1 Function Packet Editor – IBS G

1.1 Overview

In addition to the basic functions ‘Documenting ’’and ‘Configuring ’’, the function packet editor also permits an on-line connection to SIMADYN D to be generated through which direct access to the running processor programs can be made.

During the commissioning phase or for service purposes, it is therefore possible to display current values from the function drawings in SIMADYN D, as well as implementing small on-line configuring modifications.

These on-line functions are collected together in the basic function ‘IBS G ’’of the function packet editor.

Warning



Improper Operation of the Function Packet Editor in the IBS G Mode can cause Accidents Including Damage to Persons or Property.



1.1.1 IBS G Mode Tasks

The following on-line functions are currently supported:

- Cyclically display connector values,
- Modify constant input values (temporary or permanent),
- Delete connections (temporary or permanent),
- Create connections (temporary or permanent).
- Insert and copy (temporary)
- Delete function blocks (temporary)

A **temporary modification** will be lost at a subsequent rack reset. A **permanent modification** is however saved in non volatile memory such that it remains active after a restart.

Remark: Permanent modifications with the function packet editor have the advantage that the configured system and SIMADYN D remain at the same version and a recompile is not necessary.

A series of additional dialog possibilities have been implemented in the function packet, based upon these on-line basic functions. The following is a brief summary of these functions.

1.1.1.1 Actions in the Function Drawings

The previously available configuring possibilities have been expanded by an on-line component in the IBS G mode. This signifies that, for example, deleting a connection in the function drawing immediately causes the separation of the connection in SIMADYN D. Therefore a series of configuring modifications can be implemented at the same time in SIMADYN D.

The display of current values in the function drawing has been implemented by an additional display element, the so called **Value Display Window** (see section 1.6).

1.1.1.2 Actions in the Tables

The on-line basic functions described in the section 1.1.1 can also be implemented in table form, independent of the function drawing. The so called **Connector Tables** (see section 1.7) and the **Connection Table** (see section 1.9) are available for this purpose.

1.1.1.3 Formula

The **formula table** (see section 1.10) is utilized for the simultaneous modification of a series of connector values. Loading pre-defined parameter sets in SIMADYN D is therefore just as possible as the simple copying of parameters into several similarly structured sections within a configuration.

1.1.1.4 Archiving Value Responses

The value responses occurring at the connectors can be recorded with the aid of the **archiving function** (see section 1.11) and graphically displayed off-line.

Any number of connectors can be stored in a file and any number of archive files simultaneously processed both in write and read modes.

1.1.1.5 Log Book

All on-line actions are logged in a log book (see section 1.12). The log book therefore serves two purposes, to check which actions have been implemented by the function packet editor in on-line mode with SIMADYN D and to carry out an automatic update of temporary value modifications. All temporary value modifications have been made permanent subsequent to a so called **EEPROM update**.

1.1.2 Function packet Editor Environment in the On-Line Mode

The IBS G mode in the function packet editor permits an access to all processors in one or several racks simultaneously. The connection to SIMADYN D is implemented via a serial connection (DUST1) or via the network SINEC H1.
In the IBS G mode, the function packet editor can simultaneously implement accessing to all processors in a rack. The connection to SIMADYN D is implemented via a serial link (DUST1) with a fixed baud rate of 9600 baud.

The connection from the function packet editor to SIMADYN D is supported by two processes. The process 'HWA" (Hardware Connection) implements the connection from the PC to SIMADYN D. Incoming commands are reported to SIMADYN D, return messages from SIMADYN D are transferred to the next process. The process 'DA" (Data Processing) implements the analysis and the saving of the incoming data from
SIMADYN D.

The current process data is saved for the function packet editor in a shared memory area and archived to an external file when required (see section1.11).

The process structure in IBS G mode permits the process 'HWA"to be run on a second processor. This makes the actual separation of the configuring station location from the processor containing the hardware interface to SIMADYN D possible (see section 1.13.4.2).

1.1.3 Configuring Regulations for IBS G

A service block (P32) or monitor block (P16) must be configured on each processor to which on-line accessing is required. The access to this type of block can, dependent upon the configuring, be implemented from the IBS G via the following transport path alternatives:

* Direct connection via a DUST1 channel pair,
* Connection via the SIMADYN D network (peer to peer) and via a DUST1 channel pair,
* Connection via the SIMADYN D network (routing) and also via an utilized DUST1 channel pair,
* Direct connection via a SINEC H1 transport connection,

* Connection via the SIMADYN D network (peer to peer) and via a SINEC H1 transport connection,
* Connection via the SIMADYN D network (routing) and also via an utilized SINEC H1 transport connection.

The configured address at the service or monitor blocks must conform to those specified in the basic dialog (menu, options, screens, IBS G connections and communication islands); see section 7.5. Further information regarding the communication configuring can be reviewed in the configuring guide "Communication ".

1.1.4 Consistency between Configured Data and the Processor Program

It is especially important for the on-line insert/copying of function blocks (see chapter 6.4.2), that the configured data and the processor program are conform. Otherwise certain circumstances of incorrect locations can occur (see sequence number) of the newly inserted blocks.

The version number and release of the block library (FBSLIB) of both the configuring and the processor program are compared with one another at the start of the IBS G (start on-line). It must be ensured that the library on the configured processor has the same release version as the library with which the processor program was created. Otherwise an error message will be generated and it will not be possible to insert/copy/delete function blocks on-line.

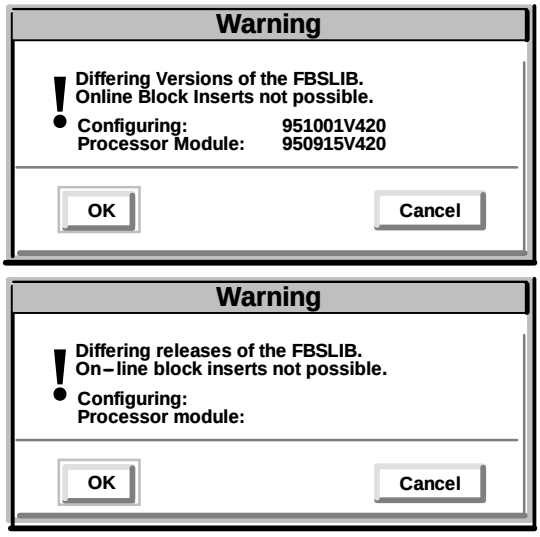


Diagram 1/1: Warning Window when differing Block Library Releases are Detected

The date of the last compiler run (PNCOP – Date) is compared with the date of the processor program at the start of the IBS G as well as at each switch of the current function packet. A differing date can imply an inconsistency between the configuration and processor program and will be reported by a warning window

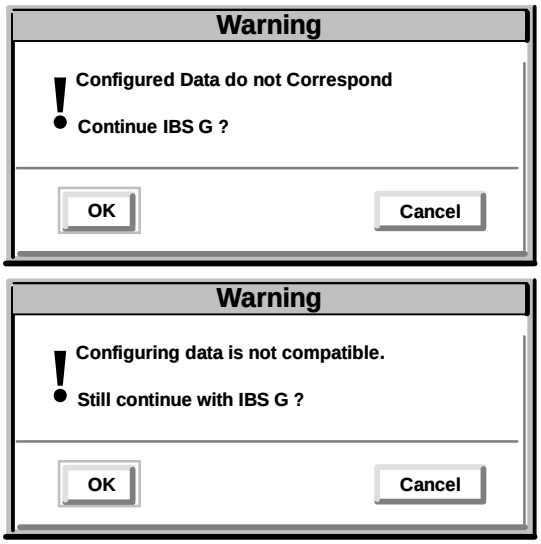


Diagram 1/2: Warning Window when differing PNCOP Dates Detected

1.1.5 Restrictions to the Configuring Possibilities in the On-Line Mode

Only a section of the configuring possibilities can be carried out on-line by the function packet editor in the IBS G mode. The selection of a basic function that is not supported on-line in SIMADYN D will lead to a warning (see Diagram 1/3). If the configuring function is implemented inspite of the warning, then the configured data and the processor program in SIMADYN D will no longer match.

