyclic request, click on the <ResumeCyclic> button. This will cause the “ResumeCyclicPipe” routine to be executed.

"VBDemo Connection Status" Dialog Box

Displays the login status of the application at the function interface. There are two statuses:

- The dialog box is shaded green and shows the function interface channel (LOG channel) that has been assigned to the application.
- The dialog box is shaded in red and shows the error code that has been caused by logging in via the “LogInIf” login routine.

Starting "VBDemo" in Diagnostics Mode

To start the “VBDemo” program in diagnostics mode, proceed as follows:

- ⇒ Open the Windows NT Explorer. To do this, click on Start, point to Programs and then click on Windows NT Explorer.
- ⇒ Via Winnt, go to Profiles and into the User Profile by which the function interface was installed.
- ⇒ Click on the Start Menu, point to Programs, then to Bosch Rexroth and finally click on “FI”.
- ⇒ Click on VBDemo and open the Properties dialog box via the Explorer menu file.
- ⇒ Click on the tab page link and enter the start parameter “/c=t /b=w” in the “Target” text field.
- ⇒ Click on the <C_l_o_s_e> button and VBDemo will be started in diagnostics mode the next time it is called.

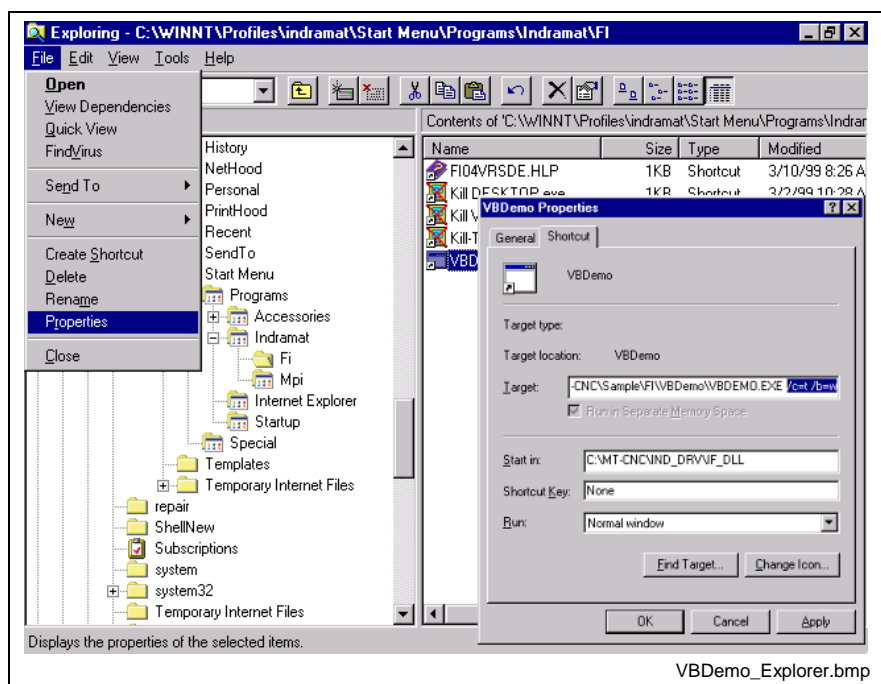


Fig. 5-9: Starting VBDemo in the diagnostics mode of the function interface

Outputting Diagnostic Messages

By passing on the start parameters when starting the management process "BOFINTFC.EXE", various function interface diagnostic messages can be outputted to the screen.

To start the function interface in diagnostics mode, proceed as follows:

⇒ Click on Start and then on the "Run" option.

Note: You can search for the management process "BOFINTFC.EXE" by clicking on the "Find" button. After a standard installation (see chapter "Installing Windows NT and the Function Interface Commands") this file is located in the default directory "C:\Programme\Indramat\MTGU\Bin\".

⇒ Enter the start parameter "/c=t /b=w" in lower case letters in the text box (observe spaces between entries).

⇒ Then click on the <OK> button.

The function interface is now started in diagnostics mode.

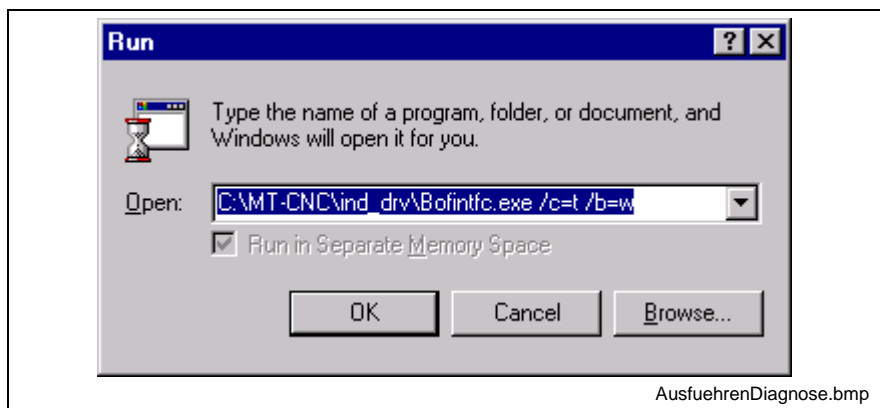


Fig. 5-10: "Run" dialog box of Windows NT: BOFINTFC.EXE\$

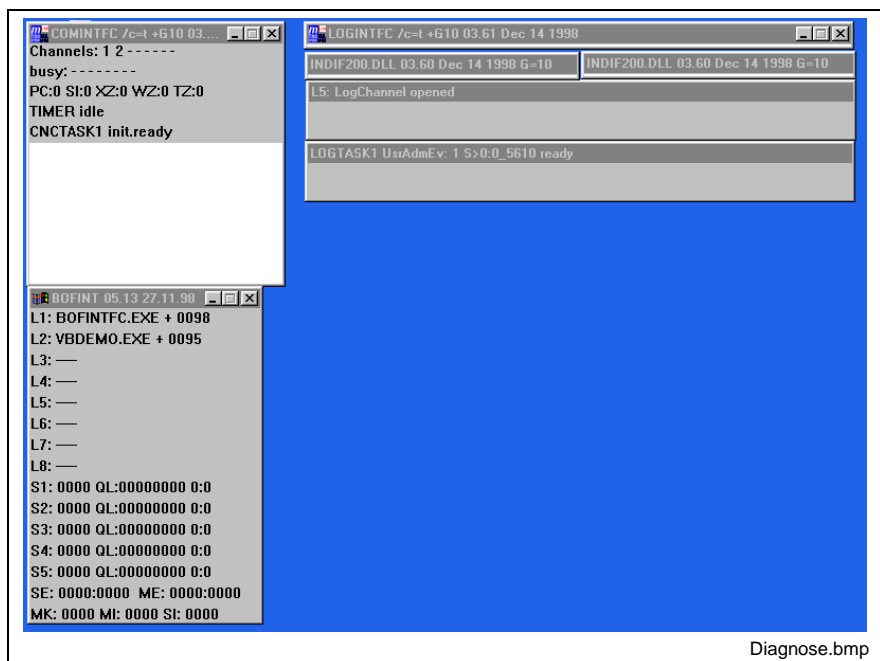


Fig. 5-11: Diagnose mode of the function interface

- Meaning of the Counters** Five counters are shown in the 3rd line of the diagnostics window of the communication process (COMINTFC.EXE). The individual counters mean the following:
- PC** Number of communication errors that have occurred in the direction of transmission from device → PC.
 - SI** Number of communication errors that have occurred in the direction of transmission from PC → device.
 - XZ** Number of communication repetitions that were required to transfer a valid telegram to the device.
 - WZ** The counter is increased if, in spite of five repetitions, it has not been possible to transmit a valid telegram to the device. The counter is increased by one if the "XZ" counter has been previously increased by five. In this case, the timeout counter is also increased by one.
 - TZ** Timeout counter. The number of timeouts that occur are counted in this counter. A timeout is generated if, in spite of five repetitions, it has not been possible to transmit a valid telegram to the device.
- The active control channels are displayed in the lower lines (CNC/DMA-Task).
- Data accesses made by the individually connected applications are displayed in the diagnostics window of the LOG channel on the left side of the screen.
- In the control window of the management process (BOFINT), the applications are shown that are known in the management mechanism of the BOF process.

Windows NT Key Combinations

The most important key combinations for using Windows NT without a mouse are displayed in the following table.

Action	Key combination
Open start bar	<Ctrl>+<Esc>
Navigate within the opened start bar and in the opened submenus in the start bar	<Arrow key left, right> or <Arrow key up, down>
Select (start) the applications in the opened submenus in the start bar	<Enter>
Start Windows NT Task Manager	<Ctrl>+<Shift>+<Esc>
Move within the Windows NT menu	<Tab>
"Right mouse click" on button moved to	<Shift>+<F10>
Switch within a menu to other tab pages	<Ctrl>+<Tab>
Switch between opened applications	<Alt>+<Tab>

6 Installing Windows NT and the Function Interface

6.1 The Windows NT Operating System

Using the Windows NT operating system and the possibility of running various applications parallel to one another requires a powerful computer.

The hardware requirements depend directly on the number and nature of the applications running concurrently on the PC. This should be taken into account during the project planning phase. The network cards used and their drivers require a great deal of computing power which might then not be available for the rest of the system. Hardware must therefore be selected with great care and utmost precision.

Note: For the Windows NT Operating System, we recommend a PC with a Pentium processor and at least 32 MB RAM, as well as at least 500 MB available space on the hard drive.

Multitasking and Windows NT

Whereas under Windows 3.1x what is known as "cooperative" or "non-pre-emptive" multitasking controlled several applications running concurrently, genuine "pre-emptive" multitasking is integrated into Windows NT.

Non-pre-emptive Multitasking Here, it is not the operating system that decides how much computing time is to be allocated, but the application itself; and the application decides when to surrender time back for a short while to the operating system. The disadvantage of this is that when several applications are running simultaneously, working with them in parallel is only possible to a limited degree.

Pre-emptive Multitasking The operating system itself decides how much computing time is to be allocated to the individual applications. Switching between individual applications is now much more fluid a process as the operating system is able to distribute computing time faster and at shorter intervals, creating the impression that several instructions really can work "simultaneously and unrestrictedly".

Windows NT Task Manager

The applications running can be monitored and controlled by the Task Manager i.e., applications that have been started can be overlaid on the desktop or can be terminated. Furthermore, it is possible to start applications or switch to other applications.

- Calling the Task Manager**
- ⇒ using the key combination <Ctrl>+<Shift>+<Esc>
 - ⇒ clicking with the right mouse button on the taskbar

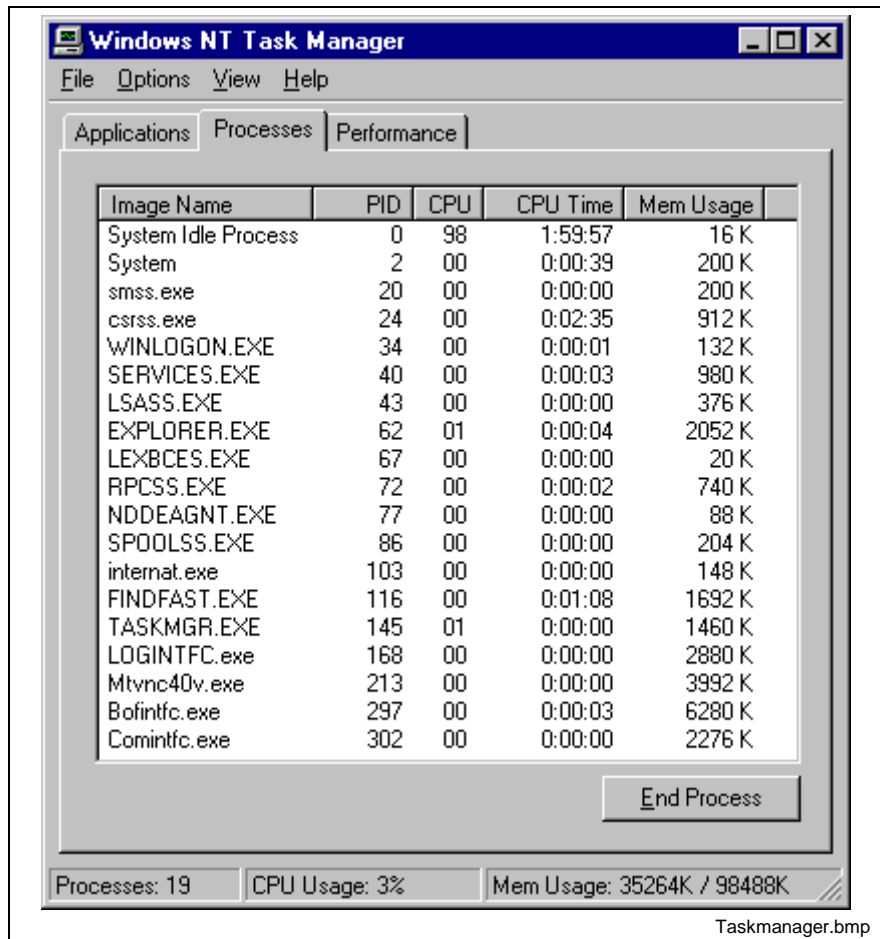


Fig. 6-1: Windows NT Task Manager

Note: You can bring applications consecutively up to the front of the screen (overlay them) using the key combination <Alt>+<Tab> without having to make your selection using Task Manager.

6.2 Setting the Windows NT System Properties

Performance Features

To guarantee an optimal reaction time for the function interface, the performance boost for the application in the foreground should be set to "none".

Note: Safe and error-free operation of the function interface is only ensured when the performance boost for the application in the foreground is set to "none".

To make this setting, proceed as follows:

- ⇒ Click on start, point to Settings, then to System Control and finally to System.
- ⇒ Click on the "System Properties" tab page and set the thumb to "none".
- ⇒ Then click on the <OK> button.

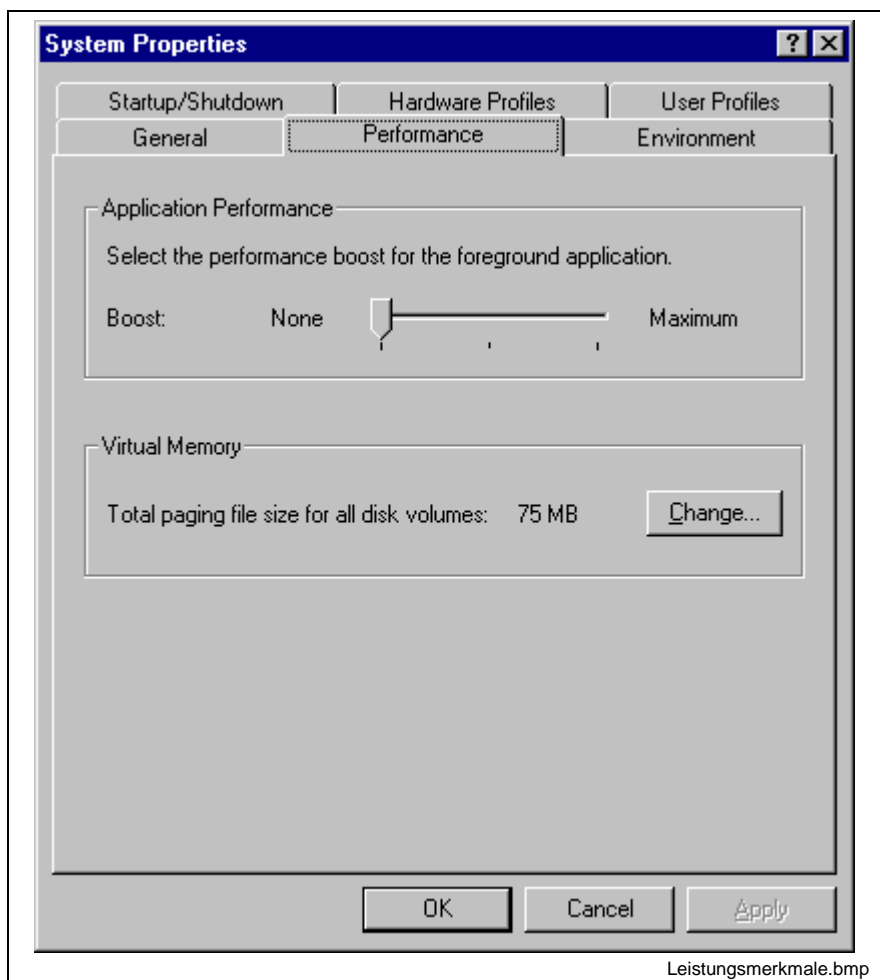


Fig. 6-2: Setting the WindowsNT system property "Performance"

Note: The setting for "Virtual Memory" may differ from the setting of your system.

Date/Time Properties

It is required for the exchange of Bosch Rexroth files between two PCs, to have an identical time zone setting. Furthermore, the automatic clock adjustment for daylight saving (switching between summer and winter times) must be deactivated.

To make this setting, proceed as follows:

- ⇒ Click on Start, point to Settings, then to System Control and finally to Date/Time Properties.
- ⇒ Click on the Time Zone tab page and deactivate the “Automatically adjust clock for daylight saving changes” toggle button.
- ⇒ Then click on the <OK> button.



Fig. 6-3: Date and time settings

6.3 Installing the Function Interface

Note: In Version 08, the function interface cannot be installed separately, but only within the context of the relevant Bosch Rexroth GUI.

6.4 Directory and File Structure of the Function Interface

Contents of the "INDRAMAT.INI" File

The global settings for the function interface are stored in the "Indramat.ini" file. The function interface looks for the file in the "C:\Programme\Indramat\MTGUI\BasicData\Resource" directory.

However, the default directories as well as the drive [LW] can be freely selected. The "INDRAMAT.INI" file corresponds to the Microsoft Windows INI standard and is constructed as follows:

Identifier	Values	Explanation
[IfConfig]		This contains the configuration settings for the function interface
IfInstDir=	e.g.: C:\Programme\ Indramat\MTGUI\Bin	Directory in which the three basic processes of the function interface are installed. This entry is set by the installation program.
AndInstDir=	e.g.: C:\MTA200	!Optional! Directory for MTA 200 control software. Details refer to the "MTA200.EXE" application.
IfDIIMode=	e.g.: 04.10 03.xx [00..70], 04.xx [00,10]	Here the mode is specified that is to be supported by the function interface. The IfDIIMode of a more recent version of the function interface can, for example, be operated in the same mode as the previous version for troubleshooting.
IfVersion=	e.g.: 06V00	Current version of the function interface.
GBOVERSION=	e.g.: 005-21Vxx	Current version of the Bosch Rexroth GUI.
INDRAMAT_x=	x=1 to 9 Name of file	Reference to directory C:\Programme\Indramat\MTGUI\BasicData\Resource The existence of the files named here is checked on starting the function interface. The following applies: File identifier without an extension is a DLL. e.g. Indramat = INDRAMT.DLL Several file identifiers are separated by a "comma".
IND_DLL_x=	x=1 to 9 Name of file	Reference to directory [LW]:\...\MTGUI\Bin. The existence of the files named here is checked on starting the function interface. The following applies: A file identifier without an extension is a DLL, e.g. NDFS100 = INDFS100.DLL. Several file identifiers are separated by a "comma".
IF_DLL_x=	x=1 to 9 Name of file	Reference to directory [LW]:\...\MTGUI\Bin. The existence of the files named here is checked on starting the function interface. The following applies: File identifier without an extension is a DLL. e.g. INDIF000 = INDIF000.DLL Several file identifiers are separated by a "comma".
[Install]		This contains entries regarding the installed System200 software components.
HMIVersion=	e.g.: 01V06	Version ID of the System200 software component WIN-HMI
TYP=	e.g.: HMI	System200 software component WIN-HMI
ServicePack=	e.g.: 2 [1,2,...]	Service Pack ID of the installed System200 software components
SP_Release=	[1, to F]	State of the Service Pack release ID (F = Final Version)

Example Entries in the "INDRAMAT.INI" File

```
[IfConfig]
IfInstDir=C:\Program Files\Indramat\MTGU\Bin
AndInstDir=C:\MTA200
IfDIIMode=04.10
IFVERSION=04V02
GBOVERSION=005-21V09
INDRAMAT_1=indramat
IND_DLL_1=indfs100,indma110,indma900,indut140,indof160
IF_DLL_1=indif000,indif120,indif130,Indif150
IF_DLL_2=indif200,indif210,indif220,indif300,indif310,indif320
IF_DLL_3=indif330,indif340,indif350,indif360,indif400
IF_DLL_4=indif500,indif510,indif520,indif530,indif540,indif550
IF_DLL_5=indif600,indif610,indif700,indif810,indif820,indif840
IF_DLL_6=indifA00

[Install]
HMIVersion=01V06
TYP=HMI
SP_Release=F
```

The DLL entries (If_DLL_1,...) can be expanded up to the ninth entry (If_DLL_9). A check for the existence of the DLLs is only made when the DLLs have been previously entered at the corresponding parameters. If the file name is given without an extension then the extension is automatically assumed to be "DLL". If the existence of another file is to be checked then the file extension of this file must also be entered, e.g., "Userprogram.dat".

Contents of the "IND_DEV.INI" File

The configuration of the individual communication addresses and the settings of the various Bosch Rexroth devices are determined in this file. The "IND_DEV.INI" file is edited by the system configurator and is located in the "[LW]\...\CONFIG\" directory.

Bosch Rexroth system configurator

The Bosch Rexroth System Configurator is an editor that sets and lists the devices connected to the control PC. The device addresses, the device type and the description of the communication path to the device are used for this. The goal is to create a 1:1 copy of the device structure connected to the control PC; this structure is termed the system configuration.

The system configuration is stored on the control PC. Furthermore, the devices can also be assigned basic properties, e.g. a Type MTVNC device (virtual MTC for the function "Offline Simulation") can be assigned to a device of Type MTC200-P or MTCNC, etc., in order to form a simulation pair. Here, the parameter records of the real device can be used by the virtual device allowing a simulation of NC programs to be started.

Note: An online help is also included in the system configurator. It can be called up by pressing the <F1> function key while the program is running.