If the configuring modification is then to be activated in SIMADYN D, the modified function packet and the associated processor program must be recompiled and a new memory sub-module programmed.

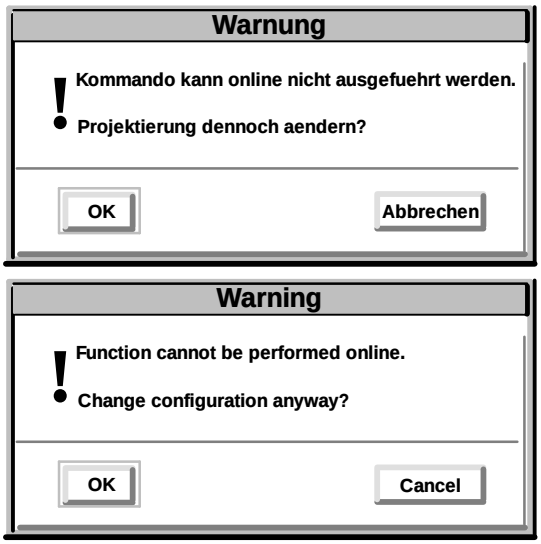


Diagram 1/3: Warning Window for Configuring Restrictions

Restrictions to the configuring possibilities in on-line mode currently exist for the following basic functions:

- FB/CONN – Call Block,
- FB/CONN – Copy Block,

- FB/CONN – Configure Block,
- FB/CONN – Delete Block,
- FB/CONN – Copy Connector Comment,
- FB/CONN – Switch Connector Type,
- FB/CONN – Copy FB Group,
- FB/CONN – Delete FB Group,
- CONNECT – Configure,
- CONNECT – Change,
- SHEET – Insert before ...,
- SHEET – Extract ...,
- SHEET – Append ...,
- SHEET – Title Panel.

1.1.6 Connector Addresses

The designation of connectors in the function packet editor IBS G is implemented in six stages:

* Network Island Name
* Rack Name
* Processor Number
* Function Packet Name
* Function Block Name
* Connector Designator

The front two sections of the connector addresses are utilized for addressing the island in the SINEC H1 network as well as the rack in the routing SIMADYN D network. All specifications must be available for a valid connector address. Matching to the STRUC G configured name is always necessary.

The list selection (see section 1.5.4) requires the definition of the available network node, racks and processors to be entered into the configuration file for the topology definitions (see section 1.13.2).

The address components are output with ‘:’ separators. A complete connector address is as follows, i.e.:

$$NODE1.Rack1.3.SAWTTH.B00040.X$$

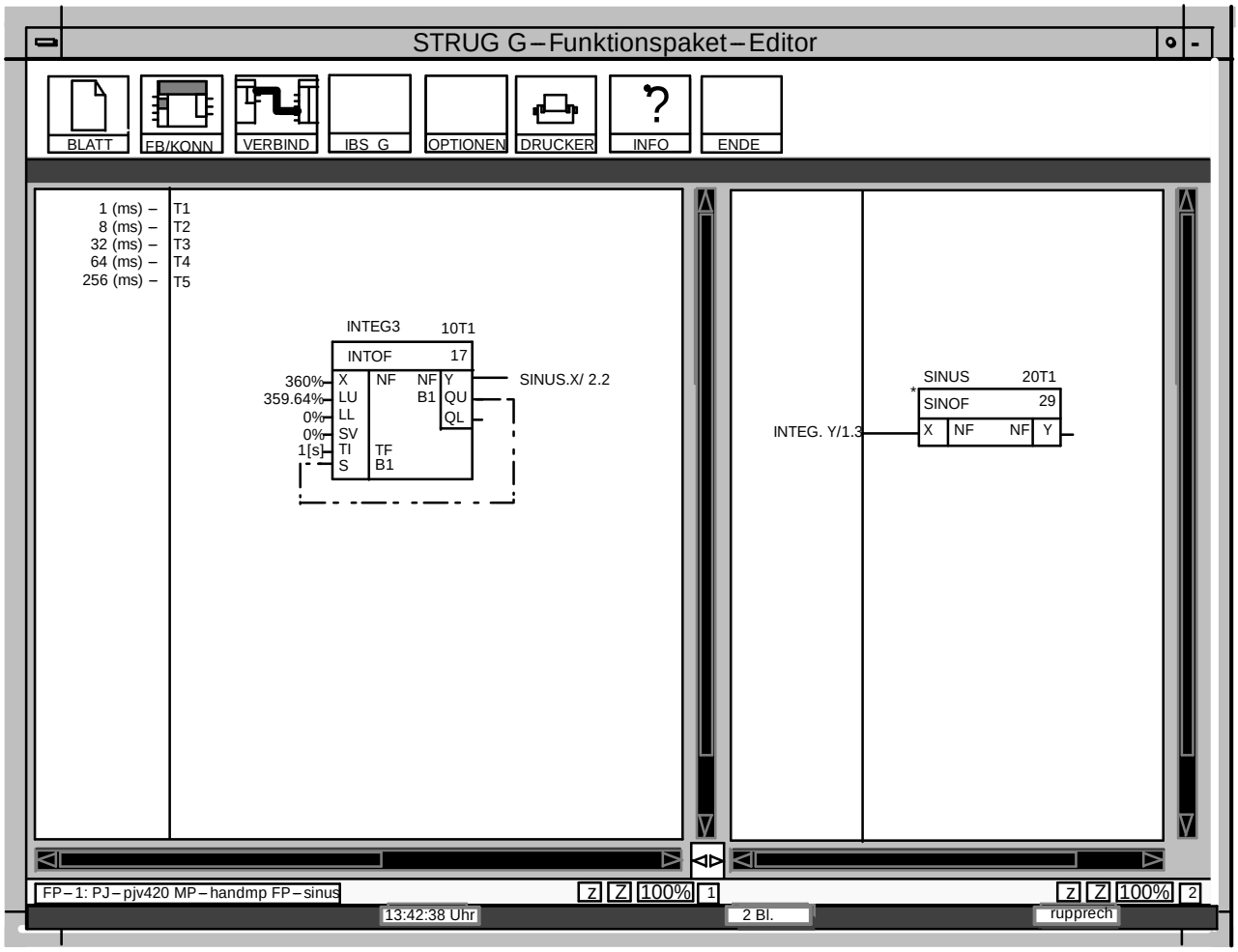
Alternatively, the output of connector addresses can also occur **symbolically**. Specifications must therefore be entered into the configuration file for symbolic path designators (see section 1.13.1). Therefore the specification of, i.e.

$$NODE1.Rack1.3.SAWTTH.B00040.X = Rated\ Speed$$

permits the output of one of these two designators in both the connector tables and also the value display windows.

1.2 Screen Layout

The screen layout of the function packet editor in the IBS G mode is expanded by an extended static menu with respect to the screen layout for configuring. The basic functions for the IBS G can be invoked via a separate menu. All other dialog elements are also available with IBS G.