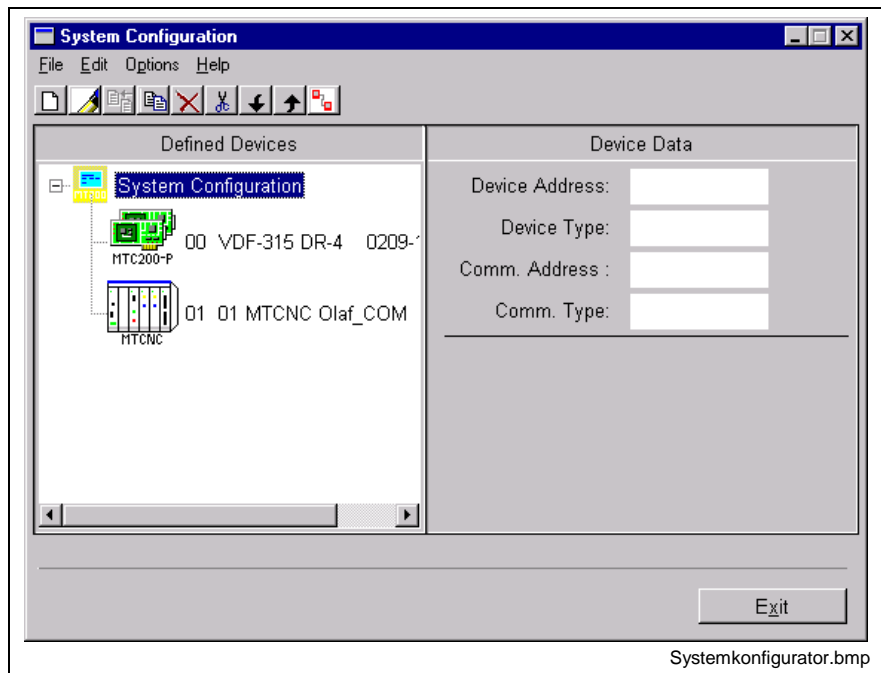


Fig. 6-4: Bosch Rexroth system configurator

The "IND_DEV.INI" file corresponds to the Microsoft Windows INI standard and is structured as follows:

Identifier	Values	Explanation
[CommAddrX]	X = 1...8	Assignment of the communication channel (thread) of the function interface.
CommStr=	V24, Port [COM1...4], baud rate, parity, type of interface, packet counter - or - DMA, address, offset, length - or - SHM, Channel No. [1...15]	Communication via RS232 serial interface, e.g. V24,COM1,19200,NONE,RS232,TCOM. Communication via RS485 serial interface, e.g. V24,COM2,19200,NONE,R485H,TCOM For communication via a dual port RAM, a DMA channel is also required for the MTC200-P, e.g., DMA,\$D000,\$0000,\$2000. Communication channel to the MTVNC via a shared memory, e.g. SHM, 1.
Timeout=	>= 1000 [msec] Preset = 3500 [msec]	! OPTIONAL ! Time in which a response must be received from the device.
- or – only for dual port RAM (e.g. for MTC200-P)		
CommStr=	DPR, address [\$C000,\$0000 ... \$FE00,\$0000], Length, RAM0, Packet-Counter	Communication via dual port-RAM, e.g., MTC200-P DPR, \$D000,\$0000,\$2000,RAM0,TCOFF.
PortAddr=	\$200, \$204, ... \$31C e.g., \$31C	Address of the MTC/MTS card according to the settings on the respective card.
PortVal=	\$20, \$21, ... \$3F e.g., \$28 at address [\$D000,\$0000]	Configuration byte for setting the physical memory address of the MTC/MTS card.
Timeout=	>= 1000 [msec] Preset = 3500 [msec]	! OPTIONAL ! Time in which a response must be received from the device.
[DeviceAddrX]	X = 0..0.15	device address
Component type1=	e.g., MTS-P01.2 NONE, MTS-P, MTS-P01.2, MTS-P02.2, MTS-R-M1, PPC-R	Name of the PLC component type
Component type2=	e.g., MTC-P-G2 NONE, MTC-P, MTC-R, PPC-R	Name of the NC component type
DeviceName=	Max. 32 ASCII characters	Device name; e.g., Processing Center 12T34

Identifier	Values	Explanation
DeviceType=	e.g. MTC200-P-G2 MTVNC MTC200-P-G2 MTC200-R-G2 ISP200-P-G2 ISP200-R-G2 TRA200-R ECODRIVE03 MTA200-P SERCANS-A SERCANS-P SYNTAX-P SYNTAX-R	Device type: MTC with PLC PC variant Virtual MTC MTC with PLC PC variant MTC with PLC RECO variant Standalone PLC PC variant Standalone PLC RECO variant TRANS 200 RECO variant Ecodrive03 MTA 200 control SERCANS-A card (via serial interface) SERCANS-P card (via serial interface) SYNTAX PC variant SYNTAX RECO variant
DeviceAssign=	0...15, NO	Assignment of a simulation pair. The MTVNC is, for example, hereby assigned to a real MTCNC.
DeviceStatus=	ON, OFF	Assignment of whether or not the device is incorporated into the management structure of the function interface.
MtvncMode=	OFF, RUN, STANDBY	! Only for virtual MTC (MTVNC) ! Status of the MTVNC with inactive utilization
MtvncMemory=	256, 257 .. 16383 [KB] Preset = 512 [KB]	! Only for virtual MTC (MTVNC) ! Size of the PC memory used by the MTVNC.
CommAddr=	1...8	Assignment of the communication address. Corresponds to the [CommAddr1...8] parameter.
PLC=	YES, NO	PLC support for the device. E.g. one MTVNC, TRANS200-R has no PLC, therefore the parameter PLC=NO is set.
[DeviceOrder]	This contains the configuration settings for the system configurator.	
Order=	0,1,2, ...15	Order in which the configured devices are displayed.
[NetManager]	This contains the configuration settings for the network driver "NETINTFC.EXE"	
NetManagerMode=	OFF, RUN	Starts the network device driver.

Example Entries in the "IND_DEV.INI" File

Entry	Explanation
[CommAddr1] CommStr=DPR,\$D000,\$0000,\$2000,RAM0,TCON PortAddr=\$31C PortVal=\$28	Communication address 1 Settings for communication via dual port RAM Port address of the MTC/MTS card Physical memory address of the MTC/MTS card.
[CommAddr2] CommStr=DMA,\$D000,\$0000,\$2000	Communication address 2 Assignment of the DMA channel.
[CommAddr3] CommStr=V24,COM1,19200,NONE,RS232,TCON	Communication address 3 Settings for communication via RS232.
[CommAddr4] CommStr=DPR,\$D200,\$0000,\$2000,RAM0,TCON PortAddr=\$318 PortVal=\$29	Communication address 4 Settings for communication via dual port RAM Port address of the MTC/MTS card Physical memory address of the MTC/MTS card.
[CommAddr5] CommStr=SHM,1	Communication address 5 Settings for communication via shared memory.

Entry	Explanation
[DeviceAddr0] CommAddr=1 Componenttype1= MTS-P-G2 Componenttype2= MTC-P-G2 DeviceAssign=NO DeviceName=VDF-315 DR-4 0209-15 DeviceStatus=ON DeviceType=MTC200-P PLC=YES	Device address 00 Assigned communication channel PLC components MTS-P-G2 CNC components MTC-P-G2 No MTVNC assigned Device name Device is available and ready for operation Device type PLC support
[DeviceAddr1] CommAddr=4 Componenttype1= MTS-P02.02 Componenttype2= MTC-P-G2 DeviceAssign=NO DeviceName= Processing center 12T35 DeviceStatus=ON DeviceType=MTC200-P PLC=YES	Device address 01 Assigned communication channel PLC components MTS-P02.02 CNC components MTC-P-G2 No MTVNC assigned Device name Device is available and ready for operation Device type PLC support
[DeviceAddr2] CommAddr=5 Componenttype1= NONE Componenttype2= NONE DeviceAssign=1 DeviceName= V-Processing center 12T34 DeviceStatus=ON DeviceType=MTVNC MtvncMemory=512 MtvncMode=RUN PLC=NO	Device address 02 Assigned communication channel PLC component not available CNC component not available Assigned to device address 01 (simulation pair) Device name Device is available and ready for operation Device type Size of the PC memory Status during inactive use No PLC support
[DeviceAddr3] CommAddr=3 Componenttype1= NONE Componenttype2= PPC-R DeviceAssign=NO DeviceName= TRANS200 DeviceStatus=ON DeviceType=TRANS200-R PLC=NO	Device address 03 Assigned communication channel PLC component not available CNC component PPC-R No MTVNC assigned Device name Device is ready for operation Device type No PLC support
[DeviceOrder] Order=3,0,1,2	Order in which the configured devices are displayed in the system configurator
[NetManager] NetManagerMode=RUN	Network driver is started
[BofManager] PollDeviceStatus=OFF PollDeviceStatusRate=4000 PollDeviceStatusCheckFactor=4	On switching on (ON), the device status of the devices is requested cyclically. The cycle time of a device request is controlled by this value. If a device can not be addressed then a request is no longer made until a time has passed that is the product of PollDeviceStatusRate multiplied by the PollDeviceStatusCheckFactor.

Contents of the "[LW]:\Winnt\System32\" System Directory

The following Microsoft class libraries are stored in the system directory of Windows NT "[LW]:\Winnt\System32\":

File	Explanation
MFC30.DLL	Microsoft class libraries
MSVCRT20.DLL	Microsoft class libraries
MFC40.DLL	Microsoft class libraries
MFC42.DLL	Microsoft class libraries
MSVCRT40.DLL	Microsoft class libraries
MSVCRT.DLL	Microsoft class libraries
MSVCP50.DLL	Microsoft class libraries
MSVBVM50.DLL	Microsoft class libraries
COMCTL32.OCX	Dialog elements for Visual Basic applications
COMDLG32.OCX	Dialog elements for Visual Basic applications
REGSVR32.EXE	Application for registering the OCX files

Contents of the "[LW]:\Winnt\System32\Drivers\" Driver Directory

The following files of the core-mode driver are stored in the driver directory of Windows NT "[LW]:\Winnt\System32\Drivers\":

File	Explanation
MTCNC001.SYS	Windows NT core-mode driver
MTCNC001.INI	Configuration file of the core-mode driver
REGINI.EXE	Application for registering the core-mode driver

Contents of the "[LW]:\...\MTGUI\BasicData\Help\" Directory

The drive as well as the path "[LW]:\..." are pre-set during the standard installation routine to "C:\Programme\Indramat\MTGUI". The following Windows 95/NT help files for the printed English and German manuals are stored in the "C:\...\MTGUI\BasicData\Help\[Language]" directory:

File	Explanation
FIVRS_DE.HLP	Windows 95/NT help file in German
FIVRS_DE.CNT	Definition file of the Windows 95/NT help file
FIVRS_EN.HLP	Windows 95/NT help file in English (in preparation)
FIVRS_EN.CNT	Definition file of the Windows 95/NT help file

Contents of the "[LW]:\...\MTGUI\BasicData\Resource" Directory

The drive as well as the path "[LW]:\..." are pre-set during the standard installation routine to "C:\Programme\Indramat\MTGUI\Bin". The following files are contained in the "C:\...\MTGUI\BasicData\Resource" directory:

File	Explanation
BOFINTFC.DAT	BOF process definition file
INDRAMAT.INI	File with global function interface settings
LOGINTFC.DAT	Definition file of the logic process
MECX.DAT	Definition file for the MECX device group
MISX.DAT	Definition file for the MWSX device group
MPCX.DAT	Definition file for the MPCX device group
MSCX.DAT	Definition file for the MSCX device group
MSYX.DAT	Definition file for the MSYX device group
MTAX.DAT	Definition file for the MWAX device group
MTCX.DAT	Definition file for the MWCX device group
VERSION.DAT	Definition file for the version ID

Example Entries in the "VERSION.DAT" File

The version ID of the individual parts of the program as well as the version of the function interface are entered in the "VERSION.DAT" file. This applies to all program parts (EXE, DLL) of the function interface. The following example shows the entries in this file:

Name	Build	Version	Date	Start Parameter
IFVERSION	113	04V00	Feb 22	
INDRAMAT.DLL	113	04.01	Feb 15	
INDFS100.DLL	113	03.14	Feb 22	
INDIF300.DLL	113	03.63	Feb 16	
INDUT140.DLL	113	03.09	Feb 22	
INDIF310.DLL	113	03.32	Feb 16	
INDOF160.DLL	113	03.15	Feb 22	
INDIF200.DLL	113	03.71	Feb 22	
BOFINTFC.EXE	113	05.16	Feb 16	/b=w/c=t
INDIF360.DLL	113	03.07	Feb 03	
LOGINTFC.EXE	113	04.00	Feb 22	/c=t +G10
COMINTFC.EXE	113	04.00	Feb 22	/c=t +G10
INDIF210.DLL	113	04.00	Feb 22	
INDIF330.DLL	113	03.30	Feb 16	
INDIF540	113	03.01	Feb 22	
INDIF130.DLL	113	03.16	Feb 22	
INDIF810.DLL	113	04.00	Feb 03	
INDIF350.DLL	113	03.35	Feb 03	
INDIF320.DLL	113	03.28	Feb 03	
INDIF340.DLL	113	03.31	Feb 03	

Contents of the "[LW]:\...\MTGUI\Bin" Directory

The drive as well as the path "[LW]:\..." are pre-set during the standard installation routine to "C:\Programme\Indramat\MTGUI". The following function libraries of the function interface are contained in the C:\...\MTGUI\Bin directory:

File	Explanation
BOFINTFC.EXE	BOF process
COMINTFC.EXE	Communication process
INDFS100.DLL	Processing the file ID
INDIF000.DLL	General functions for the user.
INDIF120.DLL	Functions for outputting the trace file.
INDIF130.DLL	Functions for the BOF process.
INDIF150.DLL	Functions for the logic and communication process.
INDIF200.DLL	Functions for the logic and communication process.
INDIF210.DLL	Functions for the logic and communication process.
INDIF220.DLL	Functions for the logic and communication process.
INDIF300.DLL	Functions for the BOF process.
INDIF310.DLL	Functions for the BOF process.
INDIF320.DLL	Functions for the DOS - Windows NT connections.
INDIF330.DLL	Functions for the BOF process.
INDIF340.DLL	Functions for the BOF process.
INDIF350.DLL	Functions for the DOS - Windows NT connections.
INDIF360.DLL	Functions for file access.
INDIF400.DLL	BOF process access to parameters.
INDIF500.DLL	Functions for access optimization.
INDIF510.DLL	Functions for access optimization.
INDIF520.DLL	Functions for access optimization.
INDIF530.DLL	Functions for access optimization.
INDIF540.DLL	Functions for access optimization.
INDIF550.DLL	Functions for access optimization.
INDIF560.DLL	Support for network connections and PLC and NC optimizer
INDIF570.DLL	Server for fast M-K communication
INDIF600.DLL	Functions for access to NC programs.
INDIF610.DLL	Functions for access to NC programs.
INDIF700.DLL	Functions for access to NC compiler.
INDIF800.DLL	Higher functions of all device groups.
INDIF810.DLL	Functions for access to the MWCX device groups.
INDIF820.DLL	Functions for access to the MSCX device groups.
INDIF830.DLL	Functions for access to the MECX device groups.
INDIF840.DLL	Functions for access to the MWAX device groups.
INDIF860.DLL	Functions for access to the MSYX device groups.
INDIF870.DLL	Functions for access to the MWSX device groups.

File	Explanation
INDIFA00.DLL	Functions for HMI support of the MWCX device groups.
INDIFZ00.DLL	Functions for access to the MWAX device groups.
INDMA900.DLL	Processing the MAP file
INDMA110.DLL	Connecting the MAP file
INDOF160.DLL	Using various system utilities
INDRAMAT.DLL	Access to global settings (GetInstPath, etc.)
INDUT140.DLL	Using various system utilities
KILLTASK.EXE	Application for terminating function interface applications (see Chapter 4 "Programming")
LOGINTFC.EXE	Logic process
NETINTFC.EXE	Application for connection of client/server
VBDemo.exe	Test program in Visual Basic
IFDemo.exe	Test program in Visual C++

7 Error Codes

7.1 General Error Result Line

If the "DataTransfer" routine returns an error code, then the requested data is not returned by the "ReadGroupItem" routine, but a general error result line is returned instead. This general error result line contains additional information regarding the possible causes of the error.

Note: How the routines work, as well as tips and tricks for working with the Bosch Rexroth function interface, is described in Chapter "Programming".

The following table shows the general structure of the error result line. One line is output consisting of 5 columns for the class of error, error code, expanded additional information, error text and additional text.

Line 1	Column 1	...	Column 5
--------	----------	-----	----------

Meaning of the Columns

- 1 = Error class
- 2 = NACK code or return error code (depends on error classes)
- 3 = Expanded additional information [hexadecimal LONG value]
- 4 = Error text [ASCII characters]
- 5 = Additional text [x= exists, -- = does not exist]

The following error classes are contained in the file "INDIF000.h" or "INDIF000.BAS":

Error class	Meaning
FI_ERROR_CLASS_NACK	NACK messages of control
FI_ERROR_CLASS_FCT	Error codes of FI protocol functions
FI_ERROR_CLASS_TEXT	ERROR TEXT is supplied
FI_ERROR_CLASS_WIN	WinNT error texts
FI_ERROR_CLASS_SERCOS	SERCOS error texts
FI_ERROR_CLASS_VM	VISUALMOTION error texts
FI_ERROR_CLASS_SIS	GENERAL SIS header error text

7.2 Error Codes 200 to 999

Code	Error Text	Name and Meaning of Error
201	BOF_INVALID_MTCNC_NUMBER	Invalid device address
202	BOF_NO_INST_PATH	No installation path found (Indramat.INI).
203	BOF_NO_MAP_FILE_FOUND	MAP file "PLCMAP.SPS" not found.
204	BOF_NO_MAP_FILE_NAME	No entry in the "PLCMAP.DAT" file has been found for the device address.
205	BOF_FILE_ERROR	File processing error.
206	BOF_VERSION_ERROR	More than the MAXIMUM NUMBER OF LINES contained in the "VERSION.DAT" file. Remedy: Delete "VERSION.DAT" file
207	BOF_MUTEX_ERROR	Error generating a MUTEX object.
208	BOF_FILE_MAPPING_ERROR	Error generating file mapping.
209	BOF_MEMORY_ERROR	Memory allocation error.
210	BOF_DATA_MAP_ERROR	DATA MAP access error.
211	BOF_MUTEX_TIMEOUT	MAP file access error.
212	BOF_DATA_LENGTH_ERROR	Data buffer is too small.
213	BOF_FILE_NOT_FOUND	File not found.
214	BOF_SYS_MAP_ERROR	SYSTEM MAP access error.
215	BOF_MAP_ELEMENT_ERROR	No valid MAP structure element.
216	BOF_INVALID_CHANNEL_ERROR	LOG channel number invalid.
217	BOF_TIMEOUT_ERROR	Pre-set timeout has expired.
218	BOF_SHMEM_ALREADY_EXIST	SHARED MEM already exists.
219	BOF_PROCESS_NOT_EXIST	Process addressed does not exist.
220	BOF_FILE_EOF	End of file reached.
221	BOF_EVENT_ERROR	Error generating an event object.
222	BOF_PROCESS_ALREADY_EXIST	Process to be started already running.
223	BOF_COMM_ADDRESS_ERROR	No valid communication address.
224	BOF_DEVICE_TYP_ERROR	No valid device type.
225	BOF_DEVICE_ERROR	No valid device address defined.
226	BOF_DEVICE_NAME_ERROR	Invalid device name.
227	BOF_DEVICE_STATUS_ERROR	No valid device status.
228	BOF_DEVICE_PLC_ERROR	No valid PLC information.
229	BOF_TASK_ID_ERROR	Invalid or false task ID.
230	BOF_TASK_ADM_ERROR	Task administration error.
231	BOF_TASK_TRIGGER_ERROR	Task trigger-event error.
232	BOF_EVENT_NOT_FOUND	Event object does not exist.
233	BOF_TASK_NAME_ERROR	Task name is too long.
234	BOF_SYS_STACK_INDEX_ERROR	Invalid SYS-MSG STACK INDEX
235	BOF_SYS_STACK_FULL_ERROR	SYS-MSG STACK is full.
236	BOF_SYS_STACK_MSG_ERROR	SYS-MSG message is not known in SYS-MSG STACK.

Code	Error Text	Name and Meaning of Error
237	BOF_SYS_STACK_ACCEPT_ERROR	SYS-MSG message could not be accepted by the SYS-MSG STACK within the pre-set time.
238	BOF_SYS_MSG_SET_ERROR	Access to SYS-MSG channel not possible in the pre-set time. (SYS-Message is issued).
239	BOF_SYS_MSG_GET_ERROR	Access to SYS-MSG channel not possible in the pre-set time (SYS-Message is fetched).
240	BOF_DATA_TIME_ERROR	A data element in the shared memory area was not released in the pre-set time.
241	BOF_DATA_ACCESS_ERROR	Access to a data element in the shared memory area is locked.
242	BOF_FCT_PAR_ERROR	An incorrect parameter value has been passed within the function.
243	BOF_SYS_STACK_QUIT_ERROR	SYS-MSG acknowledgement event has not been released in the pre-set time.
244	BOF_NO_SYS_MSG_RDY	No SYS-MSG message.
245	BOF_FORMAT_ERROR	Format error
246	BOF_SYS_MSG_LIST_ERROR	Error in the SYS-MSG list for manual acknowledgement.
247	BOF_NO_IFDLL_MODE_ERROR	Mode details missing in "IfDllMode=" entry in "INDRAMAT.INI" file.
248	BOF_LOG_GRP_COUNT_ERROR	Invalid group number for the LOG channel.
249	BOF_NO_SYS_MSG_CONF_ERROR	No SYS-MSG acknowledgement message.
250	BOF_NO_PATH_ENV_ERROR	No path environment variable.
251	BOF_LOGIN_IF_EVENT_ERROR	LOGINIF event could not be created.
252	BOF_LOGIN_SYS_MSG_ERROR	LOGINIF could not be carried out in the pre-set time due to pending SYS-MSGs.
253	BOF_LOGIN_EVENT_TIME_ERROR	No SYS-MSG could be issued because the login event took too long.
254	BOF_DLL_MUTEX_TIMEOUT_ERROR	Access to the DLL-lock Mutex was not possible in the pre-set time.
255	BOF_DLL_ALREADY_INSTALLED	DLL already installed.
256	BOF_DLL_LOAD_ERROR	DLL could not be loaded by the load library.
257	BOF_DLL_LIST_FULL_ERROR	DLL list already full.
258	BOF_DLL_LIST_DELETE_ERROR	DLL to be deleted is not in the list.
259	BOF_DOS_NT_SYS_MSG_ERROR	Invalid SYS-MSG message number in DOS → NT job processing.
260	BOF_DOS_NT_JOB_STR_ERROR	Invalid DOS → NT command string.
261	BOF_SYS_MSG_RANGE_ERROR	SYS-MSG message number is outside the valid number range.
262	BOF_DOS_NT_JOB_INFO_ERROR	DOS → NT command information is too long
263	BOF_DOS_NT_SYS_MSG_Q_ERROR	An odd SYS-MSG message number (acknowledgement) was passed by the DOS → NT command SYSC_XXX.
264	BOF_DOS_NT_FKT_NOT_FOUND_ERROR	DOS → NT command issued for which there is no processing function in the "BOFINTFC.DAT" file.
265	BOF_DOS_NT_DLL_NAME_NOT_FOUND_ERROR	No DLL name exists for DOS → NT commands in the "BOFINTFC.DAT" file.
266	BOF_DOS_NT_DLL_NOT_FOUND_ERROR	DLL for the DOS → NT commands not found.

Code	Error Text	Name and Meaning of Error
267	BOF_DOS_NT_FKT_NOT_IN_DLL_ERROR	DOS → NT processing function not found in the specified DLL.
268	BOF_DOS_NT_BOF_DAT_NOT_FOUND_ERROR	The "BOFINTFC.DAT" file could not be found.
269	BOF_TASK_NAME_NOT_FOUND_ERROR	Task name is not in the task list.
270	BOF_TASK_ID_NOT_FOUND_ERROR	No task exists for the task ID.
271	BOF_NT_CODE_ERROR	WIN-32 API error has occurred.
272	BOF_DOS_NT_PROCESS_PRIORITY_ERROR	Invalid process priority class.
273	BOF_DOS_TASK_NAME_ERROR	Error in generating the DOS-BOF task name.
274	BOF_PARENT_WIN_NAME_LEN_ERROR	Name of the parent window is too long.
275	BOF_TERMINATE_EVENT_NAME_LEN_ERROR	Name of the terminate event is too long.
276	BOF_PARENT_WIN_NOT_EXIST_ERROR	Registered task does not have a parent window.
277	BOF_DLL_NOT_EXIST_ERROR	DLL sought does not exist.
278	BOF_DLL_FUNCTION_NOT_FOUND_ERROR	Function sought does not exist in the specified DLL.
279	BOF_PROCESS_NOT_LOGIN_ERROR	An FI command has been called although the client is not yet logged in. The "LogInIf" routine has not yet been run.
280	BOF_DEVICE_HANDLE_ERROR	Device handle could not be generated.
281	BOF_DEVICE_ASSIGN_ERROR	There is no "DeviceAssign" entry in the "IND_DEV.INI" file, or the entry is invalid.
282	BOF_MEMORY_CLASS_ERROR	No valid memory class for DOS → NT command RDNT/WRNT.
283	BOF_MEMORY_DOS_NT_DATA_LEN_ERROR	Data length of RDNT/WRNT command is too long.
284	BOF_SHMEM_INDEX_ERROR	No valid SHARED MEMORY INDEX.
285	BOF_NO_PORT_ADDR_ERROR	No port address in the communication address.
286	BOF_NO_PORT_VALUE_ERROR	No port value in the communication address.
287	BOF_VRT_MANAGER_MODE_ERROR	Invalid VRT manager mode
288	BOF_VRT_START_MODE_ERROR	There is no MTVNC mode entry in the "IND_DEV.INI" file, or the entry is invalid.
289	BOF_PAR_MIN_ERROR	No PARMIN value in the "BOFINTFC.DAT" file for the FI command.
290	BOF_PAR_MAX_NUMBER_ERROR	Too many pass parameters in the FI command.
291	BOF_PAR_MAX_ERROR	No PARMAX value in the "BOFINTFC.DAT" file for the FI command.
292	BOF_PAR_NUMBER_ERROR	Number of FI parameters does not agree with the data in the DAT files (e.g.: MTCX.DAT, BOFINTFC.DAT, etc.)
293	BOF_PAR_DESCRIPTOR_ERROR	No parameter description.
294	BOF_PAR_TYPE_ERROR	Invalid FI parameter type.
295	BOF_PAR_TYPE_NOT_FOUND_ERROR	No FI parameter type description found.
296	BOF_PAR_DATA_ERROR	Invalid FI parameter data, i.e. FI data not defined in FI data type.
297	BOF_PAR_TYPE_DESCRIPTOR_ERROR	Invalid FI parameter type description.
298	BOF_PAR_INDEX_ERROR	FI parameter index is too large.
299	BOF_PAR_NO_CYCLIC_ERROR	Either there is no CYCLIC entry in the "BOFINTFC.DAT" file or the CYCLIC entry is invalid.
300	BOF_PAR_NO_CYCLIC_FI_COMMAND_ERROR	No cyclic FI command released.

Code	Error Text	Name and Meaning of Error
301	BOF_PAR_NO_BINAER_ERROR	Either there is no binary entry, or an invalid binary entry in the "BOFINTFC.DAT" file.
302	BOF_PAR_NO_BINAER_FI_COMMAND_ERROR	No binary operation of FI command released.
303	BOF_NT_DOS_CHANNEL_ACCESS_ERROR	Access to NT → DOS job channel not possible in the pre-set time.
304	BOF_NT_DOS_COMMAND_LENGTH_ERROR	NT → DOS command string is too long.
305	BOF_NT_DOS_COMMAND_INFO_LENGTH_ERROR	NT → DOS command info string is too long.
306	BOF_NT_DOS_TIMEOUT_ERROR	NT → DOS job could not be executed in the pre-set time.
307	BOF_NT_DOS_FKT_NOT_FOUND_ERROR	An NT → DOS command was issued that had not been declared in the "BOFINTFC.DAT" file.
308	BOF_NT_DOS_DLL_NAME_NOT_FOUND_ERROR	No DLL is declared in the "BOFINTFC.DAT" file for the NT→DOS command issued.
309	BOF_NT_DOS_DLL_NOT_FOUND_ERROR	DLL for the NT → DOS commands not found.
310	BOF_NT_DOS_FKT_NOT_IN_DLL_ERROR	NT → DOS processing function not found in the specified DLL.
311	BOF_NT_DOS_JOB_STR_ERROR	Invalid NT → DOS command string.
312	BOF_NT_DOS_JOB_INFO_ERROR	NT → DOS command information is too long
313	BOF_NT_DOS_DPR_TIMEOUT_ERROR	Access to NT-DOS-DPR memory not possible in the pre-set time.
314	BOF_NT_DOS_NO_COMMAND_ERROR	No NT → DOS command string.
315	BOF_NT_DOS_BOF_INDEX_ERROR	Invalid DOS-BOF INDEX when issuing an NT → DOS command.
316	BOF_PAR_INVALID_VALUE_ERROR	Pass parameter to the function has an invalid value.
317	BOF_DOS_BOF_EXE_PATH_ERROR	DOS-BOF EXE file must not contain details of path.
318	BOF_LOG_IN_LOG_OUT_TIMEOUT_ERROR	Login/Logout not possible in the pre-set time.
319	BOF_DEVICE_TYP_GROUP_ERROR	Selected device address does not exist in this device group.
320	BOF_INVALID_PROCESS_NUMBER_ERROR	Invalid NC process number
321	BOF_PROCESS_NAME_LENGTH_ERROR	Process name is too long or invalid.
322	BOF_PARAM_IDENT_REQUEST_ERROR	Invalid data was returned by the interface on requesting the parameters.
323	BOF_SWITCH_DEVICE_ERROR	An attempt was made to switch to a virtual MTC that is assigned to a real MTC.
324	BOF_DEVICE_TYPE_REQUEST_ERROR	Invalid data was returned on requesting the device type ID.
325	BOF_DEVICE_SPS_IDENT_ERROR	Invalid data was returned by the interface on requesting the long ID of the PLC MAP file.
326	BOF_INVALID_AXIS_NUMBER_ERROR	Invalid axis number received [1...32].
327	BOF_NO_GBOVERSION_ERROR	There is no "GBOVERSION=" entry in the "INDRAMAT.INI" file, or the entry is invalid.
328	BOF_NO_ACHSREF_TABLE_ERROR	Axis reference table error.
329	BOF_DEVICE_GROUP_ERROR	The device group for this job is invalid.
330	BOF_PROCESS_NOT_DEFINED	Process is not defined in the current parameters.
331	BOF_INVALID_DEVICE_GROUP_VALUE_ERROR	Invalid device group number.
332	BOF_INVALID_DEVICE_ID_STR_ERROR	Invalid device ID string.
333	BOF_INVALID_DEVICE_GROUP_STR_ERROR	Invalid device group string.

Code	Error Text	Name and Meaning of Error
334	BOF_FI_JOB_CLASS_ALREADY_RUN_ERROR	FI JOB already running.
335	BOF_FI_JOB_REQUEST_ERROR	No more FI JOBS possible.
336	BOF_FI_JOB_ID_ERROR	No valid FI JOB ID.
337	BOF_FI_JOB_NO_ID_FOUND_ERROR	No FI JOB ID found in the management structure.
338	BOF_FI_JOB_PROGRESS_TYPE_ERROR	Invalid request for the progress of an FI JOB.
339	BOF_FI_JOB_EXECUTE_FKT_NOT_FOUND_ERROR	Execute function for the FI JOB was not found in the specified DLL.
340	BOF_FI_JOB_ERROR_STRING_TO_LONG	FI JOB ERROR STRING is too long.
341	BOF_FI_JOB_TIMEOUT_ERROR	FI JOB could not be executed in the pre-set time.
342	BOF_FI_ERROR_STRING_TO_LONG	String for the general FI ERROR RESPONSE TELEGRAM (general error result line) is too long.
343	BOF_DOS_MANAGERPROG_NOT_READY_ERROR	DOS-NT manager program not running.
344	BOF_NT_DOS_ORDER_TO_LONG	NT → DOS job description is too long.
345	BOF_FILE_CLASS_OBJECT_INSTALL_ERROR	File class object for access to the BOF files could not be created.
346	BOF_FILE_DIAGOFF_NOT_FOUND_ERROR	BOF file "DIAGOFF.XXX" not found.
347	BOF_FILE_DIAGOFF_OPEN_ERROR	Error opening the "DIAGOFF.XXX" file.
348	BOF_SH_MEM_DIAGOFF_NOT_FOUND_ERROR	No SHARED MEMORY for DIAGOFFxxx found.
349	BOF_FILE_DIAGTAB_NOT_FOUND_ERROR	BOF file "DIAGTAB.XXX" not found.
350	BOF_FILE_READ_WITH_FS_CLASS_ERROR	Read error with FS classes.
351	BOF_FILE_DIAGTEXT_NOT_FOUND_ERROR	Diagnostic text file "STERRxx.YYY" not found.
352	BOF_FILE_STERR_FILE_CLOSE_ERROR	Diagnostic file "STERRxx.YYY" could not be closed.
353	BOF_FILE_STERR_FILE_OPEN_ERROR	Diagnostic text file "STERRxx.YYY" could not be opened.
354	BOF_FILE_STERR_FILE_POSITION_ERROR	File positioning in diagnostic text file "STERRxx.YYY" could not be carried out.
355	BOF_FILE_STERR_FILE_READ_ERROR	Read function of diagnostic text file "STERRxx.YYY" could not be carried out.
356	BOF_FILE_STERR_FILE_NOT_FOUND_ERROR	Diagnostic text file "STERRxx.YYY" not found.
357	BOF_FILE_DIAGTAB_POSITION_ERROR	File positioning in "DIAGTAB.xxx" could not be carried out.
358	BOF_FILE_STERR_FILE_TIMEOUT_ERROR	TIMEOUT when waiting for the MUTEX release for access to the STERR files.
359	BOF_TASK_THREAD_TRIGGER_INFO_TO_LONG	Additional information passed for the TASK THREAD triggering is too long.
360	BOF_TASK_THREAD_TRIGGER_TIMEOUT_ERROR	TIMEOUT of MUTEX release for access to the TASK-THREAD triggering.
361	BOF_FILE_SPRACHE_FILE_OPEN_ERROR	"SPRACHE.DAT" file could not be opened.
362	BOF_COMMAND_RESULT_DATA_TYPE_ERROR	A result data type that is not permissible (e.g. 00_BR_AMM1/2) was requested for an FI-command (BR...).
363	BOF_FILE_TEXT_FILE_NOT_FOUND_ERROR	Relevant TEXTxx.YY file does not exist.
364	BOF_FILE_TIND_FILE_NOT_FOUND_ERROR	Relevant TINDxx.YY file does not exist.
365	BOF_FILE_TIND_FILE_OPEN_ERROR	TINDxx.YY could not be opened.
366	BOF_TEXT_NUMBER_TO_LARGE_ERROR	Text number to be read from BOF text file is too large.

Code	Error Text	Name and Meaning of Error
367	BOF_FILE_TEXT_FILE_OPEN_ERROR	TEXTxx.YY could not be opened.
368	BOF_FILE_TEXT_FILE_POSITION_ERROR	File positioning in the text file "TEXTxx.YY" could not be carried out.
369	BOF_FILE_TEXT_FILE_READ_ERROR	Read function in the text file "TEXTxx.YY" could not be carried out.
370	BOF_DIAGNOSTIC_NUMBER_TO_LARGE_ERROR	Message number for CNC/PLC message system is too large.
371	BOF_FILE_SYSERI_NOT_FOUND_ERROR	BOF file "SYSERI.XXX" not found.
372	BOF_FILE_SYSERI_OPEN_ERROR	Error opening the "SYSERI.XXX" file.
373	BOF_FILE_SYSERI_POSITION_ERROR	File positioning in SYSERI.xxx could not be carried out.
374	BOF_SH_MEM_SYSERI_NOT_FOUND_ERROR	No SHARED MEMORY for SYSERI.xxx found.
375	BOF_FILE_SYSANW_NOT_FOUND_ERROR	Diagnosis text file SYSANW.YY is not available.
376	BOF_FILE_SYSANW_FILE_CLOSE_ERROR	Diagnostic text file SYSANW.YY could not be closed.
377	BOF_FILE_SYSANW_FILE_OPEN_ERROR	Diagnostic text file SYSANW.YY could not be opened.
378	BOF_FILE_SYSANW_POSITION_ERROR	File positioning in SYSANW.YY could not be carried out.
379	BOF_FILE_SYSANW_READ_ERROR	Read function in the diagnosis text file "SYSANW.YY" could not be carried out.
380	BOF_FILE_SYSANW_FILE_TIMEOUT_ERROR	TIMEOUT when waiting for the MUTEX release for access to the SYSANW.YY files.
381	BOF_FILE_TXERR_FILE_NOT_FOUND_ERROR	Relevant TXERR.YY file not found.
382	BOF_FILE_TXERI_FILE_NOT_FOUND_ERROR	Relevant TXERI.YY file not found.
383	BOF_FILE_TXERI_FILE_OPEN_ERROR	TXERI.YY could not be opened.
384	BOF_FILE_TXERR_FILE_OPEN_ERROR	TXERR.YY could not be opened.
385	BOF_FILE_TXERR_FILE_POSITION_ERROR	File positioning in the text file "TXERR.YY" could not be carried out.
386	BOF_FILE_TXERR_FILE_READ_ERROR	Read function in the text file "TXERR.YY" could not be carried out.
387	BOF_COMMAND_NOT_AVAILABLE_DLL_MODE	The requested FI command not available for the "IfDIIMode=" set in the "INDRAMAT.INI" file.
388	BOF_NO_PARAMETER_SET_IN_CONTROL	No valid parameter record in the control unit.
389	BOF_MTA200_COMMANDLINE_ERROR	No valid command line for the MTA 200 DRIVER.
390	BOF_FAR_DEVICE_STATUS_ERROR	No, or invalid, FARDEVICE entry.
391	BOF_DEVICE_PATH_ERROR	No, or invalid, device path entry.
392	BOF_DEVICE_PROTOCOL_ERROR	No, or invalid, device protocol entry.
393	BOF_DEVICE_IP_ERROR	No, or invalid, DEVICEIP entry.
394	BOF_DOS_NT_TASK_CHANNEL_TIMEOUT_ERROR	Access to DOS → NT job channel not possible in the pre-set time.
395	BOF_PROCESS_NAME_ERROR	A syntax error has been detected in the process name.
396	BOF_NETINTFC_MANAGER_MODE_ERROR	Invalid NETINTFC MANAGER MODE.
397	BOF_NET_MANAGER_STATUS_ERROR	Invalid NET MANAGER STATUS entered in the "IND_DEV.INI" file.
398	BOF_TERMINATE_EVENT_NOT_FOUND_ERROR	No terminate event found for the registered TASK.

Code	Error Text	Name and Meaning of Error
399	BOF_PARENT_WIN_ALREADY_EXIST_ERROR	PARENT WINDOW name already exists in the task management file.
400	BOF_NO_IFVERSION_ERROR	No "IfVersion=" entry exists in the "Indramat.INI" file.
401	BOF_NO_IFVERSION_ERROR	No IFVERSION entry in INDRAMAT.INI
402	BOF_NO_MTA200_INST_PATH	No MTA 200 installation path found.
403	BOF_SYSANW_FILTER_FILE_CREATE_ERROR	Filter file SYSSTW.XX for SYSANW.XX (SHORT MESSAGES only!) could not be created.
404	BOF_FILTER_FILE_DIRECTORY_CREATE_ERROR	The temporary sub-directory TEMPDATA could not be created for the data files of the small devices.
405	BOF_DELETE_FILE_ERROR	Data file (small devices) cannot be deleted.
406	BOF_TXERR_FILTER_FILE_CREATE_ERROR	Filter file TXEST.XX for TXERR.XX (SHORT MESSAGES only!) could not be created.
407	BOF_STERR_FILTER_FILE_CREATE_ERROR	Filter file STESTyy.XX for STERRyy.XX (SHORT MESSAGES only!) could not be created.
408	BOF_TXERR_FILTER_FILE_NOT_FOUND_ERROR	Filter file TXEST.XX for TXERR.XX (SHORT MESSAGES only!) does not exist in the temporary sub-directory TEMPDATA.
409	BOF_TXERR_FILTER_FILE_OPEN_ERROR	Filter file TXEST.XX for TXERR.XX (SHORT MESSAGES only!) could not be opened in the temporary sub-directory TEMPDATA.
410	BOF_TXEST_INDEX_FILE_CREATE_ERROR	INDEX file TXEST.XX (SHORT MESSAGES only!) could not be created.
411	BOF_BUFFER_LENGTH_ERROR	The PROCESSING BUFFER is too small for the data to be processed.
412	BOF_MSG_NUMBER_0_NOT_EXIST_ERROR	NO message number 0 exists in the message file.
413	BOF_MSG_NUMBER_TO_BIG_ERROR	Message number in message file is too big.
414	BOF_WRITE_FILE_ERROR	File could not be written.
415	BOF_SYSANW_FILTER_FILE_NOT_FOUND_ERROR	Filter file SYSSTW.XX for SYSANW.XX (SHORT MESSAGES only!) does not exist in the temporary sub-directory TEMPDATA.
416	BOF_SYSSTW_INDEX_FILE_CREATE_ERROR	Index file SYSSTW.XX (SHORT MESSAGES!) could not be created.
417	BOF_SYSANW_FILTER_FILE_OPEN_ERROR	Filter file SYSSTW.XX for SYSANW.XX (SHORT MESSAGES only!) could not be opened in the temporary directory TEMPDATA.
418	BOF_STERR_FILTER_FILE_NOT_FOUND_ERROR	Filter file STESTyy.XX for STERRyy.XX (SHORT MESSAGES only!) does not exist in the temporary directory TEMPDATA.
419	BOF_STESTYY_INDEX_FILE_CREATE_ERROR	The index file for STESTyy.XX (SHORT MESSAGES only!) could not be created.
420	BOF_STERR_FILTER_FILE_OPEN_ERROR	Filter file STESTYY.XX for STERRYY.XX (SHORT MESSAGES only!) could not be opened in the temporary sub-directory TEMPDATA.
421	BOF_WRONG_TELEGRAMM_CODE_ERROR	An INCORRECT TELEGRAM CODE has been returned by the control unit.
422	BOF_TXEST_INDEX_FILE_NOT_FOUND_ERROR	Index file TXESI.XX could not be found.
423	BOF_TXEST_INDEX_FILE_OPEN_ERROR	Index file TXESI.XX could not be opened.
424	BOF_TXEST_INDEX_FILE_READ_ERROR	Index file TXESI.XX could not be read.
425	BOF_SYSSTW_INDEX_FILE_NOT_FOUND_ERROR	Index file SYSSIW.XX could not be found.
426	BOF_SYSSTW_INDEX_FILE_OPEN_ERROR	Index file SYSSIW.XX is not open.

Code	Error Text	Name and Meaning of Error
427	BOF_SYSSTW_INDEX_FILE_READ_ERROR	Index file SYSSIW.XX could not be read.
428	BOF_STESTXX_INDEX_FILE_NOT_FOUND_ERROR	Index file STESIYY.XX could not be found.
429	BOF_STESTXX_INDEX_FILE_OPEN_ERROR	Index file STESIYY.XX could not be opened.
430	BOF_STESTXX_INDEX_FILE_READ_ERROR	Index file STESIYY.XX cannot be read.
431	BOF_DEVICE_TYPE_VALUE_TO_LARGE	DEVICE TYPE number is too large.
432	BOF_NOT_ENOUGH_MEMORY_IN_CONTROL	The required memory is not available in the selected slot number.
433	BOF_TXEST_KENNUNG_FILE_CREATE_ERROR	The ID FILE for TXEST.XX (SHORT MESSAGES only!) could not be created (TXESK.XX).
434	BOF_TXEST_KENNUNG_FILE_OPEN_ERROR	The ID FILE for TXEST.XX (SHORT MESSAGES only!) could not be opened (TXESK.XX).
435	BOF_TXEST_KENNUNG_FILE_READ_ERROR	The ID FILE for TXEST.XX (SHORT MESSAGES only!) could not be read (TXESK.XX).
436	BOF_SYSSTW_KENNUNG_FILE_CREATE_ERROR	The ID FILE for SYSSTW.XX (SHORT MESSAGES only!) could not be created (SYSSKW.XX).
437	BOF_SYSSTW_KENNUNG_FILE_OPEN_ERROR	The ID FILE for SYSSTW.XX (SHORT MESSAGES only!) could not be opened (SYSSKW.XX).
438	BOF_SYSSTW_KENNUNG_FILE_READ_ERROR	The ID FILE for SYSSTW.XX (SHORT MESSAGES only!) could not be read (SYSSKW.XX).
439	BOF_STESK_KENNUNG_FILE_CREATE_ERROR	The ID FILE for STESTxx.YY (SHORT MESSAGES only!) could not be created (STESKxx.YY).
440	BOF_STESK_KENNUNG_FILE_OPEN_ERROR	The ID FILE for STESTxx.YY (SHORT MESSAGES only!) could not be opened (STESKxx.YY).
441	BOF_STESK_KENNUNG_FILE_READ_ERROR	The ID FILE for STESTxx.YY (SHORT MESSAGES only!) could not be read (STESKxx.YY).
442	BOF_COMPONENT_TYPE_STR_TO_LARGE	The component string in IND_DEV.INI is too large.
443	BOF_INVALID_COMPONENT_NUMBER_ERROR	Invalid component number.
444	BOF_DEVICE_COMPONENT_TYPE_REQUEST_ERROR	INVALID DATA was returned by the interface on requesting the DEVICE COMPONENT TYPES.
445	BOF_DEVICE_DAT_FILE_NOT_FOUND_ERROR	Relevant DEVICE-DAT file not found for the BOF configuration.
446	BOF_MAIN_MENU_ITEM_ERROR	Invalid GUI main menu item.
447	BOF_MAIN_DEF_FILE_CONTENT_ERROR	BOF configuration file \MT_TEXTE\MAIN_DEF.INI not entered in sought device type.
448	BOF_DEVICE_INI_FILE_NOT_FOUND_ERROR	Relevant DEVICE-INI file not found for the BOF configuration.
449	BOF_DEVICE_INI_FILE_SYNTAX_ERROR	Format error in DEVICE-INI file for the BOF configuration.
450	BOF_DEVICE_POLLING_STATUS_ERROR	NO, or invalid, PollDeviceStatus in IND_DEV.INI.
451	BOF_DEVICE_POLLING_RATE_ERROR	NO, or invalid, PollDeviceStatusRate in IND_DEV.INI.
452	BOF_DEVICE_POLLING_CHECK_FACTOR_ERROR	NO, or invalid, PollStatusCheckFactor in IND_DEV.INI.
453	BOF_DOS_BOF_EXE_SYNTAX_ERROR	NO "_" character may be included in DOS-BOF-EXE file names (WITH TSR connection).
454	BOF_DOS_BOF_EXE_CMDLINE_SYNTAX_ERROR	NO "_" character may be included in the call parameters for the DOS-BOF-EXE (WITH TSR connection).
455	BOF_SYS_MSG_LENGTH_ERROR	The additional information for the SYS message is too long.