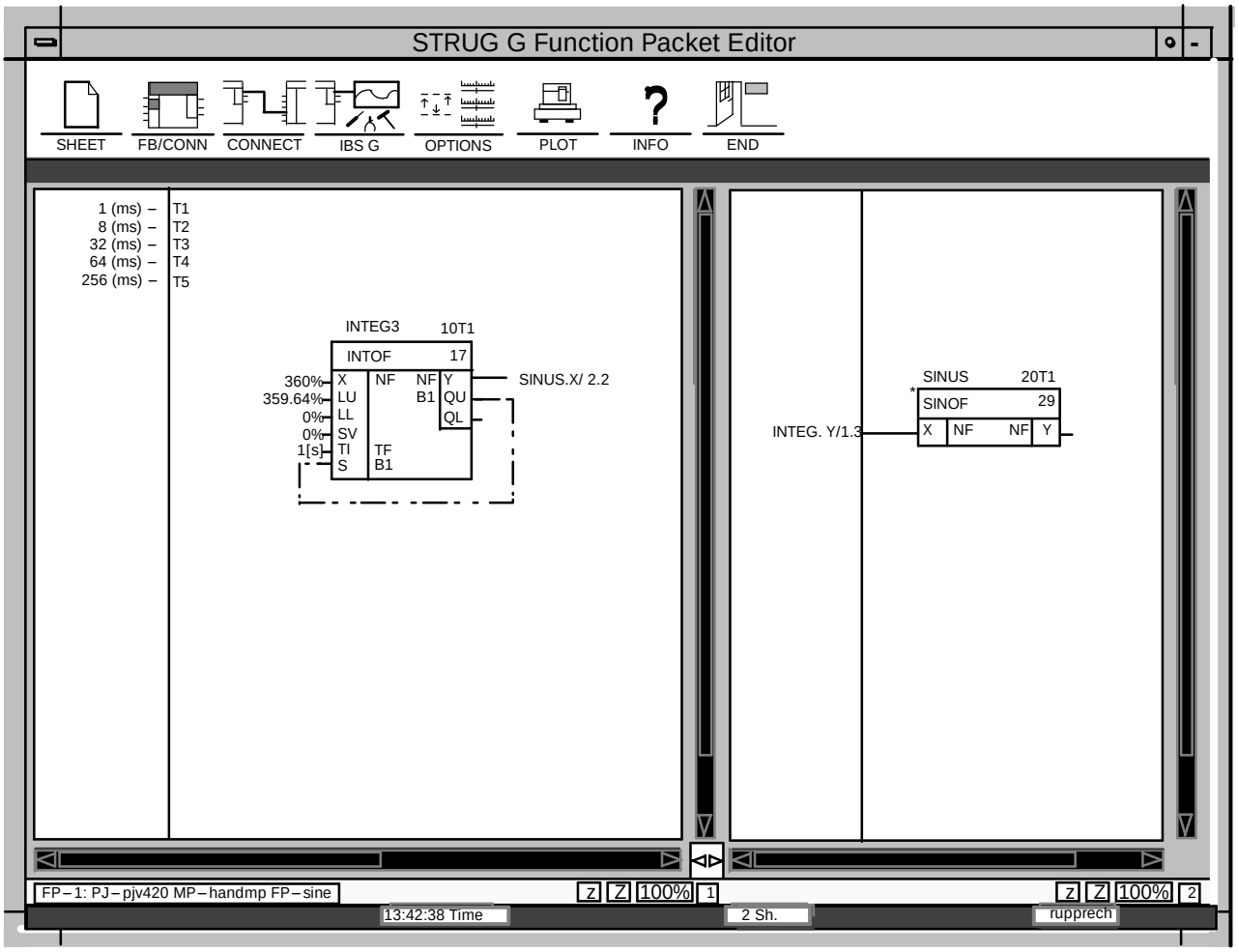


Diagram 1/4: The Screen Layout of the Function Packet Editor in IBS G Mode

1.2.1 Static Menu

The additional menu item ‘IBS G ’’is located between the menu items ‘CONNECT ’’and ‘OPTIONS ’’.

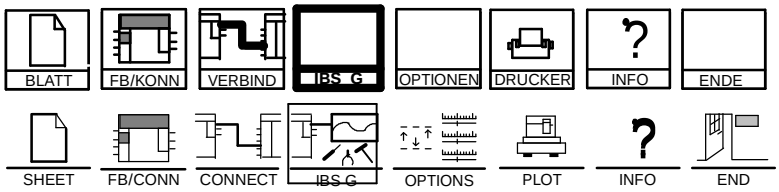
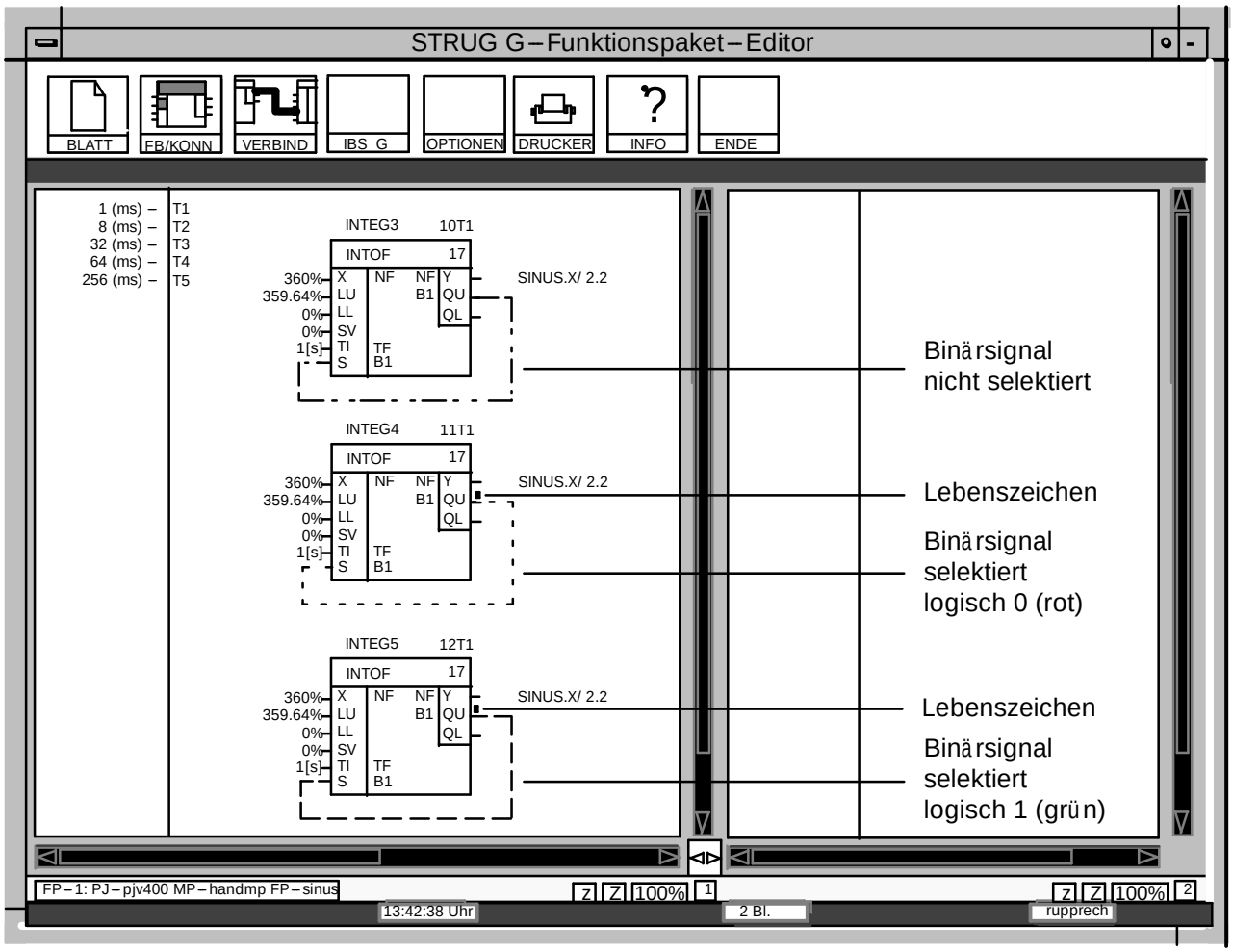


Diagram 1/5: Static Menu

1.2.2 Modifications in the Display of the Screen Sub-Divisions



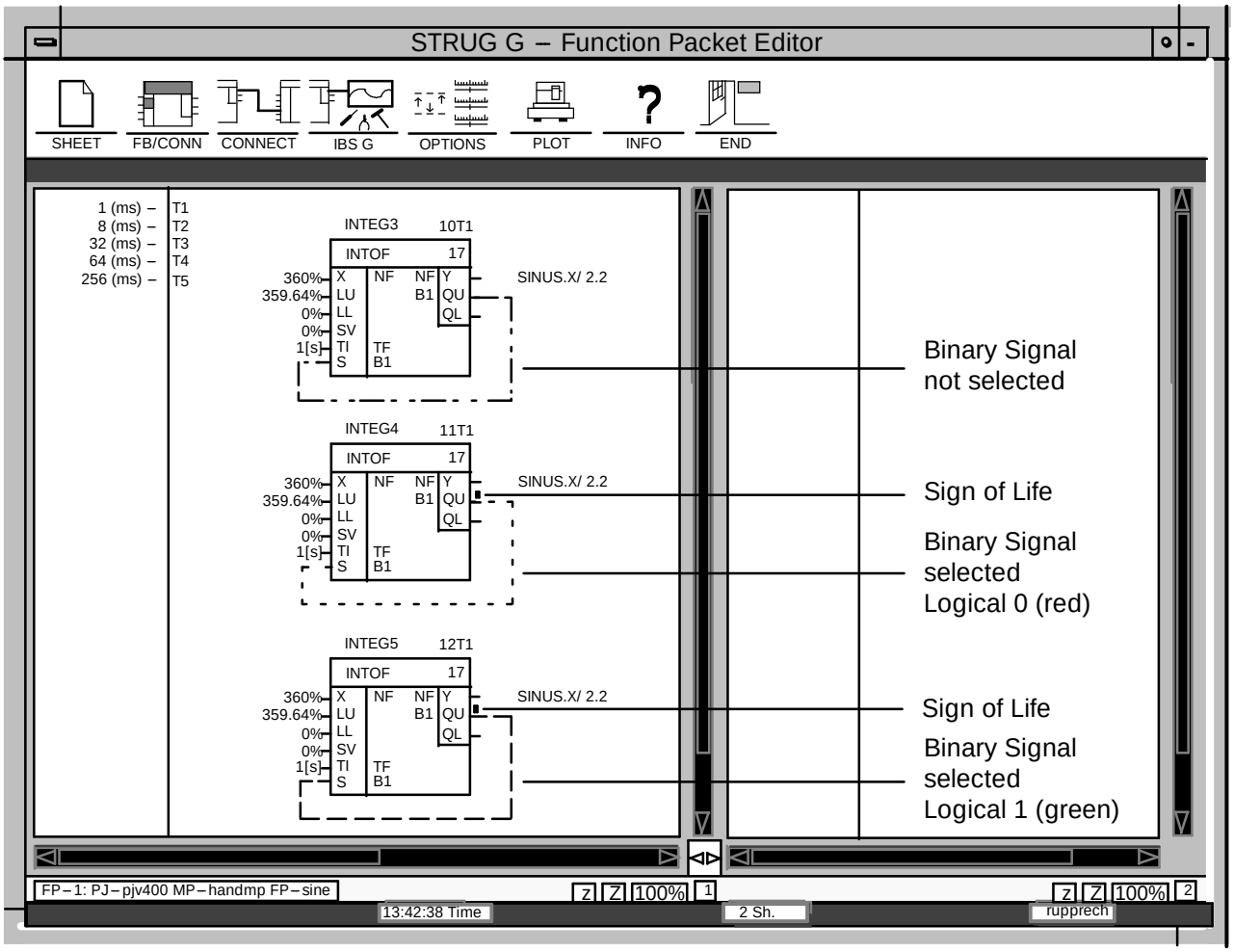


Diagram 1/6: Representation of Binary Signals in IBS G Mode of the Function Packet Editor

The screen sub-division in the IBS G mode of the function packet editor is, in principle, displayed exactly as with the configuring or documenting modes. Binary line routing is an exception.

To improve the clarity of the screen layouts, **binary line routing** in non selected status are drawn with 'Dash Dot '' lines. In the selected mode, the logical status of the binary signal is displayed in the polyline itself (Diagram 1/6).

Additional display elements (value display windows) will still become visible on the function drawing upon request. These value display windows will overlap certain sections of the configuring information.

The value display windows will not be utilized in the plotting or the zooming functions; the binary polylines will appear in the magnifying glass or the plot printout as continuous lines.

1.3 Dialog Elements

1.3.1 Menu Selection

All important basic functions can be selected via the IBS G menu. Therefore activate the symbol ‘IBS G ’’in the menu bar by a mouse click with the left-hand mouse key (will be designated **click** in the following) and invoke the desired IBS G function by selecting the corresponding menu item.

1.3.2 Dialog Screens

The screens available in the function packet editor are divided into **modal** and **non modal** dialogs.

- Modal Dialogs:
  These dialogs are popped onto screen and **must** be exited before the function packet editor will accept inputs. Modal dialogs are utilized for fault or remark windows as well as for prompting.
- Non-Modal Dialogs:
  These dialogs represent the main windows of the function packet editor. The window manager can therefore display either the function packet editor or one of the opened non-modal dialogs in the foreground.
  The size of the function packet editor main window will therefore also sometimes cause a non-modal dialog to be completely covered. When this occurs, the dialog can be returned to the foreground display by repeating the action that opened the dialog window (i.e. activate menu item or button).

1.3.2.1 Variable Screen Sizes

Dialog screens possessing contents of variable size (i.e. list selections or tables) can have the size of the screen modified by the normal Motif ’mouse actions (dragging the edge of the window).

1.3.2.2 Closing Using Alt+F4 or System Menu

All screens can be exited both via one or several buttons (‘OK ’, ‘Cancel ’’etc.) as well as via the Motif ’window manager. The key combinations ‘Alt+F4 ’’or the entry of ‘Exit ’’in the system menu (in the left-hand upper corner of most windows) are therefore active.

1.3.2.3 Keyboard Dialog

All screens can be operated with the mouse or with the keyboard. Toggling from one dialog element to the next is possible using the TAB key. An exception to this are the tables in which a connector input line exists. The TAB key causes a switch to the next input panel in this case.

If the screen element possesses an input focus (indicated by a dotted frame), then the element can receive dialog via the Motif ’normal keyboard entries.

1.4 The Basic Function ‘IBS G ”of the Editor

1.4.1 IBS G: On-Line Functions

1.4.1.1 IBS G -- Start On-Line



Diagram 1/7: Start On-Line

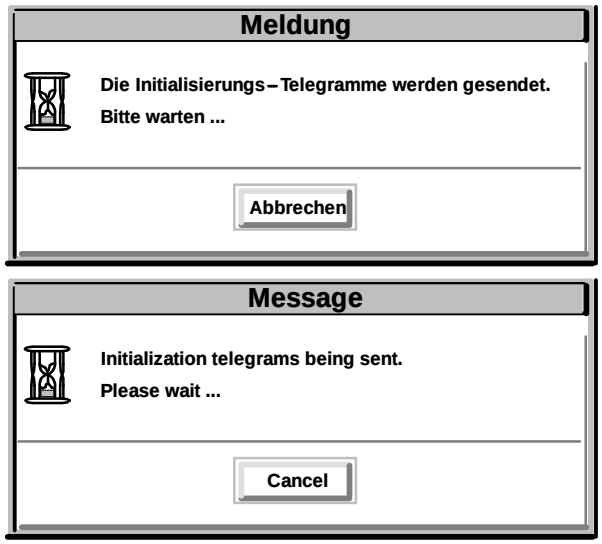


Diagram 1/8: Working Window during the Initialization

A (non modal, review section 1.3.2) dialog window (Diagram 1/9) is popped onto screen after completion of the initialization. This screen indicates to which processors a connection has been created.