Code	Error Text	Name and Meaning of Error
456	BOF_DEVICE_STATUS_INFO_ERROR	More than one "critical" condition is managed in the DEVICE-STATUS INFO (SYSTEM MAP) e.g.: parameter download.
457	BOF_SYS_MSG_HOOK_LIST_TIMEOUT_ERROR	The SYS-MSG HOOK LIST cannot be accessed within the pre-set time.
458	BOF_PROCESS_LOGOUT_TIMEOUT_NETINTFC	NETINTFC has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
459	BOF_PROCESS_LOGOUT_TIMEOUT_DESKTOP	DESKTOP has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
460	BOF_PROCESS_LOGOUT_TIMEOUT_CONTROLDATA	CONTROL DATA has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
461	BOF_PROCESS_LOGOUT_TIMEOUT_LOGDBCOM	LOGDBCOM has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
462	BOF_PROCESS_LOGOUT_TIMEOUT_MPI	MPI has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
463	BOF_PROCESS_LOGOUT_TIMEOUT_BOFINTFC	BOFINTFC has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
464	BOF_IF_DLL_MODE_TO_SMALL	IF-DLL MODE set is too small for the function to be executed.
465	BOF_WATCH_LIST_OVERRUN_ERROR	NO WATCH LIST available (overrun) for the selected device.
466	BOF_INVALID_WATCH_LIST_NUMBER_ERROR	INVALID WATCH LIST NUMBER for the selected DEVICE.
467	BOF_NO_SYSTEM_ERRORTXT_ADM	There is NO management system for access to the SYSTEM ERROR TEXTS (SYSANW.XX)
468	BOF_NO_TX_ERRORTXT_ADM	There is NO management system for access to the TRANSMISSION ERROR TEXTS (TXERR.XX)
469	BOF_NO_MECH_ERRORTXT_ADM	There is NO management system for access to the MECHANISM ERROR TEXTS (STERRyy.XX)
470	BOF_INVALID_PLC_TYPE	An INVALID PLC type has been detected for the selected device.
471	BOF_SHMEM_INTERN_INDEX_ERROR	Maximum INTERNAL SHM INDEX is too big.
472	BOF_SHMEM_INDEX_TO_BIG_ERROR	SHM INDEX is too big.
473	BOF_INTERNAL_PROCESS_NUMBER_ERROR	NO INTERNAL PROCESS NUMBER (MT-NCPROZESSE : 0..6).
474	BOF_AXIS_NUMBER_NOT_DEFINED_ERROR	Requested AXIS NUMBER is not defined in the ACTUAL PARAMETER RECORD.
475	BOF_NO_PARAMETER_DOWNLOAD_FILE_EXIST	NO PARAMETER DOWNLOAD FILE available.
476	BOF_PARAMETER_DOWNLOAD_FILE_LOAD_ERROR	The PARAMETER DOWNLOAD FILE CANNOT be loaded.
477	BOF_PARAMETER_DOWNLOAD_FILE_ID_ERROR	A LENGTH ERROR has occurred in the various parameter IDs OR necessary entries are missing.
478	BOF_ALLOCATE_MEMORY_ERROR	NO memory could be allocated.
479	BOF_PARAMETER_DOWNLOAD_FILE_INDEX_ERROR	Max. DATA INDEX ENTRY in the PARAMETER DOWNLOAD FILE is NOT available.

Code	Error Text	Name and Meaning of Error
480	BOF_PARAMETER_DOWNLOAD_DATA_TO_LONG	An individual PARAMETER DATA STRING to be passed is too long for the telegram.
481	BOF_PARAMETER_IDENTIFICATION_NOT_EXIST	NO [ID_PARAMETER] section exists in the PARAMETER DOWNLOAD FILE.
482	BOF_SYSTEM_PARAMETER_IDENTIFICATION_NOT_EXIST	NO [ID_SYSTEM] section exists in the PARAMETER DOWNLOAD FILE.
483	BOF_SYSTEM_PARAMETER_DATA_NOT_EXIST	NO [DATA_SYSTEM] section exists in the PARAMETER DOWNLOAD FILE.
484	BOF_PROCESS_PARAMETER_DATA_NOT_EXIST	NO [DATA_PROCESSx] section exists in the PARAMETER DOWNLOAD FILE.
485	BOF_AXIS_PARAMETER_DATA_NOT_EXIST	NO [DATA_AXISx] section exists in the PARAMETER DOWNLOAD FILE.
486	BOF_INVALID_PARAMETER_DATA	The control unit has detected INVALID PARAMETER DATA during PARAMETER DOWNLOAD.
487	BOF_PARAMETER_DATA_NOT_COMPLETE	INCOMPLETE PARAMETER DATA has been passed during a PARAMETER DOWNLOAD.
488	BOF_NECESSARY_FUNCTION_NOT_FOUND	The necessary FUNCTIONS are NOT contained in the selected DLL, or the DLL is not found.
489	BOF_PARAMETER_DATA_LINE_TO_MUCH	The NUMBER of DATA LINES in the PARAMETER DOWNLOAD FILE is too great.
490	BOF_PARAMETER_MAX_ALLOCATE_MEMORY_ERROR	MEMORY REQUIREMENT for the INTERNAL DATA STRUCTURE during PARAMETER DOWNLOAD is too great.
491	BOF_PARAMETER_DOWNLOAD_BREAK_ERROR	A PARAMETER DOWNLOAD PROCEDURE has been interrupted by BREAK-INFO.
492	BOF_PARAMETER_UPLOAD_FILE_ALREADY_EXIST	PARAMETER UPLOAD FILE already exists.
493	BOF_PARAMETER_IDENT_SECTION_CREATE_ERROR	NO [ID_PARAMETER] section could be created in the PARAMETER UPLOAD FILE.
494	BOF_PARAMETER_UPLOAD_DATA_WRITE_ERROR	NO UPLOAD DATA could be written in the PARAMETER UPLOAD FILE.
495	BOF_SYSTEM_PARAMETER_IDENT_SECTION_CREATE_ERROR	NO [ID_SYSTEM] section could be created in the PARAMETER UPLOAD FILE.
496	BOF_DATA_SYSTEM_SECTION_CREATE_ERROR	NO [DATA_SYSTEM] section could be created in the PARAMETER UPLOAD FILE.
497	BOF_MAX_INDEX_DATA_SECTION_CREATE_ERROR	NO [MAX_INDEX_DATA] section could be created in the PARAMETER UPLOAD FILE.
498	BOF_ID_PROCESS_SECTION_CREATE_ERROR	NO [ID_PROCESSx] section could be created in the PARAMETER UPLOAD FILE.
499	BOF_DATA_PROCESS_SECTION_CREATE_ERROR	NO [DATA_PROCESSx] section could be created in the PARAMETER UPLOAD FILE.
500	BOF_ID_AXIS_SECTION_CREATE_ERROR	NO [ID_AXISx] section could be created in the PARAMETER UPLOAD FILE.
501	BOF_SAVE_ARRAY_PROCESSING_ERROR	An error occurred when processing SAVE ARRAYS.
502	BOF_COM_INTERFACE_REQUEST_ERROR	Requested COM INTERFACE could NOT be returned.
503	BOF_COM_DIAG_SERVER_INIT_ERROR	An error occurred when INITIALIZING THE COM DIAG TEXT SERVER.
504	BOF_CO_INITIALIZE_ERROR	A co-initialize procedure has NOT been carried out in the user process.
505	BOF_COM_DIAG_TEXT_ACCESS_ERROR	An error occurred fetching a message text.

Code	Error Text	Name and Meaning of Error
506	BOF_COM_INTERFACE_DIAG_SERVER_NULL_ERROR	COM INTERFACE POINTER for accessing the DIAG SERVER is ZERO.
507	BOF_OBJECT_CREATE_ERROR	NO OBJECT could be generated – GENERAL CREATE ERROR.
508	BOF_LANGUAGE_CONVERT_ERROR	A CONVERSION ERROR has occurred during the various conversions regarding language management.
509	BOF_NO_WRITE_DATA_AVAILABLE_ERROR	NO write value is available when calling the BW command.
510	BOF_TO_MANY_SPS_VARIABLES_ERROR	Too many PLC variables were transferred when calling up the BW-MKT command.
511	BOF_COM_INTERFACE_LANGUAGE_SERVER_NULL_ERROR	COM INTERFACE POINTER for accessing the LANGUAGE SERVER is ZERO.
512	BOF_COM_CURRENT_LANGUAGE_ACCESS_ERROR	Language extension CURRENTLY set could not be determined via LANGSUPP.
513	BOF_COM_CURRENT_CONTEXT_NAME_ERROR	The CURRENT CONTEXT NAME could NOT be determined.
514	BOF_SYS_MSG_NUMBER_RANGE_ERROR	SYS-MESSAGE number is NOT within the permissible value range – ONLY EVEN NUMBERS are permitted.
515	BOF_IF_STARTUP_TIMEOUT_ERROR	STARTUP of the BOFINTEFC.EXE exceeds the max. permissible startup time.
516	BOF_TCP_IP_OBJECT_CREATE_ERROR	No further TCP-IP communication object can be created. The maximum number has already been opened.
517	BOF_TCP_IP_OBJECT_NOT_EXIST_ERROR	The TCP-IP communication object addressed does NOT exist.
518	BOF_SYSTEM_PARAMETER_AXIS_INFO_ERROR	The AXIS INFORMATION (AXIS TYPE, APR NUMBER) could NOT be converted in the parameter download file – FORMATTING ERROR!
519	BOF_NO_MACHINE_DATA_DOWNLOAD_FILE_EXIST	MACHINE DATA DOWNLOAD FILE does not exist.
520	BOF_MACHINE_DATA_DOWNLOAD_FILE_LOAD_ERROR	The MACHINE DATA DOWNLOAD FILE CANNOT be loaded.
521	BOF_MACHINE_DATA_IDENTIFICATION_NOT_EXIST	NO [ID_MACHINE_DATA] section exists in the MACHINE DATA DOWNLOAD FILE
522	BOF_MACHINE_DATA_DOWNLOAD_FILE_ID_ERROR	A LENGTH ERROR has occurred in the various machine data IDs OR necessary entries are missing.
523	BOF_MACHINE_DATA_TYPEDEF_INFORMATION_NOT_EXIST	[TYPE_DEFINITION_INFO] section does not exist in the MACHINE DATA DOWNLOAD FILE.
524	BOF_MACHINE_DATA_TYPEDEF_INFORMATION_NOT_EXIST	[TYPE_DEFINITION_INFO] section does not exist in the MACHINE DATA DOWNLOAD FILE.
525	BOF_MACHINE_DATA_TYPEDEF_INFORMATION_ERROR	A FORMATTING ERROR has occurred in the [TYPE_DEFINITION_INFO] section in the MACHINE DATA DOWNLOAD FILE.
526	BOF_MACHINE_DATA_TO_MUCH_TYPEDEF_ERROR	Too many TYPE DEFINITIONS in the MACHINE DATA DOWNLOAF FILE.
527	BOF_MACHINE_DATA_TO_MUCH_TYPEDEF_ERROR	Too many TYPE DEFINITIONS in the MACHINE DATA DOWNLOAF FILE.
528	BOF_MACHINE_DATA_PAGE_INFO_ERROR	NO [PAGE_INFO] section present in the MACHINE DATA DOWNLOAD FILE, or there is a formatting error.

Code	Error Text	Name and Meaning of Error
529	BOF_MACHINE_DATA_TO_MUCH_PAGDEF_ERROR	Too many PAGE DEFINITIONS present in the MACHINE DATA DOWNLOAD FILE, OR the PAGE NUMBER is too high.
530	BOF_MACHINE_DATA_PAGE_DEFINITION_NOT_EXIST	NO [PAGE_DEFINITION_XXX] section exists in the MACHINE DATA DOWNLOAD FILE, although a relevant [ID_PAGE_DEFINITION_XXX] section has been defined.
531	BOF_MACHINE_DATA_PAGE_DEFINITION_ERROR	A FORMATTING ERROR has occurred in the [PAGE_DEFINITION_XXX] section in the MACHINE DATA DOWNLOAD FILE.
532	BOF_MACHINE_DATA_ELEMENT_DESCRIPTION_NOT_EXIST	The [PAGE_DESCRIPTION_XXX_YYY] section which should be present based on the PAGE DEFINITION does NOT exist in the MACHINE DATA DOWNLOAD FILE – NO DATA ELEMENT DESCRIPTION exists.
533	BOF_MACHINE_DATA_ELEMENT_DESCRIPTION_ERROR	A FORMATTING ERROR has occurred in the [PAGE_DESCRIPTION_XXX_YYY] section in the MACHINE DATA DOWNLOAD FILE.
534	BOF_MACHINE_DATA_PAGE_INFORMATION_ERROR	A FORMATTING ERROR has occurred in the [PAGE_DATA_INFO] section in the MACHINE DATA DOWNLOAD FILE or necessary entries are missing.
535	BOF_MACHINE_DATA_PAGE_NOT_EXIST	NO [PAGE_DATA_XXX] section exists in the MACHINE DATA DOWNLOAD FILE
536	BOF_MACHINE_DATA_ELEMENT_INFORMATION_ERROR	A FORMATTING ERROR or logic error has occurred in the [PAGE_DATA_ELEMENTS_XXX] section in the MACHINE DATA DOWNLOAD FILE.
537	BOF_MACHINE_DATA_PAGE_LINE_ERROR	Either the requested data line does NOT exist in the [PAGE_DATA_XXX] section in the MACHINE DATA DOWNLOAD FILE, OR there is a formatting error in the data line.
538	BOF_MACHINE_DATA_VALUE_STRING_CONVERT_ERROR	A MACHINE DATA STRING that is to be written CANNOT be converted – FORMATTING ERROR, or a formatting error has occurred in the parameter string of the MDS command.
539	BOF_MACHINE_DATA_VALUE_RANGE_ERROR	A VALUE RANGE ERROR has occurred when converting the MACHINE DATA STRING.
540	BOF_MACHINE_DATA_STRUCT_TO_LARGE_ERROR	A DATA STRUCTURE that is too large is present in the MACHINE DATA that is to be written – this CANNOT be written as a COMPLETE DATA STRUCTURE.
541	BOF_MACHINE_DATA_PAGE_INFO_NOT_EXIST	NO [PAGE_INFO] section exists in the MACHINE DATA DOWNLOAD FILE
542	BOF_MACHINE_DATA_VALUE_STRING_TO_LONG	A data string that is TOO LARGE exists in the [PAGE_DATA_XXX] section in the MACHINE DATA DOWNLOAD FILE (max. 50 characters possible).
543	BOF_MACHINE_DATA_DOWNLOAD_BREAK_ERROR	A MACHINE DATA DOWNLOAD PROCEDURE has been interrupted by BREAK-INFO.
544	BOF_MACHINE_DATA_VALUE_TO_MUCH_ERROR	Too many MACHINE DATA VALUES to be written are entered in the download file, or too many MACHINE DATA VALUES have been indicated in the MDS command.
545	BOF_GLOBAL_DATA_BUFFER_INDEX_TO_LARGE	The GLOBAL DATA BUFFER INDEX is outside the permissible range.
546	BOF_GLOBAL_DATA_BUFFER_CONTENT_TO_LARGE	Too many BYTES are to be copied into the GLOBAL DATA BUFFER.

Code	Error Text	Name and Meaning of Error
547	BOF_MACHINE_DATA_UPLOAD_FILE_ALREADY_EXIST	PARAMETER UPLOAD FILE already exists.
548	BOF_NO_MACHINE_DATA_SET_IN_CONTROL	NO MACHINE DATA RECORD exists in the CONTROL.
549	BOF_MACHINE_DATA_UPLOAD_BREAK_ERROR	A MACHINE DATA UPLOAD PROCEDURE has been interrupted by BREAK-INFO.
550	BOF_MACHINE_DATA_ELEMENT_NUMBER_TO_LARGE	The pre-set MACHINE DATA ELEMENT NUMMER is outside the valid range [1..110].
551	BOF_MACHINE_DATA_ELEMENT_COMPUTING_ERROR	A computation error has occurred in computing the necessary number of telegrams for reading the machine data.
552	BOF_NO_TOOL_MANAGEMENT_EXIST	NO tool management has been switched on in the system parameters or process parameters.
553	BOF_NO_DIAG_SERVER_AVAILABLE	NO DIAG SERVER available.
554	BOF_UNKNOWN_MESSAGE_TYPE	An unknown message type has occurred during message download for the small devices via the MFD command.
555	BOF_MESSAGE_DOWNLOAD_BREAK_ERROR	A MESSAGE DOWNLOAD PROCEDURE has been interrupted by BREAK-INFO.
556	BOF_MACHINE_DATA_IDENT_SECTION_CREATE_ERROR	NO [ID_MACHINE_DATA] section could be created in the MACHINE DATA UPLOAD FILE.
557	BOF_MACHINE_DATA_UPLOAD_DATA_WRITE_ERROR	NO UPLOAD DATA could be written in the MACHINE DATA UPLOAD FILE.
558	BOF_MACHINE_DATA_ID_TYPE_SECTION_CREATE_ERROR	NO [ID_TYPE_DEFINITION] section could be created in the MACHINE DATA UPLOAD FILE.
559	BOF_MACHINE_DATA_TYPE_INFO_SECTION_CREATE_ERROR	NO [TYPE_DEFINITION_INFO] section could be created in the MACHINE DATA UPLOAD FILE.
560	BOF_MACHINE_DATA_TYPE_DEF_SECTION_CREATE_ERROR	NO [TYPE_DEFINITION_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
561	BOF_MACHINE_DATA_PAGE_INFO_SECTION_CREATE_ERROR	NO [PAGE_INFO] section could be created in the MACHINE DATA UPLOAD FILE.
562	BOF_MACHINE_DATA_PAGE_ID_SECTION_CREATE_ERROR	NO [ID_PAGE_DEFINITION_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
563	BOF_MACHINE_DATA_PAGE_DEF_SECTION_CREATE_ERROR	NO [PAGE_DEFINITION_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
564	BOF_MACHINE_DATA_PAGE_DESCRIPTION_SECTION_CREATE_ERROR	NO [PAGE_DESCRIPTION_XXX_YYY] section could be created in the MACHINE DATA UPLOAD FILE.
565	BOF_MACHINE_DATA_PAGE_DATA_INFO_SECTION_CREATE_ERROR	NO [PAGE_DATA_INFO] section could be created in the MACHINE DATA UPLOAD FILE.
566	BOF_MACHINE_DATA_PAGE_DATA_ELEMENT_SECTION_CREATE_ERROR	NO [PAGE_DATA_ELEMENTS_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
567	BOF_MACHINE_DATA_PAGE_DATA_SECTION_CREATE_ERROR	NO [PAGE_DATA_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
568	BOF_LOGIN_COMINTFC_ERROR	An attempt has been made to log in the COMINTFC PROZESS ROOM.
569	BOF_LOGIN_LOGINTFC_ERROR	An attempt has been made to log in the LOGINTFC PROZESS ROOM.
570	BOF_PARAMETER_UPLOAD_BREAK_ERROR	A PARAMETER UPLOAD PROCEDURE has been interrupted by BREAK-INFO.
571	BOF_HOMATIC_DRIVER_DLL_NOT_FOUND_ERROR	The HOMATIC DRIVER DLL (INDIFY00.DLL) could NOT be found.

Code	Error Text	Name and Meaning of Error
572	BOF_HOMATIC_DRIVER_DLL_COULD_NOT_BE_LOAD_ERROR	The HOMATIC DRIVER DLL (INDIFY00.DLL) could NOT be loaded.
573	BOF_HOMATIC_DRIVER_DLL_FUNCTION_LOAD_ERROR	The HOMATIC DRIVER DLL (INDIFY00.DLL) does NOT contain ALL THE NECESSARY functions.
574	BOF_TIME_DATE_SET_STATUS_ERROR	NO, or invalid, TimeDateSetStatus entry in IND_DEV.INI.
575	BOF_TIME_DATE_SET_STATUS_RATE_ERROR	NO, or invalid, TimeDateSetStatusRate entry in IND_DEV.INI.
576	BOF_NO_PCL_PROGRAMM_IN_CONTROL	NO valid PLC program exists in the control.
577	BOF_FI_START_DISPLAY_ERROR	Invalid FIStartDisplay entry in IND_DEV.INI.
578	BOF_FI_START_DISPLAY_MODE_ERROR	NO, or invalid, FIStartDisplayExtendedMode entry in IND_DEV.INI.
579	BOF_INVALID_DEVICE_SIMULATION_ERROR	Invalid DeviceSimulation entry in IND_DEV.INI.
580	BOF_NO_SIMULATION_DEVICE_TYPE_EXIST_ERROR	NO SIMULATION is possible for the pre-selected device address.
581	BOF_SOFT_MONITOR_START_MODE_ERROR	Invalid SoftMonitorMode entry in IND_DEV.INI.
582	BOF_SIMISP_START_MODE_ERROR	Invalid SimIspMode entry in IND_DEV.INI.
583	BOF_SEKTION_NOT_FOUND_ERROR	The desired SECTION NAME is not AVAILABLE in the file to be modified
584	BOF_SIMTRA_START_MODE_ERROR	Invalid SimTraMode entry in IND_DEV.INI.
585	BOF_DESTINATION_FILE_ALREADY_EXIST_ERROR	The destination file already exists on file copying
586	BOF_PARAMETER_OFFLINE_FILE_LOAD_ERROR	The File ParamOff.dat cannot be loaded
587	BOF_PARAMETER_OFFLINE_FILE_ID_ERROR	A LENGTH ERROR has occurred in the various parameter IDs OR necessary entries are missing.
588	BOF_PARAMETER_ONLINE_FILE_LOAD_ERROR	The File ParamOn.dat cannot be loaded
589	BOF_PARAMETER_ONLINE_FILE_ID_ERROR	A LENGTH ERROR has occurred in the various parameter IDs OR necessary entries are missing.
590	BOF_PROCESS_TYPE_ERROR	Invalid process type
591	BOF_PARAMETER_DATA_NOT_FOUND_ERROR	In the parameter data file, necessary parameter data are not available or not correct
592	BOF_DEBUG_DEVICE_MODE_ERROR	Invalid DeviceDebugMode entry in IND_DEV.INI.
593	BOF_FI_COMMAND_LENGTH_ERROR	The FI command string is too long
594	BOF_GENERAL_FILE_NOT_FILE_ERROR	General: file NOT found
595	BOF_FI_COMMAND_DEVICE_STATUS_ERROR	There is an invalid DeviceStatus for the FI command
596	BOF_OPERATING_SYSTEM_NOT_SUPPORTED_ERROR	The operating system is NOT supported by FI.
597	BOF_SOFTMONITOR_SHUTDOWN_ERROR	Invalid SoftMonitorShutDown entry in IND_DEV.INI.
598	BOF_STARTED_PROCESS_TERMINATE_ERROR	The PROCESS to be started from FI BEFORE the READY MESSAGE
599	BOF_COMM_ADDRESS_TYPE_ERROR	The communication type is invalid
600	BOF_INVALID_AXIS_MEANING_INFO	The axis meaning is invalid
601	BOF_PAR_MIN_NUMBER_ERROR	Necessary PARAMETERS in the FI requirement command are not available
602	BOF_IND_DEV_READ_ERROR	IND_DEV.INI can NOT be read in with the PROFILE CLASS
603	BOF_PROFILE_SECTION_NOT_FOUND_ERROR	The searched SECTION is NOT available in the profile