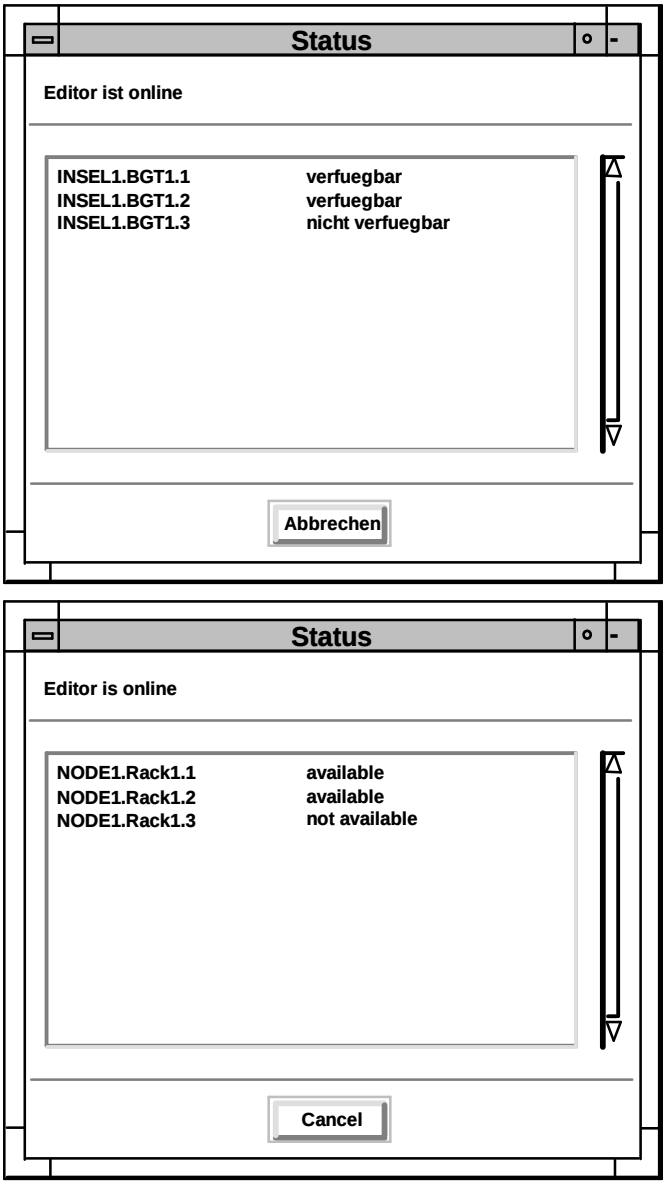


Diagram 1/9: Status Indication after Invoking IBS G – Start On-Line '

If the menu item 'Start On-Line "is invoked while a connection to the HWA exists, then this invocation will have no effect on this connection. It will just cause the dialog window (Diagram 1/9) to be popped onto screen.

If a communication fault occurs to one of the two additional on-line processes during the function packet editor session, then a window is popped onto screen from either Diagram 1/10 or Diagram 1/11 and the function packet editor will no longer be in the on-line mode.

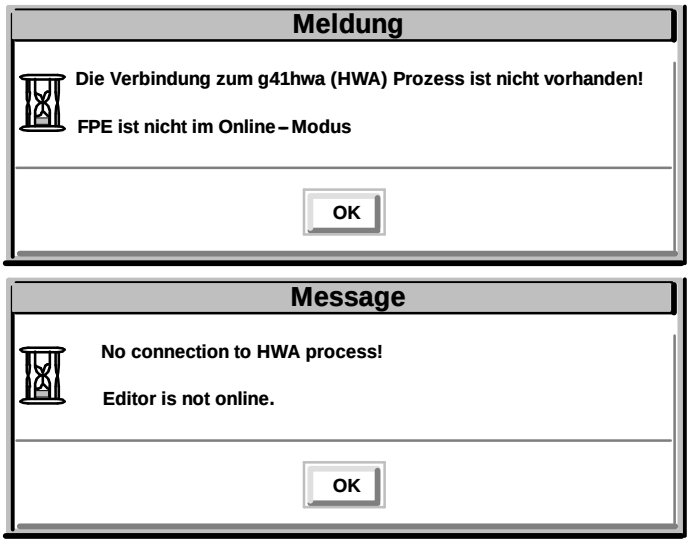


Diagram 1/10: Communication Fault to the 'HWA" Process

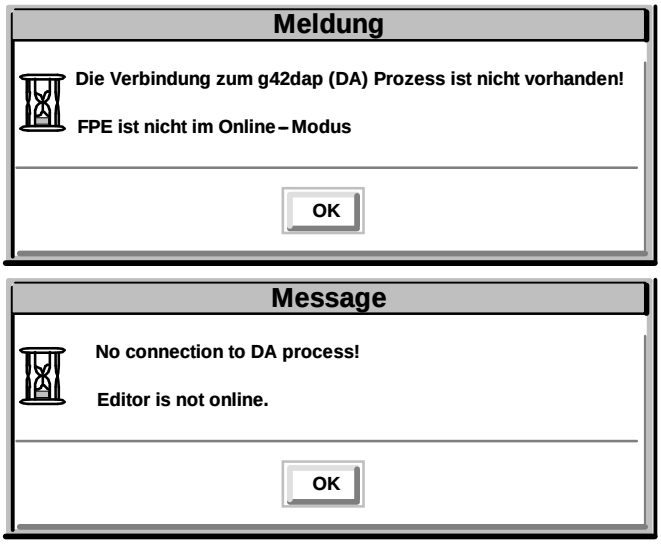


Diagram 1/11: Communication Fault to the 'DA" Process

1.4.1.2 IBS G -- Stop On-Line

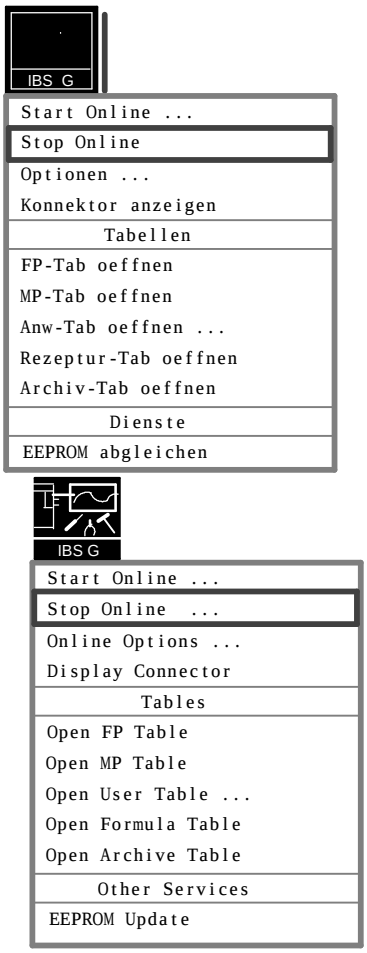


Diagram 1/12: Stop On-Line

This menu item **disconnects** a previously existing **SIMADYN D connection**. All on-line actions are stopped, the current display at the disconnect time point is retained. A subsequent start on-line action returns to the same status as present before executing of the stop on-line action.

Exiting the function packet editor will automatically terminate the connection to SIMADYN D.

This function has no effect if the function packet editor has no (further) connection to the HWA.

1.4.1.3 IBS G -- Options

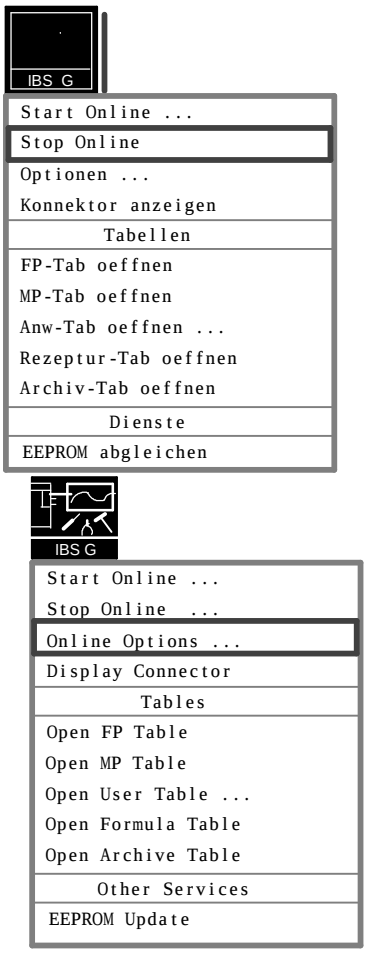


Diagram 1/13: Option

Selection of this menu item causes the option set-up screen to be invoked.

This dialog screen permits the following settings to be made:

- Permanent Modifications:
  If this switch is set, then all on-line modifications are **permanent**, otherwise they are **temporary**.
  Remark: This setting is only valid for those cases in which it is not possible in other dialog screens to select between temporary and permanent.
  Remark: Permanent modifications will be retained even after a reset or powering down of the rack. They are inserted into an EEPROM sub-module.

The options dialog is acknowledged with ‘OK ”, whereby the modifications will then become active. The ‘Cancel ’’ button exits the dialog without further action.

The specifications in the options dialog are not saved for functions invoked after the function packet editor session.

1.4.1.4 IBS G -- Display Connector



Diagram 1/14: Display Connector

This menu item will switch the function packet editor into the basic function ‘Display Connector ’.

This function permits connectors on the function drawing to be selected for display.

The following display appears in the information area:

- Editor Function: Display Values,
- Next Action: Select Signal / Conn.,
- Mouse Keys Function: <Select><Static Menu> <Apply>.

Various actions can be started in the function drawing with the aid of the mouse and are described in more detail in section 1.5.

This function can only be implemented when a connection to SIMADYN D exists. The window shown in Diagram 1/15 is popped onto screen, if this is not the case and the function is not implemented.

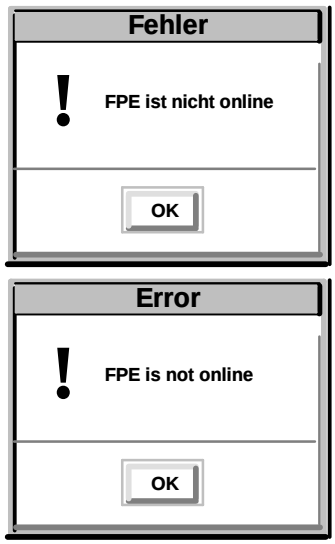


Diagram 1/15: Remark Window for Missing On-Line Connection

1.4.1.5 IBS G -- Open FP Table

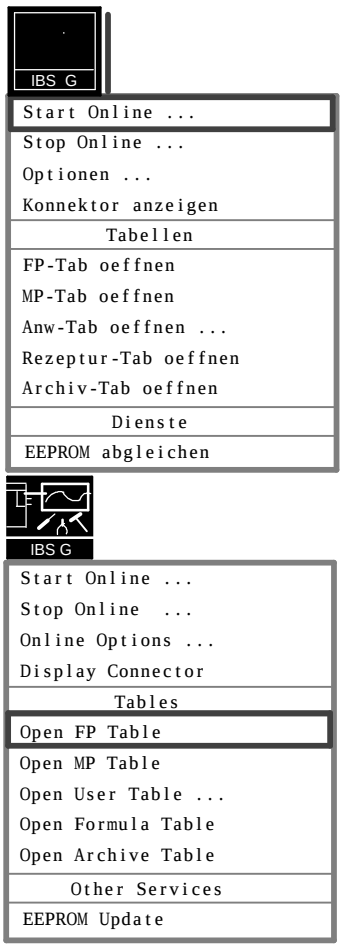


Diagram 1/16: Open FP Table

The **connector table** for the current **function packet of the left-hand screen sub-division** is loaded and displayed with this menu item. If the function packet table is already loaded, then it is brought to the foreground.

The file for the function packet table is located in the directory ‘lnk_onl ’’ and has the file name ‘mpname.fpname.fsl ’’(i.e. ‘mp1.logik.fsl ’).

The processing of connector tables is described in detail in section 1.7.

This function can only be implemented when a SIMADYN D connection exists. The window shown in Diagram 1/15 is popped onto screen, if this is not the case and the function is not implemented.

1.4.1.6 IBS G -- Open MP Table



Diagram 1/17: Open MP Table

This menu item loads and displays the **connector table** for the current **master program of the left-hand screen sub-division**. If the master program table is already loaded, then it will be brought to the foreground.

The master program table file is located in the directory ‘lnk_onl ’’and has the file name ‘mpname.msl ’’ (i.e. ‘mp1.msl ’).

Processing connector tables is described in detail in section 1.7.

This function can only be implemented when a connection to SIMADYN D exists. If this is not the case, then the screen shown in Diagram 1/15 is popped onto screen and the function is not executed.

1.4.1.7 IBS G -- Open User Table



Diagram 1/18: Open User Table

Select a file via the file selection (Diagram 1/19) to open a **user table**. The user table in this file is loaded and displayed after acknowledging with the ‘OK ’’button.

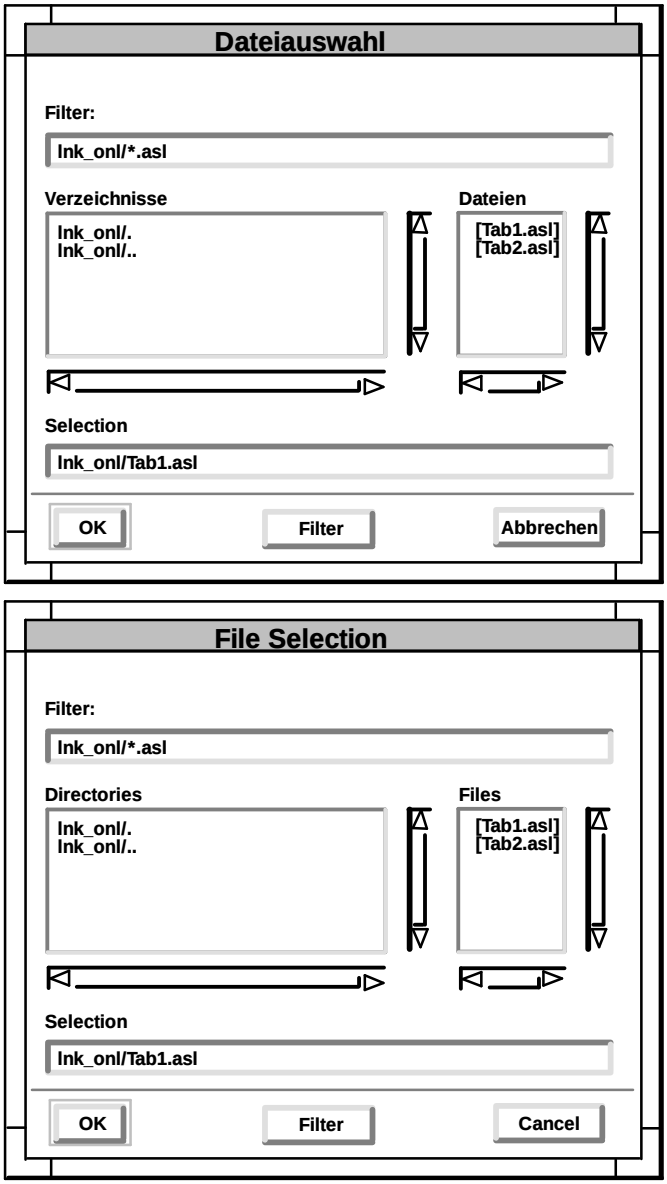


Diagram 1/19: File Selection for a User Table

User tables can be named as required and are utilized, just as FP and MP tables, to collect several connectors from various configuring areas for a single display. Several differing user tables, as opposed to FP and MP tables, can be simultaneously opened at any one time.