Code	Error Text	Name and Meaning of Error
604	BOF_FI_ERROR_TEXT_INFO_ERROR	Invalid FiErrorTextInfo entry in IND_DEV.INI.
605	BOF_DEVICE_ADDRESS_CONSISTENT_ERROR	The selected DEVICE ADDRESS in the FI command does NOT correspond to the DEVICE ADDRESS in the binary telegram.
606	BOF_WRONG_DEVICE_PROTOCOL_ERROR	A false data protocol was chosen for the selected DEVICE ADDRESS
607	BOF_TOOL_POSITION_TOO_LARGE_ERROR	The information on max. tool storage is too large according to the process parameters.
608	BOF_FI_COMMAND_STACK_FULL_ERROR	There is NO more free space in the FI command stack
609	BOF_FI_COMMAND_STACK_SOURCE_DATA_LENGTH_ERROR	The SOURCE data is too long for the FI command stack
610	BOF_FI_COMMAND_STACK_RESULT_DATA_LENGTH_ERROR	The RESULT data is too long for the FI command stack
611	BOF_FI_COMMAND_STACK_INDEX_OUT_OF_RANGE_ERROR	Invalid FI command stack index – out of range
612	BOF_FI_COMMAND_STACK_ERROR	Invalid FiCommandStack entry in IND_DEV.INI.
613	BOF_FI_COMMAND_STACK_RATE_ERROR	NO, or invalid, FiCommandStackRate entry in IND_DEV.INI.
614	BOF_DEVICE_POLLING_OFF_ERROR	NO device polling is switched on in IND-DEV.INI
615	BOF_TIMEOUT_FOR_FURTHER_INFO_FROM_DEVICE_ERROR	In the selected delay time, the controller did NOT supply the additional information
616	BOF_ERROR_TEXT_NUMBER_CONVERT_ERROR	The error number can NOT be converted through sscanf()
617	BOF_ERROR_TEXT_NOT_FOUND_ERROR	The error number can NOT be resolved in an error text
618	BOF_MACHINE_DATA_PAGE_SIZE_ERROR	A defined PAGE is larger than 64Kbyte
619	BOF_RECEIVED_TELEGRAM_LENGTH_TOO_LARGE_ERROR	The received telegram data is too long
620	BOF_SPS_VARIABLE_NAME_LENGTH_ERROR	The names of the PLC variables are too long
621	BOF_LOGDBCOM_TIMEOUT_ERROR	LOGDBCOM could NOT be started in the preselected delay time
622	BOF_SSCANF_CONVERT_ERROR	Conversion through sscanf() is NOT possible
623	BOF_PROVI_ADM_FILE_LOAD_ERROR	The PROVI administration file can NOT be read
624	BOF_PROVI_ADM_FILE_ALREADY_EXISTS	The PROVI administration file already exists
625	BOF_PROFILE_SECTION_CREATE_ERROR	The SECTION to be written could NOT be generated through the profile class
626	BOF_PROVI_TEXT_FILE_NOT_FOUND_ERROR	The PROVI TEXT FILE does NOT exist
627	BOF_PROVI_INDEX_FILE_NOT_FOUND_ERROR	The PROVI INDEX FILE does NOT exist
628	BOF_PROVI_TEXT_FILE_OPEN_ERROR	The PROVI TEXT FILE can NOT be opened
629	BOF_PROVI_INDEX_FILE_OPEN_ERROR	The PROVI INDEX FILE can NOT be opened
630	BOF_PROVI_MESSAGE_FILE_CREATE_BREAK_ERROR	The generation of the PROVI MESSAGE FILES was interrupted by BREAK INFO
631	BOF_PROVI_MESSAGE_ACCESS_ERROR	General access error with PROVI MESSAGES via the DIAG SERVER
632	BOF_PROVI_MESSAGE_TYPE_ERROR	There is an Invalid PROVI MESSAGE TYPE
633	BOF_PROVI_ADM_FILE_DATA_ERROR	Necessary data is NOT available in the PROVI administration file

Code	Error Text	Name and Meaning of Error
634	BOF_PROVI_ADM_FILE_SIZE_ERROR	There are too many data entries in the PROVI administration file
635	BOF_PROVI_TEXT_FILE_NOT_EXIST_ERROR	The selected PROVI message text file is NOT available
636	BOF_NO_MEMORY_EXIST_IN_SLOT_NUMBER_ERROR	There is NO free memory in the requested SLOT NUMBER
637	BOF_PROVI_MESSAGE_FILE_DOWNLOAD_BREAK_ERROR	The download of the PROVI MESSAGE FILES was interrupted by BREAK INFO
638	BOF_M_KEY_ADM_TIMEOUT_ERROR	The interlocking mutex for the M key administration could NOT be assigned in the preselected time
639	BOF_WRITE_DATA_TO_LONG_ERROR	On calling the BW command. the write value is too long
640	BOF_SERCOS_CHANNEL_ERROR	Access via the SERCOS channel NOT possible
641	BOF_DATE_TIME_READ_ERROR	The information on date and time could NOT be read from the controller
642	BOF_DATE_TIME_STRING_ERROR	There is a FORMAT error in the transferred date/time string
643	BOF_SYNTAX_IN_SLAVE_MODE_ERROR	The addresses SYNAX controller is in SLAVE mode
644	BOF_PC_TIME_DATE_SET_STATUS_ERROR	Invalid PCTimeDateSetStatusFromDeviceAddr entry in IND_DEV.INI.
645	BOF_PC_TIME_DATE_SET_RATE_ERROR	NO or invalid PCTimeDateSetStatusRateFromDeviceAddr entry in IND_DEV.INI
646	BOF_PC_TIME_DATE_SET_DEVICE_ADDR_ERROR	NO or invalid PCTimeDateSetFromDeviceAddr entry in IND_DEV.INI
647	BOF_PCI_DRIVER_DLL_NOT_FOUND_ERROR	The PCI DRIVER DLL (PCIDP_IF32.DLL) could NOT be found
648	BOF_PCI_DRIVER_DLL_COULD_NOT_BE_LOAD_ERROR	The PCI DRIVER DLL (PCIDP_IF32.DLL) could NOT be loaded
649	BOF_PCI_DRIVER_DLL_FUNCTION_LOAD_ERROR	The PCI DRIVER DLL (PCIDP_IF32.DLL) does NOT CONTAIN ALL NECESSARY functions
650	BOF_PCI_OBJECT_NOT_EXIST_ERROR	The addresses PCI communication object does NOT exist
651	BOF_TRIGGER_EVENT_COUNTER_TO_MUCH_ERROR	A maximum of 1280 event handles are admissible in the trigger list
652	BOF_INVALID_VISUAL_MOTION_ASCII_PROTOCOL_ERROR	There is an invalid VISUAL-MOTION ASCII request
653	BOF_INVALID_VISUAL_MOTION_ERROR_TEXT	There is an invalid VISUAL-MOTION ERROR TEXT answer
654	BOF_VISUAL_MOTION_ERROR	There is a VISUAL MOTION ERROR
655	BOF_NO_DMA_CHANNEL_AVAILABLE_ERROR	No DMA channel in IND_DEV.INI available.
656	BOF_NO_IBS_CMD_INFO_FILE_EXIST_ERROR	NO IBSCMDINFO DATA FILE available.
657	BOF_IBS_CMD_INFO_FILE_READ_ERROR	The IBSCMDINFO DATA FILE can NOT be read.
658	BOF_DEVICE_NO_PLC_COMPONENT_ERROR	The device does NOT have a PLC component.
659	BOF_IBS_CMD_TOOL_ALREADY_RUNNING_ERROR	CMD tool is already running.
660	BOF_NO_CMD_MAIL_BOX_ID_AVAILABLE_ERROR	NO MAIL-BOX-IDs for the CMD tool available.
661	BOF_CMD_SV_REGISTER_IN_USED_ERROR	SV register (CMD tool) is still occupied.

Code	Error Text	Name and Meaning of Error
662	BOF_TO_MUCH_FI_COMMAND_STR_FOR_SELECTIVE_LOAD_ERROR	The max. number for the selected loading of FI commands is exceeded.
663	BOF_CMD_RECEIVED_STATUS_ERROR	A faulty status value occurred in CMD binary telegram.
664	BOF_DEFAULT_CONTEXT_SET_STATUS_ERROR	Invalid DefaultContextSet entry in IND_DEV.INI.
665	BOF_HILSCHER_DRIVER_DLL_NOT_FOUND_ERROR	The HILSCHER DRIVER DLL (INDIF370.DLL) could NOT be found.
666	BOF_HILSCHER_DRIVER_DLL_COULD_NOT_BE_LOAD_ERROR	The HILSCHER DRIVER DLL (INDIF370.DLL) could NOT be loaded.
667	BOF_HILSCHER_DRIVER_DLL_FUNCTION_LOAD_ERROR	The HILSCHER DRIVER DLL (INDIF370.DLL) does NOT contain ALL NECESSARY functions.
668	BOF_INVALID_NC_MEMORY_ERROR	An INVALID NC memory was addressed.
669	BOF_NC_PACKAGE_ID_STR_LENGTH_ERROR	The transferred NC package identification string is too long.
670	BOF_NO_SYCON_INFO_FILE_EXIST_ERROR	NO SYCONINFO DATA FILE available.
671	BOF_SYCON_INFO_FILE_READ_ERROR	The SYCONINFO DATA FILE can NOT be read.
672	BOF_SYCON_TOOL_ALREADY_RUNNING_ERROR	SYCON tool is already running.
673	BOF_NO_IO_CONFIGURATION_LIST_IN_SPS_ERROR	There is NO IO configuration list in the addressed PLC.
674	BOF_INVALID_FIELD_BUS_SYSTEM_ERROR	An unknown field bus system is addressed.
675	BOF_IFSIM_FILE_READ_ERROR	The SIFSIM.TXT DATA FILE can NOT be read.
676	BOF_NO_IFSIM_FILE_EXIST_ERROR	The IFSIM.TXT DATA FILE is NOT available.
677	BOF_NO_FOUND_KEY_IN_IFSIM_FILE_ERROR	The addressed key is NOT available in IFSIM.TXT-DATA FILE.
678	BOF_INVALID_DATA_IN_IFSIM_FILE_ERROR	Invalid data is entered in IFSIM.TXT-DATA FILE.
679	BOF_DATA_STRUCT_LENGTH_ERROR	The length of the data structure does NOT correspond.
680	BOF_NO_SYCON_DATA_AVAILABLE_ERROR	SYCON DATA from control is NOT available.
681	BOF_NODE_IS_NOT_READY_ERROR	Status error 0x87 occurred during data communication of bus configurators.
682	BOF_INVALID_COMMUNICATION_TYPE_ERROR	Unknown communication type in COMINTFC.EXE.
683	BOF_MPI_MANAGER_MODE_ERROR	Invalid MPI MANAGER MODE entry in IND_DEV.INI.
684	BOF_MPI_MANAGER_SHUTDOWN_ERROR	Invalid MPI MANAGER SHUT-DOWN entry in IND_DEV.INI.
685	BOF_NET_MANAGER_SHUTDOWN_ERROR	Invalid NET MANAGER SHUT-DOWN entry in IND_DEV.INI.
686	BOF_FI_LOGIN_DISABLE_ERROR	FI login procedure is currently disabled.
687	BOF_DRIVE_PARAMETER_UPLOAD_FILE_ALREADY_EXIST	DRIVE PARAMETER UPLOAD FILE already exists.
688	BOF_DRIVE_ADDRESS_NOT_AVAILABLE_ERROR	NO drives are defined or the selected drive is not defined.
689	BOF_FUNCTION_NOT_IMPLEMENTED_ERROR	The addressed functionality is NOT YET implemented.
690	BOF_UNKNOWN_ERROR_CLASS	An unknown error class is available.
691	BOF_UNKNOWN_DRIVE_PARAMETER_TYPE_ERROR	NO standard parameter (S parameter) or product parameter (P parameter).
692	BOF_DRIVE_PARAMETER_UPLOAD_BREAK_ERROR	A DRIVE PARAMETER UPLOAD PROCEDURE has been interrupted by BREAK-INFO.

Code	Error Text	Name and Meaning of Error
693	BOF_SERCOS_DATA_LINES_TO_MUCH_ERROR	The max. number of data lines for SERCOS operating data is exceeded.
694	BOF_VIRTUAL_DRIVE_ERROR	The selected drive is a virtual axis.
695	BOF_MTA_SERCOS_CONNECTION_ERROR	Access via the MTA-SERCOS data channel is NOT possible.
696	BOF_DRIVE_PARAMETER_DOWNLOAD_FILE_NOT_EXIST	DRIVE PARAMETER DOWNLOAD FILE is NOT available.
697	BOF_DRIVE_PARAMETER_DOWNLOAD_FILE_READ_ERROR	DRIVE PARAMETER DOWNLOAD FILE could NOT be read.
698	BOF_DRIVE_PARAMETER_ERROR_FILE_ALREADY_EXIST	The DRIVE PARAMETER ERROR FILE already exists.
699	BOF_DRIVE_PARAMETER_DOWNLOAD_FILE_DATA_ERROR	Necessary entries are NOT available in DRIVE PARAMETER DOWNLOAD FILE.
700	BOF_TIMEOUT_FOR_SERCOS_PHASE_SWITCH_ERROR	NO "stable" phase of the drive was reached within the defined delay time.
701	BOF_DRIVE_NOT_IN_PARAMETER_MODE_ERROR	The drive could NOT be switched into parameterization mode.
702	BOF_SERCOS_ATTRIBUTE_CHECK_ERROR	The SERCOS attribute of the operation date to load does NOT correspond with the attribute of the drive.
703	BOF_SERCOS_OPERATING_DATA_NOT_WRITE_ERROR	The SERCOS operation date should NOT be written.
704	BOF_SERCOS_PARAMETER_NOT_AVAILABLE_ERROR	The SERCOS operation date is NOT contained in drive parameter download file.
705	BOF_DRIVE_ADDRESS_NOT_AVAILABLE_IN_DOWNLOAD_FILE	The selected drive address is NOT available in DRIVE PARAMETER DOWNLOAD FILE.
706	BOF_GENERAL_DELETE_FILE_ERROR	The file can NOT be deleted (write-protected).
707	BOF_DRIVE_PARAMETER_ERROR_FILE_NOT_EXIST	DRIVE PARAMETER ERROR FILE is NOT available.
708	BOF_DRIVE_PARAMETER_ERROR_FILE_READ_ERROR	The DRIVE PARAMETER ERROR FILE can NOT be read – format error.
709	BOF_DRIVE_ADDRESS_NOT_AVAILABLE_IN_ERROR_FILE	The selected drive address is NOT available in DRIVE PARAMETER ERROR FILE.
710	BOF_SERCOS_PARAMETER_DATA_FILE_NOT_EXIST	The SERCOS DATA FILE is NOT available.
711	BOF_SERCOS_DATA_NOT_AVAILABLE	The selected SERCOS DATA are NOT available.
712	BOF_INVALID_SERCOS_OPERATING_DATA_ERROR	NO valid SERCOS operation date available.
713	BOF_TOOL_LIST_UPLOAD_FILE_ALREADY_EXIST	The TOOL LIST UPLOAD FILE already exists.
714	BOF_PROCESS_TOOL_MANAGEMENT_NOT_EXIST_ERROR	Selected PROCESS does NOT have a tool management.
715	BOF_PROCESS_NO_TOOLS_AVAILABLE	Selected PROCESS does NOT have tools.
716	BOF_NO_TOOLS_AVAILABLE	NO tools available (in all processes).
717	BOF_TOOL_LIST_UPLOAD_BREAK_ERROR	THE TOOL LIST UPLOAD PROCEDURE has been interrupted by BREAK-INFO.
718	BOF_TOOL_LIST_DOWNLOAD_FILE_NOT_EXIST	The TOOL LIST DOWNLOAD FILE is NOT available.
719	BOF_TOOL_LIST_DOWNLOAD_FILE_DATA_ERROR	Necessary entries are NOT available in TOOL LIST DOWNLOAD FILE.
720	BOF_TOOL_LIST_ERROR_FILE_ALREADY_EXIST	The TOOL LIST ERROR FILE already exists.
721	BOF_TOOL_LIST_MAX_EDGE_NUMBER_ERROR	The tool edge number in download file is larger than the currently defined tool edge number.

Code	Error Text	Name and Meaning of Error
722	BOF_TOOL_LIST_MAX_BASIC_USER_DATA_ERROR	The max. defined user data in download file is larger than the max. currently defined user data.
723	BOF_TOOL_LIST_MAX_EDGE_USER_DATA_ERROR	The max. defined tool edge user data in download file is larger than the max. currently defined tool edge user data.
724	BOF_TOOL_LIST_PROCESS_DATA_ERROR	NO TOOL LIST DATA available for the process selected in download file.
725	BOF_TOOL_LIST_SPINDLE_ERROR	NOT all tools could be transferred in the spindles.
726	BOF_TOOL_LIST_GRIPPER_ERROR	NOT all tools could be transferred in the grippers.
727	BOF_TOOL_LIST_POCKET_ERROR	NOT all tools could be transferred in the magazine locations.
728	BOF_TOOL_LIST_DATA_WRITE_ERROR	A GENERAL write error occurred during transmission of tool list data.
729	BOF_TOOL_LIST_DOWNLOAD_BREAK_ERROR	The TOOL LIST DOWNLOAD PROCEDURE has been interrupted by BREAK-INFO.
730	BOF_TOOL_LIST_ERROR_FILE_NOT_EXIST	The TOOL LIST ERROR FILE is NOT available.
731	BOF_TOOL_LIST_ERROR_FILE_READ_ERROR	The TOOL LIST ERROR FILE can NOT be read – format error.
732	BOF_SERCOS_DATA_ACCESS_ERROR	NOT all necessary SERCOS parameters could be processed.
733	BOF_SERCOS_BUSY_FLAG_ERROR	SERCOS data could NOT be accessed within the defined time (BUSY-FLAG set).
734	BOF_NO_ONLINE_MODE_POSSIBLE	The device could NOT be switched ONLINE, as NO communication path is defined (CommStr=OFF).
735	BOF_KILL_MANAGER_INVALID_DATA_ERROR	Invalid data in [KillManager] entry in IND_DEV.INI.
736	BOF_GENERAL_WINDOWS_ERROR	A general WINDOWS error occurred.
737	BOF_TIMEOUT_FOR_CREATE_FI_DATA_MAP_ERROR	The SYSTEM MAP or COMMON-MAP could NOT be created in the defined delay time.

7.3 Error Codes 1000 to 1999

Code	Error Text	Name and Meaning of Error
1001	BOF_FAULT_FCT	Invalid function code passed (e.g. "CW" for a read function).
1002	BOF_DATA_FAULT	Data is invalid.
1003	BOF_FAULT_PIPE_NR	Incorrect pipe number
1004	BOF_NO_CREATED_PIPE	Pipe not created.
1005	BOF_PIPE_NOT_RUN	Pipe not running.
1006	BOF_NO_DATA_CREATED	Data not created.
1007	BOF_PIPE_NOT_BREAK	Pipe not running.
1008	BOF_NO_VALUE	No value string.
1009	BOF_BUFFER_SIZE_TO_SMALL	Buffer is too small.
1010	BOF_NO_INDEX_DATA	No index data.
1011	BOF_FAULT_INDEX_NR	No index number.
1012	BOF_DATA_NO_FOUND	Data not found.
1013	BOF_FUNC_LOCK	Function blocked; repeat access.
1014	BOF_NEGATIVE_ACKNOWLEDGE	Negative acknowledge for the FI command executed.

Code	Error Text	Name and Meaning of Error
1015	BOF_PARAMETER_INVALID	Invalid parameter details.
1016	BOF_FUNCTION_INVALID	Invalid FI command.
1017	BOF_DEVICE_TIMEOUT	Timeout of NC-Task
1018	BOF_INDEX_DATA_ERROR	Index data from the resultbuf is corrupt.
1019	BOF_UNKNOWN_TOOL_STORE	Unknown type of memory (tool store)!= magazine, spindle, gripper.
1020	BOF_MAX_COUNT_ERROR_FOR_TOOL_DATA	Maximum count error for tool data.
1021	BOF_NO_TOOLMANAGMENT	No tool management.
1022	BOF_NO_TOOLMANAGMENT_FOR_PROCESS	Tool management not available for process.
1025	BOF_RESULT_BUF_TYPE_ERROR	Error result type is incorrect or not supported.
1030	BOF_NC_PACKET_IS_PRESENT	NC package already present in control.
1031	BOF_NC_PARTPROGRAM_IS_NOT_PRESENT	NC program is not present.
1032	BOF_NC_PROGRAM_DIRECTORY_IS_EMPTY	Part-directory or program directory is empty.
1033	BOF_NC_PROGRAM_COMPILER_ERROR	Error flag set by program.
1034	BOF_NC_DAT_FILE_NO_PRESENT	NC-DAT file does not exist or cannot be opened.
1035	BOF_NC_PACKET_DIR_NOT_PRESENT	Package directory does not exist.
1036	BOF_NC_PACKET_DIR_READ_ERROR	Package directory can not be read in.
1037	BOF_NC_PARTPROGRAM_DIR_NOT_PRESENT	Program directory does not exist.
1038	BOF_NC_PARTPROGRAM_DIR_READ_ERROR	Program directory can not be read in.
1039	BOF_PIPE_CYCLE_LIST_EMPTY	Pipe request list is empty.
1040	BOF_PIPE_RUN	Pipe already running.
1041	BOF_ITEM_DATA_INVALID	Partial result is invalid.
1042	BOF_FUNC_INVALID_PARAM	Invalid parameter for function
1043	BOF_PIPE_NO_FREE_PIPE	All pipes already assigned.
1044		Communication channel is already running.

Code	Name and Meaning of Error
1301	Exception
1302	No "Common" information section present
1303	No "Common" information key present
1304	FI job interrupted
1305	Download control file cannot be read
1310	No "Package" information section present
1311	No "Package" information key present
1312	"Package" key value not permitted
1315	No ListOfPrograms section present
1321	No NC program section present
1322	Program key value not permitted
1323	No NC program info present
1324	Process number does not conform
1325	Data file cannot be read
1331	Parameter record not active

Code	Name and Meaning of Error
1332	No memory for creating object
1333	Invalid value passing
1334	Invalid access mode
1340	No valid function call
1341	No COMPILER info section present
1342	No COMPILER info key present
1350	No variable/events/D-correction section present
1351	Variable/event/D-correction data file cannot be read
1355	No events section present
1360	No D-correction section present
1361	No D-correction parameter record present
1370	Invalid transfer of parameters
1371	Invalid key
1372	Invalid PLC variables type
1373	Telegram delimiter reached
1374	Configuration file does not exist
1375	Section "project" does not exist
1376	Key "project" does not exist
1377	Section "variables" does not exist
1378	Key "variables" does not exist
1379	Section "buffer" does not exist
1380	Key "buffer" does not exist
1381	Invalid parameter transfer at "buffer" section
1382	Invalid parameter transfer at "buffer" section
1383	Section "trigger\condition" does not exist
1384	Key "trigger\condition" does not exist
1390	No I/O tables section
1391	I/O tables value do not exist
1392	Short identification is invalid

Code	Error Text	Name and Meaning of Error
1501	BOF_FUNC_NAME_LIMIT150	Name of interface 'B' functions is too large.
1502	EXCEPTION	Internal error.
1503	EXCEPTION	Internal error.
1504	EXCEPTION	Internal error.
1505	EXCEPTION	Internal error.
1506	EXCEPTION	Internal error.
1507	EXCEPTION	Internal error.
1508	EXCEPTION	Internal error.
1509	EXCEPTION	Internal error.
1510	EXCEPTION	Internal error.