The files for the user tables have the extension ‘:asl ’.

The processing of connector tables is described in detail in section 1.7.

This function can only be executed when a connection to SIMADYN D exists. If this is not the case, then the screen from Diagram 1/15 is displayed and the function is not executed.

1.4.1.8 IBS G -- Open Formula Table



Diagram 1/20: Open Formular Table

This menu item loads and displays the **formula table**. If a formula table is already loaded, then it is brought to the foreground.
Only **one** formula table is possible.

The file for the formula table is located in the directory ‘lnk_onl ’’and has the file name ‘bnline.rzt ’’.

The processing of the formula table is described in detail in section 1.10 .

This function can only be implemented when a connection to SIMADYN D exists. If this is not the case, then the screen from Diagram 1/15 is displayed and the function is not executed.

1.4.1.9 IBS G -- Open Archive Table

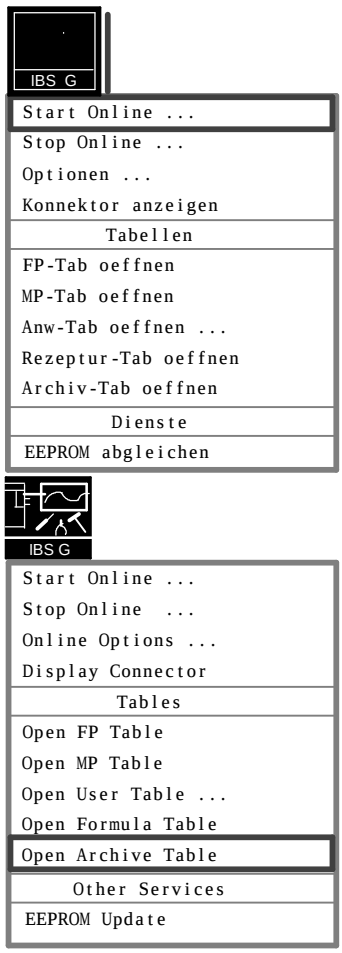


Diagram 1/21: Open Archive Table

The archive table is opened by this menu item. All the archive files (with extension ‘‘:arc ~ ’) located in directory ‘lnk_onl ’’are displayed in this table. If the archive table is already opened, then it is brought to the foreground.

The processing of archives is described in detail in the section 1.11.

1.4.1.10 IBS G -- EEPROM Update

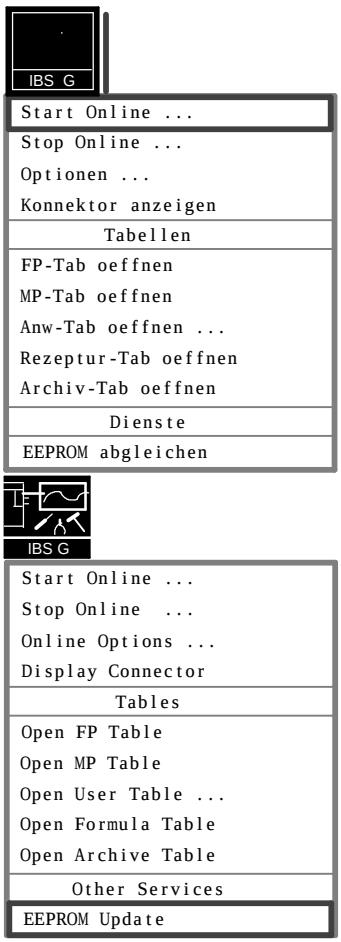


Diagram 1/22: EEPROM Update

Temporary modifications (review page 6–1) are subsequently made permanent with the aid of the 'EEPROM Update".

The information required for this procedure is extracted from the modification log book (see section 1.12). The exact sequencing of the EEPROM update function is described in section 1.12.2.

This function can only be implemented when a connection to SIMADYN D exists. If this is not the case, then the screen shown in Diagram 1/15 is displayed and the function is not executed.

1.4.2 FB/CONN: Block Processing

If the function packet editor is switched to on-line operation, then inserting, copying and deleting of function blocks is also implemented on-line in SIMADYN D. This only requires the utilization of the available dialog actions in the configuring mode such as 'FB/CONN – Call Block', 'FB/CONN – Copy Block', as well as 'FB/CONN – Delete Block'.

Utilizing the function FB/CONN – call block and copy will select, set the parameters and locate a function block in the usual manner. The block is simultaneously activated and run on the selected processor module. This block is therefore 'Idly" running as would a configured spare block.

The inserted function block can now be linked into the running configuration by utilizing the functions from the menu 'CONNECT "(create, delete) or from the menu IBS G (display connector, modify mode value) as described below.

Function blocks can be inserted in both cyclic and interrupt controlled function packets, with the exception of the @ packets (@SIMD, special FP transmit and receive).

Function blocks can only be inserted/copied or deleted in P32 and only temporarily. These modifications are therefore lost when the rack is reset !

1.4.2.1 FB/CONN – Call Block

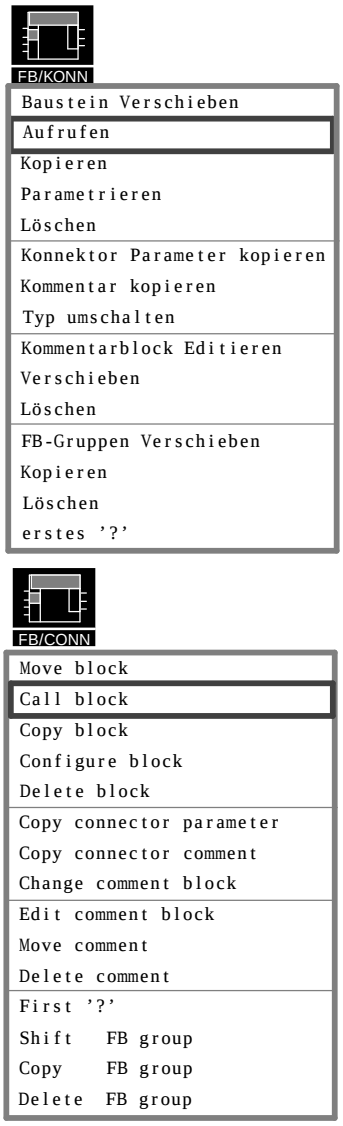


Diagram 1/23: Function Block and Connector Processing

This function permits a new function block to be inserted from the block library of the standard function blocks (FBSLIB) on-line into SIMADYN D. The differences and exceptions to off-line configuring are explained below.

All function block types that are appended with an _o at the type names in the block selection menu can be inserted on-line. Blocks without this on-line coding can only be inserted during configuring. If the block type is entered per keyboard into the FB type panel, then the on-line code does not need to be entered.