Code	Error Text	Name and Meaning of Error
1511	EXCEPTION	Internal error.
1512	BOF_FUNC_EOF_STRING_150	FI command incomplete.
1513		Maximum number of lines has been reached
1514		Maximum number of columns has been reached

7.4 Error Codes 2000 to 2999

Code	Meaning
2001	No channel free.
2002	Channel already open.
2003	Channel cannot be closed.
2004	Channel not open.
2005	Re-initialization error.
2006	Channel cannot be opened.
2007	Version is incompatible to file "LOGINTFC.EXE".
2008	Channel flags are blocked.
2009	Access to controls temporarily blocked due to download.
2010	Receive request timeout.
2011	No request active.
2012	Invalid event in receive.
2013	Status request still active.
2014	Cyclic request still active.
2015	No cyclic request active.
2016	Single request still active.
2017	Pass format of routine "GetSysMsg" is faulty.
2018	System message (SysMsg) cannot be issued.
2019	DMA request is still active.
2020	Invalid FI command code.
2021	Invalid result type.
2022	Result too long for receive buffer.
2023	Invalid FI command during group request.
2024	Empty result buffer
2025	Request too long for request buffer.
2026	Faulty input format.
2050	"LOGINTFC.DAT" file cannot be opened.
2051	No channel free.
2052	Communication process (COM task) not responding.
2053	"LOGINTFC.EXE" file not found.
2059	Error message from the LOG process.
2060	False syntax in the command string
2081	The addressed controller is off line (DeviceStatus=OFF)
2082	The addresses controller is in monitor mode

Code	Meaning
2126	Faulty result type
2127	Faulty request on response
2150	"LOGINTFC.DAT" file cannot be opened.
2154	File Version Mismatch.
2155	"LOGINTFC.DAT" file is too large.
2156	Internal configuration error.
2157	Faulty ChFreeList from GetDeviceCommAddrExtend()
2160	Invalid command string.
2161	Telegram code not implemented.
2162	Parameter outside the limit value.
2163	Invalid parameter syntax.
2164	Unknown PLC variable.
2165	Not enough parameters transmitted.
2166	PLC map file cannot be opened.
2167	PLC variable type not implemented.
2168	PLC variable reference error.
2169	Date cannot be edited.
2170	Checksum error.
2171	Undefined telegram code.
2172	Missing processing rule.
2173	Too much data for the response telegram.
2174	Unknown additional diagnostics information.
2175	Unknown unit.
2176	PLC variable is larger than 240 byte.
2177	Device has no PLC
2178	NDC parameter too large
2179	Result buffer too small
2180	No axis defined
2181	The addressed controller is off line (DeviceStatus=OFF)
2182	The addresses controller is in monitor mode
2201	Input string "Date-Time" not in format: "DD.MM.YY hh:mm:ss".
2202	Effective data length of SIS telegram is too large.
2304	Specified file not found.

7.5 Error Codes 4000 to 4999

Code	Error Text	Name and Meaning of Error
4000		An error has been detected in checking the composition of the request of the "BR_NPA1....." and ff command. (see also FI command: "NPA1_/?") The following error messages in the error window provide additional information regarding the error.
	ERROR : invalid ParNo/value	An incorrect parameter number has been transmitted.
	ERROR : invalid ParNo/value	An error has been detected in checking transmission of the parameter. The possible cause of this is an invalid parameter name or an error in the order in which the entry was made. The first parameter number must be smaller than the second parameter number. (see also FI command: "NPA1_/?")
	invalid Parametervalue or No.: [<ParNr>]	An error has been detected in checking the command. Either a directory number has been selected that is outside of the range of validity or a parameter name is invalid.
	[No.] missing Startparameter	The command has not been passed on in its entirety.
	Illegal start parameter value [wrong value passed]	An incorrect value has been detected for the parameter number.
	ERROR : different Parametertypes	Requesting different types of parameter within one request command is not possible.
	ERROR : Second ParNo before First ParNo	The parameter request must be made starting from the lower number and moving to the higher number. (see also FI command: "NPA1_/?")
	ERROR : Invalid startparameter - ProcNo out of Range	When requesting one or more process parameters, an invalid definition range has been detected. A request is only possible within the NC process numbers [0...6].
	ERROR : Invalid startparameter - AxesNo out of Range	When requesting one or more axis parameters, an invalid definition range has been detected. A request is only possible within 1 to 20 or 32.
4001	ERROR : invalid function	The FI command contains an invalid parameter.
4002	NO_PARAMETER_DATA_FOUND	The requested parameter(s) do not exist. Either parameters have been requested that have not already been defined or the appropriate parameter has been removed. Check all entries and make sure that the corresponding data exists in BOF menu item <F5> (Parameters).
4003	Verz_No_Out_of_Range	An invalid range has been detected when checking the command passed. Check the directory number entries.
4004	BR_NPA_No_Data_File_exists	The attempt to read data from a file could not be executed. Re-check your entries for possible processes or axes on the definition range. Otherwise, try to view the data using BOF menu item <F5> (Parameters). The data may not exist or the installation has not been made correctly. In this case, please contact our customer service department.

Code	Error Text	Name and Meaning of Error
4005	BR_NPA_No_INI_File_exists	<p>Parameter data could not be read from an initialization file. Possible causes are:</p> <p>The file does not exist. There has been an installation error or the file has been deleted accidentally. ⇒ Execute Update/Installation</p> <ul style="list-style-type: none"> • There is an error in the file. The file has been accidentally edited or illegally copied. Data recognition has thereby been rendered invalid. <p>⇒ Carry out an update installation or contact our customer service department.</p> <ul style="list-style-type: none"> • The file has been damaged, either by a system crash or by a defect on the storage media. <p>⇒ Contact our customer service department.</p>
4006	Device Address out of Range	A system outside the definition range has been selected in the command.
4007	Buffer error detected =[Error Code]	<p>Internal error. The data range set for provision of the results is not large enough. This problem can be remedied as follows:</p> <ul style="list-style-type: none"> • Request fewer data. Use a group request. • Increase the memory made available for the data range when creating the application yourself. <p>⇒ Contact our customer service department.</p>
4012	Create_DLL_Error detected!	The result buffer could not be initialized. Contact our customer service department.
4013	Function will not run for DLL-Version-Mode:[DLL-Version]	An attempt has been made to execute a command that is not available in the existing DLL version.
4014	Corrupted Parameter Identification = [Parameterident.]	Initialization of the required data memory is not possible due to an error in parameter recognition. Check to make sure that there is a valid parameter record for all devices. If necessary, re-transmit the parameter(s) to the controls. If the error remains, or the parameter(s) cannot be transmitted, then please contact our customer service department.
4015	Wrong version installed	<p>This error message always appears on starting the GUI when the memory could not be initialized based on the version being used.</p> <p>Up to and including version 18, error code 4109 is returned. From version 19, the corrected error code 4015 is returned.</p>
4017	**OK** (none Parameterset in CNC) – finished function FillParamDataInCncDataMap	This text message only appears in the starting-up phase with the setting "/U0" of the start parameter (for TSRPG25I.EXE) if an empty parameter name has been transmitted. This means that no parameter record is as yet in the control. No error is returned.
4100	Couldn't open ParameterIndexFile: [File Error=xxxx]	<p>An error has been detected when attempting to open the parameter directory file. Any of the following could trigger this error:</p> <ul style="list-style-type: none"> • Versions do not conform <p>⇒ Parameters need to be converted. (refer to "Converting Parameters")</p> <ul style="list-style-type: none"> • The parameter directory file has been accidentally destroyed. • The disk drive is faulty.
4102	ParameterIndexFile has wrong structure	An error has been detected when reading the parameter directory file indicating that the data in the file is not in the correct format. Check this by running "Converting Parameters". If the error continues to occur after this then you must contact our customer service department.

Code	Error Text	Name and Meaning of Error
4103	to much Indizies found – File has wrong structure ?:	An error has been detected when creating the directory data. More directories have been recognized than allowed by the definition range. Probably the parameter directory data is from an earlier version. Execute "Converting Parameters". If the error persists, please contact our customer service department.
4104	Invalid parameter value detected	An invalid range was detected when initializing (booting up the GUI). Contact our customer service department.
4105	Can't create Parameterindexbuffer: [filename]	No data could be provided in the memory. Close other applications to free up enough memory for the compilation of the data.
4108	Don't found the Parameter: [Parameternummer]	This message text only appears in the starting phase with the setting "/U0" (in case of TSRPG25I.EXE). This error code is only returned when an attempt has been made to request a non-defined parameter.
4109	Didn't get BOF-Version – BOF installed? [error code]	The attempt to determine the GUI version has failed. Contact our customer service department.
4110	Couldn't load Parameters in shared Memory – Error= [ErrorCode]	Initialization failed when starting the GUI. Contact our customer service department.
4111	Invalid parameter value Cxx.053 [Cxx.053 <Value>]	Initialization failed when starting the GUI. An invalid axis meaning has been detected in the current parameter record of a device. Switch the corresponding system to offline and correct the appropriate parameter record. After you have done this, the system should be brought back online and the altered parameter record should then once more be transmitted to the controls. If the problem persists, please contact our customer service department.
4200	Invalid start parameter	This message text only appears in the starting phase with the setting "/U0" (in case of TSRPG25I.EXE). This error code is always returned when a parameter request has been made outside the definition range. Otherwise, please contact our customer service department.
4201	Invalid parameter type	A parameter request has been made with a non-defined parameter type. Check the entry and/or request
4202	Buffer size not enough	The result of the parameter request cannot be transmitted as the transmission range is not large enough. For applications that you have created yourself, increase the size of the transmission range. Otherwise, please contact our customer service department.
4203	Error detect by ReadPar_Value – can't read Data [Error number or directory number]	The requested parameter could not be formed or found. Re-check your request or contact our customer service department.
4204	Could not find direct. entry	No error message is emitted. The error code is always returned when, after a request for a particular parameter directory entry, the parameter number has not been found.
4205	Function will not running by InterFace-Ver- sion: [Version]	During the command request, the program has detected that it cannot be run on this version. Contact our customer service department.
4220	Invalid Save Order by Save function please test the ParType by Save_Begin;	The "writing parameters" function has been repeatedly started before the previously started command has been completed.
4221	invalid IndexNo by Save[ParameterNumber]	The parameter number is outside the definition range.
4222	co_str_ConWData_Buffer_Size_to_small [defined size 2000]	This message text only appears in the starting phase with the setting "/U0" (for TSRPG25I.EXE). The error code is always returned if the defined memory range in the program is too small. In this case, please contact our customer service department.

Code	Error Text	Name and Meaning of Error
4223	WriteError by Config-SCR-File = [error number]	An error has been detected when writing the configuration parameters. The function has been cancelled. Contact our customer service department.
4224	SaveError – Couldn't rename DAT->old [file name]	This message text only appears in the starting phase with the setting "/U0" (for TSRPG25I.EXE). The attempt to rename the original file could not be executed. Check the properties for the relevant parameter file and also the remaining free space on the storage medium.
4225	SaveError – Couldn't rename tmp → dat ⇒ copy old to DAT[Dateiname]	This message text only appears in the starting phase with the setting "/U0" (for TSRPG25I.EXE). The attempt to recopy the newly created file could not be executed. Check the properties for the relevant parameter file and also the remaining free space on the storage medium.
4226	Missing file = [file name]	The previously created file could not be found or opened. Check the free memory on the storage medium.
4227	Create instance failed - can't save Data	An internal error has occurred. Contact our customer service department.
4228	Can't create file = [file name <additional info>]	The specified file could not be created acc. to the additional info. Check the amount of free memory on the storage medium and the access properties of the directory concerned. Otherwise, please contact our customer service department.
4229	Can't create file = [file name]	Specified file could not be found. Check your entry. Perhaps an incorrect directory number has been entered.
4230	ConWData_Error_by_WPar_Begin	This error code is reported after an error has occurred in transmitting an error code to the function interface.
4231	ConWData_Error_by_WPar_End	This error code is reported after an error has occurred in transmitting an error code to the function interface.
4232	Error detect by WritePar_Value - can't save Data:[file name or parameter value]	This error message is only displayed internally when in debug mode. The value of a parameter could not be transferred to the file specified. Contact our customer service department.
4233	Attention - Return value of process definition undefined	This error message is only displayed internally when in debug mode. An error has been detected in the generation of the process definition. Check the process definitions within the processing of the parameter. Otherwise, please contact our customer service department.
4234	Can't actualize direct.line [Parameter directory line]	The specified parameter directory line could not be updated.
4235	Can't actualize date or length in directory line	The date or length could not be updated when the parameter directory line was being updated. Contact our customer service department.
4236	CreateFiErrorResult_DLL failed	This error message is only displayed internally when in debug mode. The error message could not be transmitted to the FI. Contact our customer service department.
4237	Can't write by undefined parameter number [Parameter number]	An attempt has been made to write a non-defined parameter for this type of parameter. Check your entry. Check, e.g. that parameters exist for the various axis types.
4238	Cxx.083 : more as defined Elements for Cxx.083 found:	An attempt has been made to transmit a larger number of compensation values than is listed as the max. range of a compensation list. Re-check your entry. A maximum of 1000 values can be included in a list.
4239	Installation error - missing file: [File name]	Specified file not found. Re-run the update installation. If the error persists after this then you must contact our customer service department.

Code	Error Text	Name and Meaning of Error
4240	Invalid Parameter value =[parameter line]	An invalid range has been detected in the specified parameter line. Re-check your entry.

Converting Parameters

An update installation of the Bosch Rexroth GUI automatically results in a parameter conversion from version "xx" to the next version "yy". Parameters can be converted by calling the "COPAxxy.EXE" conversion program in standard installation directory "C:\Programme\Indramat\MTGUI\Bin". Both wildcards "xx" and "yy" represent the directory ID from which version and into which version the conversion is carried out.

Note: In case of an error, you can start the conversion program "COPAxxy.EXE" with the starting parameter "/" to receive additional messages.

7.6 Error Codes 5000 to 5999

Code	Name and Meaning of Error
5001	Alias used is not defined (is not yet used)
5002	Invalid device address
5003	Syntax error in the device address
5007	Invalid device address
5010	No request active
5011	Error in request string
5012	No response buffer specified
5013	Wrong mode during cyclic login (internal error)
5014	Invalid receive telegram
5015	Invalid receive telegram
5050	No access to remote PC possible
5051	Remote connection has been canceled
5052	Network interface cannot be initialized
5101	Unexpected general error (internal error)
5102	Memory error
5151	No memory for data to be sent
5152	No memory for data to be received
5153	No memory for telegram to be received
5401	Unspecific error
5402	Invalid parameter passed to function
5403	Transfer timeout, remote PC not ready, or network connection down
5404	Send failed; error sending to a remote PC
5405	Memory shortage; in remote access of the interface
5406	Invalid connection to a remote PC
5407	Service disabled
5408	Connection to remote partner aborted

Code	Name and Meaning of Error
5409	Invalid parameter hook ID; Sys Message Handling
5410	Invalid device number

7.7 Error Codes 6000 to 6999

Code	Error Text	Name and Meaning of Error
6001	BOF_C_TYP_FAULT	Transmitted data type not OK.
6002	BOF_C_LEN_FAULT	Transmitted data length not OK.
6003	BOF_C_DEV_FAULT	Transmitted system number not OK.
6004	BOF_C_PAKNR_FAULT	Transmitted package number not OK.
6005	BOF_C_PROZ_FAULT	Transmitted process number not OK.
6006	BOF_C_PROG_FAULT	Transmitted program number not OK.
6007	BOF_C_FILE_NOT_DEL	File cannot be deleted
6008	BOF_C_NO_NCPROG_CREATED	No NC program in part-program directory
6009	(BOF_C_NCPROG_CREATED)	NC program exists (where check =1)
6010	BOF_C_DESCR_FAULT	Identifier, e.g. data length not OK
6011	BOF_C_FILE_WRITE_CLOSE_ERROR	Error writing or closing a file.
6012	BOF_C_PACK_EXIST	NC package already available
6013	BOF_C_INVALID_MTCNC_NUMBER	Invalid system number
6014	BOF_C_FILE_NOT_FOUND	File not found
6015	BOF_C_PAR4_FAULT	Parameter 4 not OK
6016	BOF_C_NO_NC_SEEK_SET	NC program cannot be positioned to N0000
6017	BOF_C_NCPROG_NOT_READ	File cannot be opened
6018	BOF_C_PART_PROGR_DIRECTORY_ERROR	Part-program directory could not be read.
6019	BOF_C_PACKET_DIRECTORY_ERROR	Package program directory could not be read.
6020	BOF_C_PAR5_FAULT	Parameter 5 not OK
6021	BOF_C_PAR6_FAULT	Parameter 6 not OK
6022	BOF_C_COMP_ERROR	Test error after commands to be compiled.
6023	BOF_C_CURS_FILE_ERROR	Handling error in NCCPxx.DAT file.
6024	BOF_C_TOOL_SETUP_LIST_NOT_READ	Error in setup list
6025	BOF_C_TOO_MUCH_TOOLS_IN_LIST	More tools in the setup list than in the parameters.

7.8 Error Codes 7000 to 7999

All error codes – except for error code 7000, which shows a syntax error in the compiled NC program – normally require you to contact Bosch Rexroth for further clarification of their cause. Either this is a software error or files for the GUI have been deleted or corrupted.

Note: As for all error codes, additional information regarding an error that has occurred can be requested via the "General error result line" (see chapter entitled "Error Codes"). The error information informs the user in plain text about the cause of the error.

Code	Meaning and Notes Regarding Diagnosis and Troubleshooting
7000	Syntax error in NC program. The "General Error Result Line" contains further information.
7002	File with incorrect information. The "General Error Result Line" contains the file name and the line.
7005	File not found. The "General Error Result Line" contains the file name.
7006	File cannot be created. The "General Error Result Line" contains the file name.
7008	File cannot be read. The "General Error Result Line" contains the file name.
7009	Error in connecting the function interface. No connection can be made to the device (control unit) via the function interface.
7015	Too many axes defined. More than 9 Axes used in NC process.
7016	Invalid number of parameters. The number of parameters in the "NCPRG.CFG" file has been exceeded.
7017	Axis name is invalid. The axis name in axis parameter "CXX.001" or "CXX.075" is invalid.
7018	Axis meaning is invalid. The axis meaning in axis parameter "CXX.053" is invalid.
7019	Maximum axis speed is invalid. The value of axis parameter "CXX.016" is invalid.
7020	Maximum axis acceleration is invalid. The value of axis parameter "CXX.018" is invalid.
7021	Lowest run time of an NC record = [2.5...30ms]. The counter value of the parameter "METB" in the NC options of the BOF/GBO is outside the allowed range.
7022	Lowest run time of an NC record is invalid. The counter value of the parameter "METB" in the NC options of the BOF/GBO is invalid.
7023	Only 4 or 5 decimal places are allowed. Process parameter "BXX.002" is invalid.
7024	Invalid counter value. The counter value of parameter "VFBT" or "BBTC" in the NC options of the BOF/GBO is invalid.
7025	Only 0 (mm) or 1 (inch) permitted! The process parameter "BXX.001" is invalid.
7026	Counter value outside the allowed range. Axis parameter "CXX.006" is smaller than 0.1.

Code	Meaning and Notes Regarding Diagnosis and Troubleshooting
7027	Internal block number is invalid. The block numbers of the NC program file are in the wrong order.
7028	Block number in the file is invalid. The "General Error Result Line" (p. 7-1) contains the names of the file in which the block numbers are incorrect.
7070	Counter value outside the permitted range (1..10). The counter value of parameter "BBTC" in the NC options of the BOF/GBO is outside the permitted range.
7077	Counter value outside the permitted range (1.0.25). The counter value of parameter "VFBT" in the NC options of the BOF/GBO is outside the permitted range.
7083	Invalid parameter. The "General Error Result Line" contains an invalid control parameter.

7.9 Error Codes 8000 to 8999

Code	Error Text	Name and Meaning of Error
8000	OUTOFMEMORY	Heap memory is full
8001	PARAMETER_FAILURE	Error in transmitting parameter (response telegram)
8002	INVALIDARG	Incorrect request string
8003	REQUEST_NOT_FILLED	Internal run error
8004	GET_ATTRIBUT_FAILED	Incorrect attribute contained in response telegram
8005	WALK	Internal run error
8006	EXTRACT_COMMON_INFO_FAILED	Error in transmitting parameter (response telegram)
8007	WRONG_DATA_SIZE	Undefined data length in the response
8008	ELEMENT_UNEXPECTED	Unexpected coding in BW_SPA1
8009	SERCOS_LONG_TO_ASCII	Result conversion error.
8010	VERSION_MISMATCH	Command did not yet exist for set IfDIIMode.
8011	ERROR_BYTE_INFO	Error reading error byte information
8012	CANT_OPEN_MODULDEF_INI	The "Moduldef.ini" file cannot be opened.
8013	WRONG_PROFILE_FILENAME	Wrong profile file name
8014	WRONG_SECTION_INFORMATION	Wrong section information in profile
8015	ERROR_IN_LAST_LINE	Error in the last profile line
8016	Reserved	Reserved
8017	Reserved	Reserved
8018	SECTION_NOT_FOUND	Section not found; (e.g., incorrect device or module parameter).
8019	LANGUAGE_NOT_FOUND	Language not supported
8020	Reserved	Reserved
8021	MODUL_NOT_FOUND	Module not found; (e.g., missing keyword module name).
8022	DEVICE_ADDR_GENERAL_NOT_FOUND	No device entry found.
8023	FB_NOT_FOUND	No function component found; (e.g., error or message keyword missing).
8024	DEVICE_ADR_FALSE	Device address not in the valid range.
8025	MODULE_NO_FALSE	Module number not within valid range (0-99)
8026	KEY_WORD_FALSE	Wrong keyword; (e.g., no ModulY in section name [DeviceAddrXModulY])

Code	Error Text	Name and Meaning of Error
8027	MODULE_ASSIGN_PROCESS	No module can be found for the specified process.
8028	PROCESS_NO_FALSE	Process number not within valid range (0-31)
8031	RESULT_TYPE_INVALID	Invalid result type.
8032	E_COM_SIS_TEL_TOO_LONG	Transmitted length of telegram exceeds maximum SIS telegram length.
8033	E_COM_SIS_TEL_POS	Telegram position addressed is outside the SIS telegram range.
8034	E_COM_SIS_TEL_NO_LEN	SIS telegram length is "0"
8035	E_COM_OPERATING_SYSTEM_NOT_SUPPORTED	Operating system is not supported.
8036	SERCOS_ASCII_TO_LONG	Error in converting the value to be written.
8038	PROCESS_NOT_DEFINED	The process addressed does not exist
8039	NO_TOOLMANAGEMENT	The tool management is not activated for the process
8040	WRONG_TOOL_NUMBER	Invalid tool number
8041	WRONG_SPINDLE_NUMBER	Wrong spindle number
8042	WRONG_GRIPPER_NUMBER	Wrong gripper number
8043	UNKNOWN_TOOL_STORE	Unknown tool store (memory)
8044	INVALID_VALUE	Value or element of the value list not correctly formatted
8045	MUTEX_TIMEOUT	The command access control was not quit in time
8046	UNKNOWN_DEVICETYPE	An unknown device type has been detected

7.10 Error Codes 10000 to 10999

Code	Meaning
10001	The WinHMI component is not installed.
10002	Incorrect WinHMI version
10021	DDS not installed
10022	Incorrect DDS version
10023	WinPcl not installed.
10024	Incorrect WinPcl version.
10101	Incorrect version of the function interface.
10102	The "CreateGroup" routine has failed.
10103	Error in command string.
10104	Unknown variable requested.
10105	Error in determining the status.
10107	"HMI_Data.DLL" file not found.
10110	WinHMI has not been started in the same process
10111	Error on determination of the ProVi message.
10112	ProVi message type does not exist.
10201	DDS not ready yet.
10210	Unknown error
10211	Unknown function has been requested
10212	Function of this controller is not available.
10221	Internal error.
10222	Invalid module number.
10223	Faulty message ID.
10224	Invalid diagnosis type
10225	Incorrect detail type
10226	Unknown detail
10227	Unknown step.
10228	Unknown sequencer.
10229	Sequencer is not disturbed.
10230	Step is not disturbed.
10231	Unknown variable requested.
10232	Unknown error ID.
10233	Unknown message number.
10234	Retain variable up or download error
10235	Retain variable up or download message
10236	Error during creating the MI import data

7.11 Error Codes 35500 to 35599

Code	Error Text	Name and Meaning of Error
35500	PARA_NOT_DEFINED	CMOS parameter not yet defined. Recovery: Describe CMOS parameter.
35501	VALUE_TYPE_INVALID	Invalid coding type.
35502	ERROR_VERSION_MISMATCH	Command does not yet exist for set IfDIIMode.

7.12 Error Codes 100000 to 100999

Code	Error Text	Name and Meaning of Error
100101	FS_NO_TEXT_FILE_ACCESS	Record file is opened in text mode.
100102	FS_REC_SIZE_TO_SMALL	Invalid record size
100103	FS_REC_FILE_BOUND_ERROR	Invalid file position
100104	FS_NO_CREATE_OBJECT	An interface object could not be created.
100105	FS_ERROR_SIM5	Without "iMTc" ID
100106	FS_ERROR_FILETYP	Different file type
100107	FS_ERROR_FILEVERSION	Current file version is larger than file version
100108	FS_ERROR_FILELENGTH	Current file length != ID length
100109	FS_ERROR_FILEDATE	File date != ID date
100110	FS_ERROR_FILETIME	File time != ID time
100111	FS_ERROR_FILENAME	File name != ID name
100112	FS_ERROR_CHECKSUM	Checksum is incorrect
100113	FS_ERROR_FILE_NOT_EXIST	File does not exist
100114	FS_ERROR_FILE_MIN_LENGTH	File with ID must be at least 65 byte.
100115	FS_ERROR_T04	Without "iT04" ID
100116	FS_ERROR_FILE_NOT_OPEN	File cannot be opened.
100117	FS_ERROR_NO_SIGN	File has no ID (sign)
100118	FS_ERROR_MMIVERSION	GUI version is smaller than file version.

7.13 Error Codes 110000 to 110999

Code	Error Text	Name and Meaning of Error
110001	BOF_MAP_VERSION_FUNC_ERROR	Incorrect Dll mode set
110002	BOF_MAP_FILE_VERSION_ERROR	Incorrect file version number
110003	BOF_MAP_LONGID_VERSION_ERROR	If the long ID version is invalid
110004	BOF_MAP_LONGID_INVALID_ERROR	If the long ID is invalid
110005	BOF_MAP_LONGID_PARAM_ERROR	Missing parameter in split long ID
110006	BOF_MAP_COMMON_ERROR	Error not clearly defined
110007	BOF_MAP_FILE_NOT_OPEN	File could not be opened.
110008	BOF_MAP_FILE_IS_OPEN	File is already open.
110009	BOF_MAP_PLAUSIBLE_TEST_ERROR	Plausibility test of map file long ID is negative.
110010	BOF_MAP_ID_ERROR	Long ID comparison is negative.
110011	BOF_MAP_TO_MANY_IMPORT_TAB_ENTRIES	Too many import table entries (>65535).
110012	BOF_MAP_INVALID_DATA	MAP file contains invalid data.
110013	BOF_MAP_PARAMETER_INVALID	Missing parameters for a function.
110014	BOF_MAP_INVALID_DEVICE_NO	Transmitted device number does not agree with the system number in the MAP.
110015	BOF_MAP_INVALID_STATUS	Invalid access status
110016	BOF_MAP_ACCESS_ERROR	Access to a MAP when MAP has not been loaded, incorrect DeviceNo
110017	BOF_MAP_NO_LOAD_ERROR	MAP file is not loaded internally
110018	BOF_MAP_NO_LOAD_MAPFILE_ERROR52	MAP file is not loaded with error 52
110019	BOF_MAP_NO_LOAD_MAPFILE_ERROR52	MAP file is not loaded with error 55
110020	BOF_MAP_MAPFILE_INVALID_VERSION_ERROR56	MAP file has invalid version 56
110021	BOF_MAP_VARIABLE_NO_FOUND_ERROR	13...46 → variable not found.
110022	BOF_MAP_LONGID_DIFFERENT_TO_MAP12	MAP file long ID is different from PLC long ID
110023	BOF_MAP_INVALID_ARRAY_INDEX50	Invalid array index
110024	BOF_MAP_INVALID_STRING_INDEX51	Invalid string index
110025	BOF_MAP_NO_CREATED_MAP_ACCESS	MAP access has not been generated
110026	BOF_MAP_LONGID_INVALID_NO_MAP_ERROR	Long ID is not valid and no PLC MAP access has yet been initialized.
110027	BOF_MAP_OUTOFMEMORY	No more memory available for creating object
110028	BOF_MAP_STRUCT_ELEMENT_NO_FOUND	Structure element does not exist
110029	BOF_MAP_STRUCT_ELEMENT_NO_FOUND	Global management information has not been created
110030	BOF_MAP_DOWNLOAD_STATUS	Access to MAP during a download
110031	BOF_MAP_NO_DOS_PCL	System is not DOS - PLC
110032	BOF_MAP_NO_WIN_PCL	System is not WIN - PLC
110033	BOF_MAP_EXCEPTION	Exception has been triggered while determining PLC address
110034	BOF_WINPCL_INSTANCE	WinPCL object instance could not be created
110035	BOF_WINPCL_CREATE	Error creating access to WinPCL

Code	Error Text	Name and Meaning of Error
110036	BOF_WINPCL_INIT	Error initializing WinPCL; current PLC long ID differs from database
110037	BOF_WINPCL_ACCESS	WinPCL access object has not been created
110038		Parameter too small
110040		An administration error occurred
110041		Initialization of the WinPCL address server not possible
110050	BOF_MAP_COMMON_FILETOOL_ERROR	Basic number cErrorGroup_filetool
110100	BOF_MAP_COMMON_MAP_BAS_C_ERROR	Basic number ErrorGroup_map_bas_c
110150	BOF_MAP_COMMON_LKENN_ERROR	Basic number cErrorGroup_lkenn
110200	BOF_MAP_COMMON_GROUP_ERROR	Basic number general error
110263	BOF_MAP_VARIABLE_NO_FOUND_ERROR_ BASE+13	PLC variable does not exist; to clearly identify the error, the error number is added to the BASE.
....	
110296	BOF_MAP_VARIABLE_NO_FOUND_ERROR_ BASE+46	

7.14 Error Codes 210900 to 210999

Code	Meaning
210917	String is too long
210920	String does not begin with '
210921	String does not end with '
210923	Counter value has been exceeded
210924	Minimum counter value not achieved
210925	Incorrect counter format

7.15 SERCOS Errors

Code	Error Messages in Serial Protocol
0x0000	No error in NC/MMI service channel.
0x0001	NC/MMI service channel not opened.
0x0009	Incorrect access to Element 0.
0x0090	The control is currently busy. The request is not possible at the moment. Please try again later.
0x00A0	"invalid request" e.g. access to S-/P parameter in initialization mode.
0x00B0	"invalid element" Only the operating date element is valid for write access.
0x00C0	"invalid drive address" The drive address is larger than allowed or the drive is not active within the SERCOS ring (deactivated or does not exist).
0x00F0	"Fatal software error" An internal CLC error has occurred during parameter transmission (see C-0-0041), that has affected the exchange of data.
0x1001	IDN does not exist.
0x1009	Incorrect access to Element 1.
0x13E8	Syntax error during transmission between PC and drive.
0x13E9	The addressed drive is not available.
0x13EA	The required data transmission was cancelled (BUSY flag set).
0x13EB	The required data channel was closed.
0x13EC	The required data channel to drive is disturbed.
0x2001	Name does not exist.
0x2002	Name transmitted too short.
0x2003	Name transmitted too long.
0x2004	Name cannot be changed.
0x2005	Name currently write-protected.
0x3002	Attribute transmitted too short.
0x3003	Attribute transmitted too long.
0x3004	Attribute cannot be changed.
0x3005	Attribute currently write-protected.
0x4001	Unit does not exist.
0x4002	Unit transmitted too short.
0x4003	Unit transmitted too long.
0x4004	Unit cannot be changed.
0x4005	Unit currently write-protected.
0x5001	Minimum input value does not exist.
0x5002	Minimum input value transmitted too short.
0x5003	Minimum input value transmitted too long.
0x5004	Minimum input value cannot be changed.
0x5005	Minimum input value currently write-protected.

Code	Error Messages in Serial Protocol
0x6001	Maximum input value does not exist.
0x6002	Maximum input value transmitted too short.
0x6003	Maximum input value transmitted too long.
0x6004	Maximum input value cannot be changed.
0x6005	Maximum input value currently write-protected.
0x7002	Date transmitted too short.
0x7003	Date transmitted too long.
0x7004	Date can not be changed.
0x7005	Date currently write-protected.
0x7006	Date smaller than min. input value.
0x7007	Date larger than max. input value.
0x7008	Incorrect date.
0x7009	Date is write-protected by password.
0x700A	The operating date is currently write-protected as it has been configured cyclically (IDN is configured with MDT or AT; therefore, writing via the service channel is not allowed)
0x700B	Invalid list element IDN is not supported, value outside of the input limits).
0x700C	Operating data write-protected at the moment because of other settings (e.g. parameter, operating mode, drive release, drive ON, etc.).
0x700D	"Length of date cannot currently be changed" The length of the date cannot be changed in the current mode.
0x700E	"Length of the date cannot currently be changed" The length of the date is permanently write-protected.
0x7010	Command already active
0x7011	Command cannot be interrupted
0x7012	Command cannot be executed at present (e.g. command cannot be activated in this phase)
0x7013	Command cannot be executed (invalid or incorrect parameters)
0x710B	Date only modifiable via control parameter
0x710C	Date outside of figure range
0x710D	Length of date cannot be changed at present
0x710E	Length of date can not be changed.
0x8001	"Service channel is currently assigned (BUSY)" The required access is not currently possible as the service channel is assigned. Data transmission is not executed.
0x8002	"Fault in service channel" Access to the required drive is not currently possible.
0x8003	"Phase switching via serial protocol is not possible" a) NC does not allow phase switching b) enabling is set for at least one drive.
0x8004	Wrong phase indication via serial protocol.
0x8005	Drive address incorrect Phase upshifting is not carried out if the addresses of the drives in the ring do not agree with the addresses whose parameters have been set.

Code	Error Messages in Serial Protocol
0x800B	Transmission has been cancelled by the control unit as it must currently communicate with the same drive (higher priority).
0x800C	Unauthorized access (service channel is still active); last transmission has not yet been completed and a new request has been started.
0xD002	"Switching: Phase 2 => 3 not possible" At least one drive refuses switching into phase 3.
0xD003	"Switching: Phase 3 => 4 not possible" At least one drive refuses switching into phase 4.
0xD005	"Phase switching still active" A new phase switching is indicated while another phase switching is not yet completed.
0xD006	Phase switching in controller enabling not allowed. "AF" is set for at least one drive.
0xD007	Phase switching during rotating lead axis is not allowed
0xD008	Phase switching during enabled lead axis function is not allowed.

7.16 Global SERCANS Errors

The global SERCANS errors are not directly related to the message transmitted. These are fatal communication errors that result in the breakdown of communication with one or more drives.

The following global SERCANS error codes have been defined:

Code	Error Messages in Serial Protocol
0x8006	HS timeout
0x8007	Double AT breakdown.
0x8008	Optical waveguide ring not closed.
0x8009	Optical waveguide ring interrupted.
0x800A	"Test operation: zero bit current or continuous light". Test operation is set on the SERCANS assembly in order to check the optical transmission route on the SERCOS interface.
0xC001	Invalid command control word.
0xC002	IDN is not a command.
0xC003	Command channel cannot currently be activated.
0xD001	Drive error (status class 1, S-0-0011).
0xD004	Command cannot be executed in drive.
0xF001	"Configuration error". An error occurred on configuration of the setpoint or actual value channel: a) too many setpoint or actual values have been configured, or the setpoint or actual values are not supported.
0xF002	"Error in calculating time slot" a) Telegram configured is too long b) Communication cycle time is too short
0xF003	Incorrect phase details from the NC
0xF004	"Error in life counter". The control no longer accesses the DPR of SERCANS cyclically.
0xF005	SERCANS: internal error

Code	Error Messages in Serial Protocol
0xF006	"Copy times too long". The copy times of the command values and actual values taken together are larger than the time between the end of the last ATs and the beginning of the MDTs.
0xF007	Checksum error (Y parameter).
0xF008	Breakdown of input signal SYNCIN
0xF009	Error in storing the system parameter or the system parameter has been changed. A check of the min/max values failed.
0xF00A	Parameter is write-protected.

7.17 Structure of Error File after Download

If an error is generated during downloading, this error is recorded in an error file and the *DownloadError* key is set to the value "YES" in the download file. The error file corresponds to the download file with the ending ".ERR".

An error is assigned to the section in which it occurs. The following error keys are set.

XX indicates a serial number

Key: ErrorLine_XX

Line in which the error has occurred (optional).

Key: ErrorLink_XX

Shows the connection with the error.

Key: ErrorSection_XX

Identifies the section in which the error has occurred.

Key: ErrorText_XX

Error text generated during the program flow.

Key: ErrorToken_XX

Error number

Key: TextFileName_XX

Name of the text file used (without language extension).

Examples for the MWCX Device Group

NC Cycle Download: CCA

Example: Download file

[Common]

ErrorSection_01= ListOfCycPrograms

ErrorToken_01 = 1014

[ListOfCycPrograms]

ErrorLink_01= K:\Program Files\Indramat\Mtgui\Project_000\Cyc-Prg-00-01.dat

ErrorToken_01 = 1014

Example: NC cycle program file

[Data]

ErrorColumn_01 = 7

ErrorLine_01 = 6

ErrorText_01 = Format error in the NC program

ErrorToken_01 = 1014

NC D-Correction Download: DCA	<p><u>Example:</u> Download file</p> <p>[Common]</p> <p>;Errors of a general kind. Cannot be assigned to a specific section. E.g., Compiler not found</p> <p>ErrorToken_01 = 1234</p> <p>ErrorText_01="Download error"</p> <p>ErrorSection_01= DCorrection_1</p> <p>[DCorrectionPackage_Info]</p> <p>ErrorToken_01=5678</p> <p>ErrorText_01=Missing info value</p> <p>[DCorrection_1]</p> <p>ErrorToken_01=1014</p> <p>ErrorText_01=Transmission error</p>
NC Program Download: NCA	<p><u>Example:</u> Download file</p> <p>[Common]</p> <p>ErrorSection_01= ListOfNCPrograms</p> <p>ErrorToken_01 = 1014</p> <p>[ListOfNCPrograms]</p> <p>ErrorLink_01= K:\Program Files\Indramat\Mtgui\Project_000\NC-PRG-00-01.Dat</p> <p>ErrorToken_01 = 1014</p> <p><u>Example:</u> NC program file</p> <p>[Data]</p> <p>ErrorColumn_01 = 7</p> <p>ErrorLine_01 = 6</p> <p>ErrorText_01 = Format error in the NC program</p> <p>ErrorToken_01 = 1014</p>
NC Events Download: NEA	<p><u>Example:</u> Download file</p> <p>[Common]</p> <p>;Errors of a general kind. Cannot be assigned to a specific section. E.g., Compiler not found</p> <p>ErrorToken_01 = 1234</p> <p>ErrorText_01="Download error"</p> <p>ErrorSection_01= NCEvents_1</p> <p>[NCEventsPackage_Info]</p> <p>ErrorToken_01=5678</p> <p>ErrorText_01=Missing info value</p> <p>[NCEvents_1]</p> <p>ErrorToken_01=1014</p> <p>ErrorText_01=Transmission error</p>

NC Zero Point Download: NUA Example: Download file
[Common]
;Errors of a general kind. Cannot be assigned to a specific section. E.g.,
Compiler not found
ErrorToken_01 = 1234
ErrorText_01="Download error"
ErrorSection_01= OffsetDataPackage_Info

[OffsetDataPackage_Info]
ErrorToken_01=5678
ErrorText_01=Missing info value

[OffsetData_0\0\0] ; process 0, zero point data base 0, axis X
ErrorToken_01=1014
ErrorText_01=Transmission error

NC Variables Download: NVA Example: Download file
[Common]
;Errors of a general kind. Cannot be assigned to a specific section. E.g.,
Compiler not found
ErrorToken_01 = 1234
ErrorText_01="Download error"
ErrorSection_01= NCVariables_1

[NCVariablesPackage_Info]
ErrorToken_01=5678
ErrorText_01=Missing info value

[NCVariables_1]
ErrorToken_01=1014
ErrorText_01=Transmission error

8 Answers to Frequently Asked Questions

8.1 Function Interface FAQs

In this chapter, you'll find a collection of **Frequently Asked Questions** from our customers' feedback on the Bosch Rexroth Function Interface.

Question 1 A message box appears when starting my application. Has the message box been issued by the function interface?

Note: As message boxes are entered in the Windows NT Task Manager as "applications", it is easy to see what has actually issued the message box.

Answer To do this, open Windows NT Task Manager e.g. using key combination: <Ctrl>+<Shift>+<Esc>
Highlight the message box entry in the "applications" tab page and click with the right-hand mouse button.

Note: The key combination <Ctrl>+<F10> does not work here for the right mouse button!

Select the "Switch to Process" command in the context menu that opens for the highlighted object.

If one of the following processes is displayed

- LOGINTFC.exe
- BOFINTFC.exe
- COMINTFC.exe

then this is a basic process of the function interface.

Question 2 Can group requests also be issued via the "DataTransfer" routine?

Answer No, the "DataTransfer" routine is only for issuing single read or write requests. Group requests are issued via the routines for cyclic reading via pipes.

Question 3 Why does the login procedure for my application to the function interface take so long?

Answer During the function interface initialization phase numerous safety checks are carried out.

8.2 Windows NT FAQs

This chapter contains FAQs regarding Windows NT from customer feedback.

Question 1 How can I log in with my name and password automatically (AutoLogin)?

Answer You must make the following entries in the Windows NT registry using the registry editor "REGEDT32" under key

HKEY_LOCAL_MACHINE\ Software Microsoft\ Windows NT\ Current Version Winlogon:

Value	Type	Content	Info
AutoAdminLogon	REG_SZ	1	Switch AutoLogin on/off
DefaultUserName	REG_SZ	<user name>	User login name
DefaultPassword	REG_SZ	<password>	User password (a password must exist)
DefaultDomainName	REG_SZ	<domain name>	Login must be carried out on another computer

Notes: No further message box will appear. If you want to log in using another name then you must keep the <Shift> key pressed during the starting procedure. You will now be prompted to enter your name and password.

If no password is entered in the registry then AutoLogin will only function once and Windows will then reset "AutoAdminLogon" to "0". Entering the password is absolutely essential. Please note that the password is then visible in the registry for anyone to see!

9 Index

A

AutoLogin 8-2
Availability 3-1

B

Basic process 4-1
Bosch Rexroth GUI WIN200 4-4
Bosch Rexroth system configurator 6-6

C

Calling the Task Manager 6-2
Communication process 4-1
Complete reinitialization 5-38
Core 6-10
Core mode 6-10
Core-mode driver 6-10

D

DataTransfer 5-8
Diagnosis 5-42
Diagnostic messages 5-42
DLL 3-1
Dual Port RAM 3-1, 4-2

I

INDIF000.H 5-2
INDIFX00.H 5-2

K

Key combination <Alt>+<Tab> 6-2

L

LOG channels 4-1
Logic channel 4-1
Logic process 4-1
LogInIf 5-4
LogOutIf 5-7

M

Management process 4-1, 5-42
MPI (Multi-Protocol-Interface) 4-3
MTCNC001.SYS 6-10

N

Non-pre-emptive Multitasking 6-1

O

Objective 3-1

P

Parameterized device 4-2
Physical communication address 3-1

Pre-emptive Multitasking 6-1
Problem 5-37
Program languages 5-1

R

Remedy 5-37
Requirements 3-1
Right mouse click 5-43

S

Serial interface 3-1
Shared memory 4-2
Shared Memory area 3-1
Simulation pair 6-6
Start bar 5-43
SYS-MSGs 5-1
System configuration 6-6
System configurator 6-6
System Messages 5-1

T

Tab pages 5-43
Task-Manager 6-2

W

Windows NT core-mode driver 6-10
Windows NT Task Manager 8-1
WinHMI 4-4
WinMTC 4-4

10 Service & Support

10.1 Helpdesk

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main steht Ihnen mit Rat und Tat zur Seite. Sie erreichen uns

Our service helpdesk at our headquarters in Lohr am Main, Germany can assist you in all kinds of inquiries. Contact us

- telefonisch - by phone:
über Service Call Entry Center
- via Service Call Entry Center
- per Fax - by fax:
- per e-Mail - by e-mail: service.svc@boschrexroth.de

49 (0) 9352 40 50 60

Mo-Fr 07:00-18:00
Mo-Fr 7:00 am - 6:00 pm

+49 (0) 9352 40 49 41

10.2 Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service direkt ansprechbar unter

After helpdesk hours, contact our service department directly at

+49 (0) 171 333 88 26

oder - or

+49 (0) 172 660 04 06

10.3 Internet

Unter www.boschrexroth.com finden Sie ergänzende Hinweise zu Service, Reparatur und Training sowie die **aktuellen** Adressen *) unserer auf den folgenden Seiten aufgeführten Vertriebs- und Servicebüros.

- Verkaufsniederlassungen
- Niederlassungen mit Kundendienst

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit unserem für Sie nächstgelegenen Ansprechpartner auf.

*) Die Angaben in der vorliegenden Dokumentation können seit Drucklegung überholt sein.

At www.boschrexroth.com you may find additional notes about service, repairs and training in the Internet, as well as the **actual** addresses *) of our sales- and service facilities figuring on the following pages.

- sales agencies
- offices providing service

Please contact our sales / service office in your area first.

*) Data in the present documentation may have become obsolete since printing.

10.4 Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

1. detaillierte Beschreibung der Störung und der Umstände.
2. Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.
3. Tel./Faxnummern und e-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind.

For quick and efficient help, please have the following information ready:

1. Detailed description of the failure and circumstances.
2. Information on the type plate of the affected products, especially type codes and serial numbers.
3. Your phone/fax numbers and e-mail address, so we can contact you in case of questions.

10.5 Kundenbetreuungsstellen - Sales & Service Facilities

Deutschland - Germany

vom Ausland:

(0) nach Landeskennziffer weglassen!

from abroad:

don't dial (0) after country code!

<p>Vertriebsgebiet Mitte Germany Centre</p> <p>Rexroth Indramat GmbH Bgm.-Dr.-Nebel-Str. 2 / Postf. 1357 97816 Lohr am Main / 97803 Lohr</p> <p>Kompetenz-Zentrum Europa</p> <p>Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885</p>	<p>SERVICE</p> <p>CALL ENTRY CENTER MO – FR von 07:00 - 18:00 Uhr from 7 am – 6 pm</p> <p>Tel. +49 (0) 9352 40 50 60 service.svc@boschrexroth.de</p>	<p>SERVICE</p> <p>HOTLINE MO – FR von 17:00 - 07:00 Uhr from 5 pm - 7 am</p> <p>+ SA / SO</p> <p>Tel.: +49 (0)172 660 04 06 oder / or Tel.: +49 (0)171 333 88 26</p>	<p>SERVICE</p> <p>ERSATZTEILE / SPARES verlängerte Ansprechzeit - extended office time -</p> <ul style="list-style-type: none"> ◆ nur an Werktagen - only on working days - ◆ von 07:00 - 18:00 Uhr - from 7 am - 6 pm - <p>Tel. +49 (0) 9352 40 42 22</p>
<p>Vertriebsgebiet Süd Germany South</p> <p>Bosch Rexroth AG Landshuter Allee 8-10 80637 München</p> <p>Tel.: +49 (0)89 127 14-0 Fax: +49 (0)89 127 14-490</p>	<p>Vertriebsgebiet West Germany West</p> <p>Bosch Rexroth AG Regionalzentrum West Borsigstrasse 15 40880 Ratingen</p> <p>Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406 +49 (0)2102 409-430</p>	<p>Gebiet Südwest Germany South-West</p> <p>Bosch Rexroth AG Service-Regionalzentrum Süd-West Siemensstr.1 70736 Fellbach</p> <p>Tel.: +49 (0)711 51046-0 Fax: +49 (0)711 51046-248</p>	
<p>Vertriebsgebiet Nord Germany North</p> <p>Bosch Rexroth AG Walsroder Str. 93 30853 Langenhagen</p> <p>Tel.: +49 (0) 511 72 66 57-0 Service: +49 (0) 511 72 66 57-256 Fax: +49 (0) 511 72 66 57-93 Service: +49 (0) 511 72 66 57-783</p>	<p>Vertriebsgebiet Mitte Germany Centre</p> <p>Bosch Rexroth AG Regionalzentrum Mitte Waldecker Straße 13 64546 Mörfelden-Walldorf</p> <p>Tel.: +49 (0) 61 05 702-3 Fax: +49 (0) 61 05 702-444</p>	<p>Vertriebsgebiet Ost Germany East</p> <p>Bosch Rexroth AG Beckerstraße 31 09120 Chemnitz</p> <p>Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333</p>	<p>Vertriebsgebiet Ost Germany East</p> <p>Bosch Rexroth AG Regionalzentrum Ost Walter-Köhn-Str. 4d 04356 Leipzig</p> <p>Tel.: +49 (0)341 25 61-0 Fax: +49 (0)341 25 61-111</p>

Europa (West) - Europe (West)

vom Ausland: (0) nach Landeskenziffer weglassen, **Italien:** 0 nach Landeskenziffer mitwählen
from abroad: don't dial (0) after country code, **Italy:** dial 0 after country code

Austria - Österreich Bosch Rexroth GmbH Electric Drives & Controls Stachegasse 13 1120 Wien Tel.: +43 (0)1 985 25 40 Fax: +43 (0)1 985 25 40-93	Austria – Österreich Bosch Rexroth GmbH Electric Drives & Controls Industriepark 18 4061 Pasching Tel.: +43 (0)7221 605-0 Fax: +43 (0)7221 605-21	Belgium - Belgien Bosch Rexroth NV/SA Henri Genessestraat 1 1070 Bruxelles Tel: +32 (0) 2 582 31 80 Fax: +32 (0) 2 582 43 10 info@boschrexroth.be service@boschrexroth.be	Denmark - Dänemark BEC A/S Zinkvej 6 8900 Randers Tel.: +45 (0)87 11 90 60 Fax: +45 (0)87 11 90 61
Great Britain – Großbritannien Bosch Rexroth Ltd. Electric Drives & Controls Broadway Lane, South Cerney Cirencester, Glos GL7 5UH Tel.: +44 (0)1285 863000 Fax: +44 (0)1285 863030 sales@boschrexroth.co.uk service@boschrexroth.co.uk	Finland - Finnland Bosch Rexroth Oy Electric Drives & Controls Ansatie 6 017 40 Vantaa Tel.: +358 (0)9 84 91-11 Fax: +358 (0)9 84 91-13 60	France - Frankreich Bosch Rexroth SAS Electric Drives & Controls Avenue de la Trentaine (BP. 74) 77503 Chelles Cedex Tel.: +33 (0)164 72-70 00 Fax: +33 (0)164 72-63 00 Hotline: +33 (0)608 33 43 28	France - Frankreich Bosch Rexroth SAS Electric Drives & Controls ZI de Thibaud, 20 bd. Thibaud (BP. 1751) 31084 Toulouse Tel.: +33 (0)5 61 43 61 87 Fax: +33 (0)5 61 43 94 12
France – Frankreich Bosch Rexroth SAS Electric Drives & Controls 91, Bd. Irène Joliot-Curie 69634 Vénissieux – Cedex Tel.: +33 (0)4 78 78 53 65 Fax: +33 (0)4 78 78 53 62	Italy - Italien Bosch Rexroth S.p.A. Via G. Di Vittorio, 1 20063 Cernusco S/N.MI Hotline: +39 02 92 365 563 Tel.: +39 02 92 365 1 Service: +39 02 92 365 326 Fax: +39 02 92 365 500 Service: +39 02 92 365 503	Italy - Italien Bosch Rexroth S.p.A. Via Paolo Veronesi, 250 10148 Torino Tel.: +39 011 224 88 11 Fax: +39 011 224 88 30	Italy - Italien Bosch Rexroth S.p.A. Via Mascia, 1 80053 Castellamare di Stabia NA Tel.: +39 081 8 71 57 00 Fax: +39 081 8 71 68 85
Italy - Italien Bosch Rexroth S.p.A. Via del Progresso, 16 (Zona Ind.) 35020 Padova Tel.: +39 049 8 70 13 70 Fax: +39 049 8 70 13 77	Italy - Italien Bosch Rexroth S.p.A. Via Isonzo, 61 40033 Casalecchio di Reno (Bo) Tel.: +39 051 29 86 430 Fax: +39 051 29 86 490	Netherlands - Niederlande/Holland Bosch Rexroth Services B.V. Technical Services Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0) 411 65 16 40 +31 (0) 411 65 17 27 Fax: +31 (0) 411 67 78 14 +31 (0) 411 68 28 60 services@boschrexroth.nl	Netherlands – Niederlande/Holland Bosch Rexroth B.V. Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0) 411 65 19 51 Fax: +31 (0) 411 65 14 83 www.boschrexroth.nl
Norway - Norwegen Bosch Rexroth AS Electric Drives & Controls Berghagan 1 or: Box 3007 1405 Ski-Langhus 1402 Ski Tel.: +47 (0) 64 86 41 00 Fax: +47 (0) 64 86 90 62 Hotline: +47 (0)64 86 94 82 jul.ruud@rexroth.no	Spain - Spanien Bosch Rexroth S.A. Electric Drives & Controls Centro Industrial Santiga Obradors s/n 08130 Santa Perpetua de Mogoda Barcelona Tel.: +34 9 37 47 94 00 Fax: +34 9 37 47 94 01	Spain – Spanien Goimendi S.A. Electric Drives & Controls Parque Empresarial Zuatzu C/ Francisco Grandmontagne no.2 20018 San Sebastian Tel.: +34 9 43 31 84 21 - service: +34 9 43 31 84 56 Fax: +34 9 43 31 84 27 - service: +34 9 43 31 84 60 sat.indramat@goimendi.es	Sweden - Schweden Bosch Rexroth AB Electric Drives & Controls - Varuvägen 7 (Service: Konsumentvägen 4, Älfsjö) 125 81 Stockholm Tel.: +46 (0)8 727 92 00 Fax: +46 (0)8 647 32 77
Sweden - Schweden Bosch Rexroth AB Electric Drives & Controls Ekvändan 7 254 67 Helsingborg Tel.: +46 (0) 42 38 88 -50 Fax: +46 (0) 42 38 88 -74	Switzerland East - Schweiz Ost Bosch Rexroth Schweiz AG Electric Drives & Controls Hemrietstrasse 2 8863 Buttikon Tel. +41 (0) 55 46 46 111 Fax +41 (0) 55 46 46 222	Switzerland West - Schweiz West Bosch Rexroth Suisse SA Av. Général Guisan 26 1800 Vevey 1 Tel.: +41 (0)21 632 84 20 Fax: +41 (0)21 632 84 21	

Europa (Ost) - Europe (East)

vom Ausland: (0) nach Landeskenziffer weglassen
from abroad: don't dial (0) after country code

<p>Czech Republic - Tschechien</p> <p>Bosch -Rexroth, spol.s.r.o. Hviezdoslavova 5 627 00 Brno Tel.: +420 (0)5 48 126 358 Fax: +420 (0)5 48 126 112</p>	<p>Czech Republic - Tschechien</p> <p>DEL a.s. Strojirenská 38 591 01 Zdar nad Sázavou Tel.: +420 566 64 3144 Fax: +420 566 62 1657</p>	<p>Hungary - Ungarn</p> <p>Bosch Rexroth Kft. Angol utca 34 1149 Budapest Tel.: +36 (1) 422 3200 Fax: +36 (1) 422 3201</p>	<p>Poland – Polen</p> <p>Bosch Rexroth Sp.zo.o. ul. Staszica 1 05-800 Pruszków Tel.: +48 22 738 18 00 – service: +48 22 738 18 46 Fax: +48 22 758 87 35 – service: +48 22 738 18 42</p>
<p>Poland – Polen</p> <p>Bosch Rexroth Sp.zo.o. Biuro Poznan ul. Dabrowskiego 81/85 60-529 Poznan Tel.: +48 061 847 64 62 /-63 Fax: +48 061 847 64 02</p>	<p>Romania - Rumänien</p> <p>East Electric S.R.L. Bdul Basarabia no.250, sector 3 73429 Bucuresti Tel./Fax: +40 (0)21 255 35 07 +40 (0)21 255 77 13 Fax: +40 (0)21 725 61 21 eastel@rdsnet.ro</p>	<p>Romania - Rumänien</p> <p>Bosch Rexroth Sp.zo.o. Str. Drobety nr. 4-10, app. 14 70258 Bucuresti, Sector 2 Tel.: +40 (0)1 210 48 25 +40 (0)1 210 29 50 Fax: +40 (0)1 210 29 52</p>	<p>Russia - Russland</p> <p>Bosch Rexroth OOO Wjatskaja ul. 27/15 127015 Moskau Tel.: +7-095-785 74 78 +7-095 785 74 79 Fax: +7 095 785 74 77 laura.kanina@boschrexroth.ru</p>
<p>Russia - Russland</p> <p>ELMIS 10, Internationalnaya 246640 Gomel, Belarus Tel.: +375/ 232 53 42 70 +375/ 232 53 21 69 Fax: +375/ 232 53 37 69 elmis_ltd@yahoo.com</p>	<p>Turkey - Türkei</p> <p>Bosch Rexroth Otomasyon San & Tic. A..S. Fevzi Cakmak Cad No. 3 34630 Sefaköy Istanbul Tel.: +90 212 413 34 00 Fax: +90 212 413 34 17 www.boschrexroth.com.tr</p>	<p>Turkey - Türkei</p> <p>Servo Kontrol Ltd. Sti. Perpa Ticaret Merkezi B Blok Kat: 11 No: 1609 80270 Okmeydani-Istanbul Tel: +90 212 320 30 80 Fax: +90 212 320 30 81 remzi.sali@servokontrol.com www.servokontrol.com</p>	<p>Slovenia - Slowenien</p> <p>DOMEL Otoki 21 64 228 Zelezniki Tel.: +386 5 5117 152 Fax: +386 5 5117 225 brane.ozebek@domel.si</p>

Africa, Asia, Australia – incl. Pacific Rim

<p>Australia - Australien</p> <p>AIMS - Australian Industrial Machinery Services Pty. Ltd. 28 Westside Drive Laverton North Vic 3026 Melbourne</p> <p>Tel.: +61 3 93 14 3321 Fax: +61 3 93 14 3329 Hotlines: +61 3 93 14 3321 +61 4 19 369 195 enquires@aimservices.com.au</p>	<p>Australia - Australien</p> <p>Bosch Rexroth Pty. Ltd. No. 7, Endeavour Way Braeside Victoria, 31 95 Melbourne</p> <p>Tel.: +61 3 95 80 39 33 Fax: +61 3 95 80 17 33 mel@rexroth.com.au</p>	<p>China</p> <p>Shanghai Bosch Rexroth Hydraulics & Automation Ltd. Waigaoqiao, Free Trade Zone No.122, Fu Te Dong Yi Road Shanghai 200131 - P.R.China</p> <p>Tel.: +86 21 58 66 30 30 Fax: +86 21 58 66 55 23 richard.yang_sh@boschrexroth.com.cn qf.zhu_sh@boschrexroth.com.cn</p>	<p>China</p> <p>Shanghai Bosch Rexroth Hydraulics & Automation Ltd. 4/f, Marine Tower No.1, Pudong Avenue Shanghai 200120 - P.R.China</p> <p>Tel.: +86 21 68 86 15 88 Fax: +86 21 58 40 65 77</p>
<p>China</p> <p>Bosch Rexroth China Ltd. 15/F China World Trade Center 1, Jianguomenwai Avenue Beijing 100004, P.R.China</p> <p>Tel.: +86 10 65 05 03 80 Fax: +86 10 65 05 03 79</p>	<p>China</p> <p>Bosch Rexroth China Ltd. Guangzhou Repres. Office Room 1014-1016, Metro Plaza, Tian He District, 183 Tian He Bei Rd Guangzhou 510075, P.R.China</p> <p>Tel.: +86 20 8755-0030 +86 20 8755-0011 Fax: +86 20 8755-2387</p>	<p>China</p> <p>Bosch Rexroth (China) Ltd. A-5F., 123 Lian Shan Street Sha He Kou District Dalian 116 023, P.R.China</p> <p>Tel.: +86 411 46 78 930 Fax: +86 411 46 78 932</p>	<p>China</p> <p>Melchers GmbH BRC-SE, Tightening & Press-fit 13 Floor Est Ocean Centre No.588 Yanan Rd. East 65 Yanan Rd. West Shanghai 200001</p> <p>Tel.: +86 21 6352 8848 Fax: +86 21 6351 3138</p>
<p>Hongkong</p> <p>Bosch Rexroth (China) Ltd. 6th Floor, Yeung Yiu Chung No.6 Ind Bldg. 19 Cheung Shun Street Cheung Sha Wan, Kowloon, Hongkong</p> <p>Tel.: +852 22 62 51 00 Fax: +852 27 41 33 44 alexis.siu@boschrexroth.com.hk</p>	<p>India - Indien</p> <p>Bosch Rexroth (India) Ltd. Electric Drives & Controls Plot. No.96, Phase III Peenya Industrial Area Bangalore – 560058</p> <p>Tel.: +91 80 51 17 0-211...-218 Fax: +91 80 83 94 345 +91 80 83 97 374 mohanvelu.t@boschrexroth.co.in</p>	<p>India - Indien</p> <p>Bosch Rexroth (India) Ltd. Electric Drives & Controls Advance House, II Floor Ark Industrial Compound Narol Naka, Makwana Road Andheri (East), Mumbai - 400 059</p> <p>Tel.: +91 22 28 56 32 90 +91 22 28 56 33 18 Fax: +91 22 28 56 32 93 singh.op@boschrexroth.co.in</p>	<p>India - Indien</p> <p>Bosch Rexroth (India) Ltd. S-10, Green Park Extension New Delhi – 110016</p> <p>Tel.: +91 11 26 56 65 25 +91 11 26 56 65 27 Fax: +91 11 26 56 68 87 koul.rp@boschrexroth.co.in</p>
<p>Indonesia - Indonesien</p> <p>PT. Bosch Rexroth Building # 202, Cilandak Commercial Estate Jl. Cilandak KKO, Jakarta 12560</p> <p>Tel.: +62 21 7891169 (5 lines) Fax: +62 21 7891170 - 71 rudu.karimun@boschrexroth.co.id</p>	<p>Japan</p> <p>Bosch Rexroth Automation Corp. Service Center Japan Yutakagaoka 1810, Meito-ku, NAGOYA 465-0035, Japan</p> <p>Tel.: +81 52 777 88 41 +81 52 777 88 53 +81 52 777 88 79 Fax: +81 52 777 89 01</p>	<p>Japan</p> <p>Bosch Rexroth Automation Corp. Electric Drives & Controls 2F, I.R. Building Nakamachidai 4-26-44, Tsuzuki-ku YOKOHAMA 224-0041, Japan</p> <p>Tel.: +81 45 942 72 10 Fax: +81 45 942 03 41</p>	<p>Korea</p> <p>Bosch Rexroth-Korea Ltd. Electric Drives and Controls Bongwoo Bldg. 7FL, 31-7, 1Ga Jangchoong-dong, Jung-gu Seoul, 100-391</p> <p>Tel.: +82 234 061 813 Fax: +82 222 641 295</p>
<p>Korea</p> <p>Bosch Rexroth-Korea Ltd. 1515-14 Dadae-Dong, Saha-gu Electric Drives & Controls Pusan Metropolitan City, 604-050</p> <p>Tel.: +82 51 26 00 741 Fax: +82 51 26 00 747 eunkyong.kim@boschrexroth.co.kr</p>	<p>Malaysia</p> <p>Bosch Rexroth Sdn.Bhd. 11, Jalan U8/82, Seksyen U8 40150 Shah Alam Selangor, Malaysia</p> <p>Tel.: +60 3 78 44 80 00 Fax: +60 3 78 45 48 00 hockhwa@hotmail.com rexroth1@tm.net.my</p>	<p>Singapore - Singapur</p> <p>Bosch Rexroth Pte Ltd 15D Tuas Road Singapore 638520</p> <p>Tel.: +65 68 61 87 33 Fax: +65 68 61 18 25 sanjay.nemade@boschrexroth.com.sg</p>	<p>South Africa - Südafrika</p> <p>TECTRA Automation (Pty) Ltd. 71 Watt Street, Meadowdale Edenvale 1609</p> <p>Tel.: +27 11 971 94 00 Fax: +27 11 971 94 40 Hotline: +27 82 903 29 23 georgv@tectra.co.za</p>
<p>Taiwan</p> <p>Bosch Rexroth Co., Ltd. Taichung Branch 1F., No. 29, Fu-Ann 5th Street, Xi-Tun Area, Taichung City Taiwan, R.O.C.</p> <p>Tel.: +886 - 4 -23580400 Fax: +886 - 4 -23580402 jim.lin@boschrexroth.com.tw david.lai@boschrexroth.com.tw</p>	<p>Taiwan</p> <p>Bosch Rexroth Co., Ltd. Tainan Branch No. 17, Alley 24, Lane 737 Chung Cheng N.Rd. Yungkang Tainan Hsien, Taiwan, R.O.C.</p> <p>Tel.: +886 - 6 -253 6565 Fax: +886 - 6 -253 4754 charlie.chen@boschrexroth.com.tw</p>	<p>Thailand</p> <p>NC Advance Technology Co. Ltd. 59/76 Moo 9 Ramintra road 34 Tharang, Bangkhen, Bangkok 10230</p> <p>Tel.: +66 2 943 70 62 +66 2 943 71 21 Fax: +66 2 509 23 62 Hotline +66 1 984 61 52 sonkawin@hotmail.com</p>	

Nordamerika – North America

USA Headquarters - Hauptniederlassung Bosch Rexroth Corporation Electric Drives & Controls 5150 Prairie Stone Parkway Hoffman Estates, IL 60192-3707 Tel.: +1 847 6 45 36 00 Fax: +1 847 6 45 62 01 servicebrc@boschrexroth-us.com repairbrc@boschrexroth-us.com	USA Central Region - Mitte Bosch Rexroth Corporation Electric Drives & Controls Central Region Technical Center 1701 Harmon Road Auburn Hills, MI 48326 Tel.: +1 248 3 93 33 30 Fax: +1 248 3 93 29 06	USA Southeast Region - Südwest Bosch Rexroth Corporation Electric Drives & Controls Southeastern Technical Center 3625 Swiftwater Park Drive Suwanee, Georgia 30124 Tel.: +1 770 9 32 32 00 Fax: +1 770 9 32 19 03	USA SERVICE-HOTLINE - 7 days x 24hrs - +1-800-REX-ROTH +1 800 739 7684
USA East Region – Ost Bosch Rexroth Corporation Electric Drives & Controls Charlotte Regional Sales Office 14001 South Lakes Drive Charlotte, North Carolina 28273 Tel.: +1 704 5 83 97 62 +1 704 5 83 14 86	USA Northeast Region – Nordost Bosch Rexroth Corporation Electric Drives & Controls Northeastern Technical Center 99 Rainbow Road East Granby, Connecticut 06026 Tel.: +1 860 8 44 83 77 Fax: +1 860 8 44 85 95	USA West Region – West Bosch Rexroth Corporation 7901 Stoneridge Drive, Suite 220 Pleasant Hill, California 94588 Tel.: +1 925 227 10 84 Fax: +1 925 227 10 81	
Canada East - Kanada Ost Bosch Rexroth Canada Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Tel.: +1 905 335 5511 Fax: +1 905 335 4184 Hotline: +1 905 335 5511 michael.moro@boschrexroth.ca	Canada West - Kanada West Bosch Rexroth Canada Corporation 5345 Goring St. Burnaby, British Columbia Canada V7J 1R1 Tel.: +1 604 205 5777 Fax: +1 604 205 6944 Hotline: +1 604 205 5777 david.gunby@boschrexroth.ca	Mexico Bosch Rexroth Mexico S.A. de C.V. Calle Neptuno 72 Unidad Ind. Vallejo 07700 Mexico, D.F. Tel.: +52 55 57 54 17 11 Fax: +52 55 57 54 50 73 mariofelipe.hernandez@boschrexroth.com.mx	Mexico Bosch Rexroth S.A. de C.V. Calle Argentina No 3913 Fracc. las Torres 64930 Monterrey, N.L. Tel.: +52 81 83 65 22 53 +52 81 83 65 89 11 +52 81 83 49 80 91 Fax: +52 81 83 65 52 80 mario.quiroga@boschrexroth.com.mx

Südamerika – South America

Argentina - Argentinien Bosch Rexroth S.A.I.C. "The Drive & Control Company" Rosario 2302 B1606DLD Carapachay Provincia de Buenos Aires Tel.: +54 11 4756 01 40 +54 11 4756 02 40 +54 11 4756 03 40 +54 11 4756 04 40 Fax: +54 11 4756 01 36 +54 11 4721 91 53 victor.jabif@boschrexroth.com.ar	Argentina - Argentinien NAKASE Servicio Tecnico CNC Calle 49, No. 5764/66 B1653AOX Villa Balester Provincia de Buenos Aires Tel.: +54 11 4768 36 43 Fax: +54 11 4768 24 13 Hotline: +54 11 155 307 6781 nakase@usa.net nakase@nakase.com gerencia@nakase.com (Service)	Brazil - Brasilien Bosch Rexroth Ltda. Av. Tégula, 888 Ponte Alta, Atibaia SP CEP 12942-440 Tel.: +55 11 4414 56 92 +55 11 4414 56 84 Fax sales: +55 11 4414 57 07 Fax serv.: +55 11 4414 56 86 alexandre.wittwer@rexroth.com.br	Brazil - Brasilien Bosch Rexroth Ltda. R. Dr.Humberto Pinheiro Vieira, 100 Distrito Industrial [Caixa Postal 1273] 89220-390 Joinville - SC Tel./Fax: +55 47 473 58 33 Mobil: +55 47 9974 6645 prochnow@zaz.com.br
Columbia - Kolumbien Refflutec de Colombia Ltda. Calle 37 No. 22-31 Santafé de Bogotá, D.C. Colombia Tel.: +57 1 368 82 67 +57 1 368 02 59 Fax: +57 1 268 97 37 reflutec@neutel.com.co reflutec@007mundo.com			

Notes

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 93 52-40-50 60
Fax +49 93 52-40-49 41
service.svc@boschrexroth.de
www.boschrexroth.com