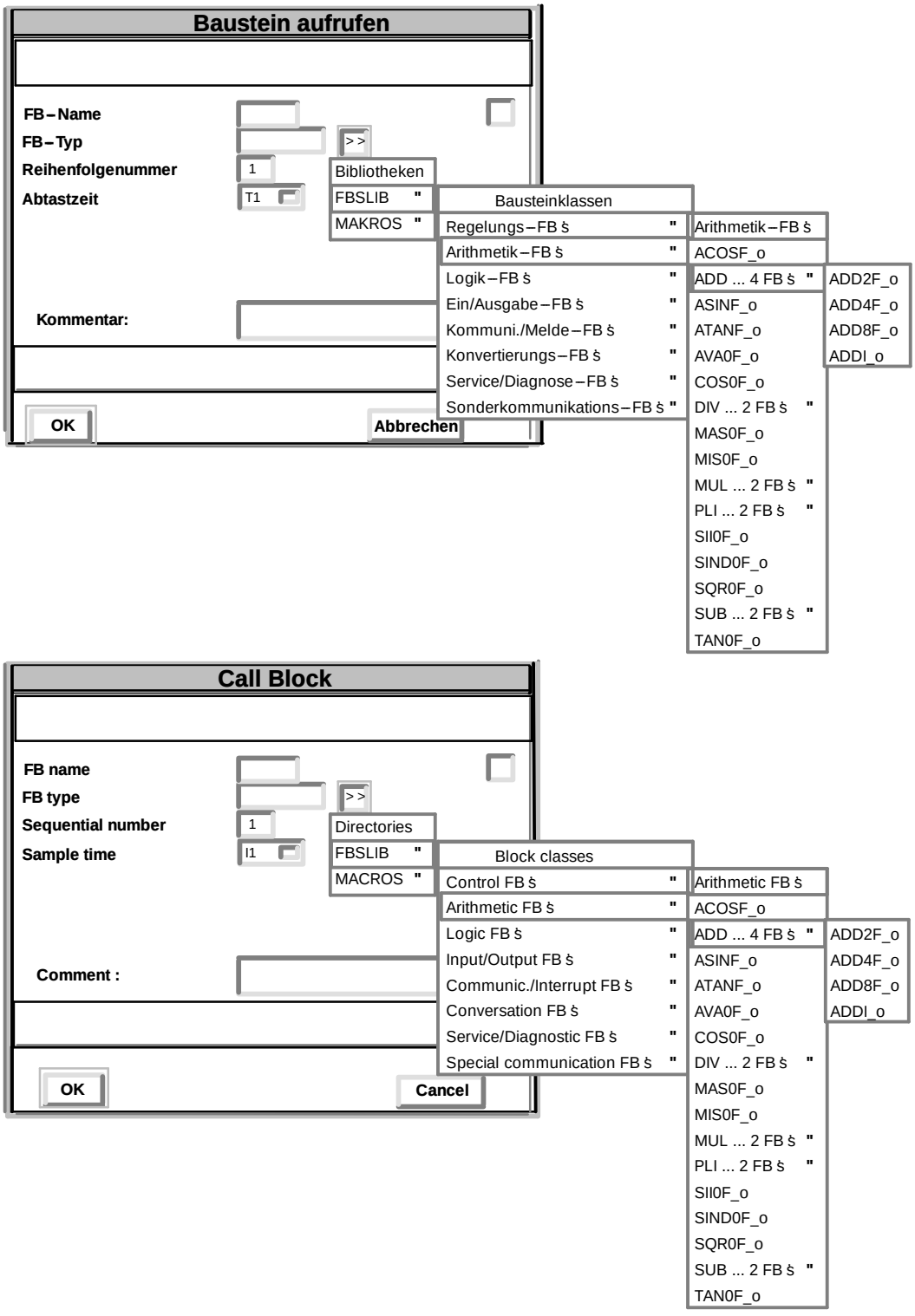


Diagram 1/24: Function Block Header Form with Block Selection Menu
All Displayed Arithmetic Blocks have the On-Line Code and can therefore be Inserted On-Line

Function blocks can only be inserted on-line into previously defined sampling times (entered into the Master Program). The on-line capable sampling times are designated in the selection menu with an appended _o.

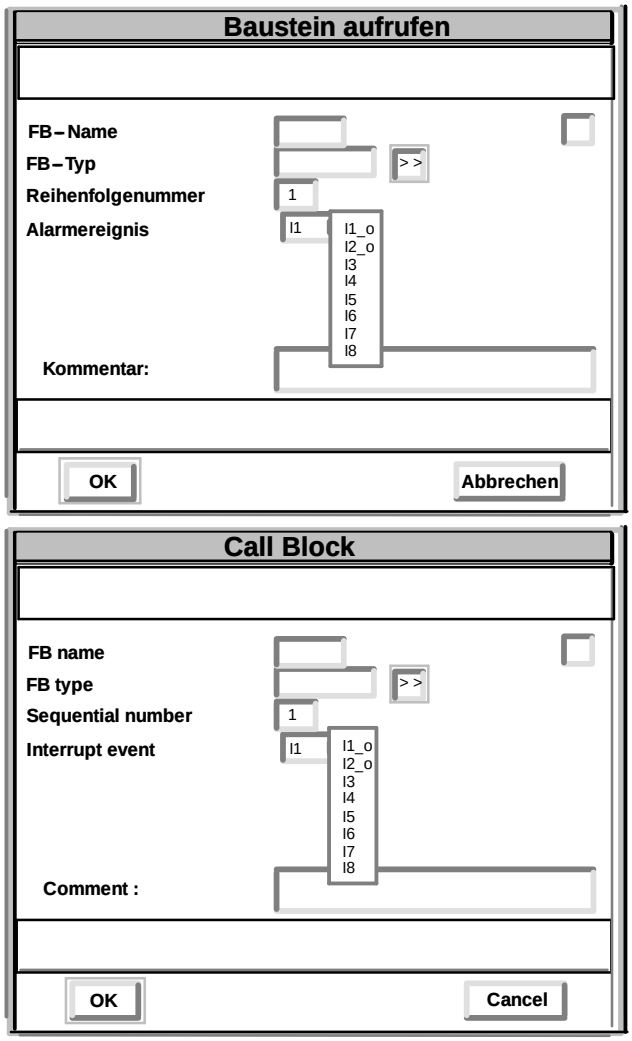


Diagram 1/25: Function Block Header Form with Menu for the Interrupt Event Selection The Interrupt Events I1 and I2 have the On-line Code, only this Code may be Utilized for an On-line inserted Block.

Entries into the comment panel are only stored in the configuring and are not transferred to the processor module.

The correction flag (with ?) is set as a standard feature in the correction flag panel when blocks are inserted on-line. This permits on-line inserted blocks to be found quickly. The ? ' can be deleted from the screen.

After locating/displaying an on-line inserted block, a "?" appears at open input connectors in exactly the same manner as with off-line configured blocks. The default values of these inputs on the SIMADYN D processor are 0 (0%,0H00,0[ms],LOG0). The menu 'IBS G Display Value "permits this value to be displayed, modified and saved in the configuring. All connectors are created without connector attributes (SCAL,MIN,MAX,DIM,...). Modifications at the connectors (modify values or connections) can only be made temporarily, since the block is only temporarily inserted.

An on-line inserted block can be deleted again using the UNDO function.

1.4.2.2 FB/CONN -- Copy Block



Baustein Verschieben
Aufrufen
Kopieren
Parametrieren
Löschen
Konnektor Parameter kopieren
Kommentar kopieren
Typ umschalten
Kommentarblock Editieren
Verschieben
Löschen
FB-Gruppen Verschieben
Kopieren
Löschen
erstes '?'



Move block
Call block
Copy block
Configure block
Delete block
Copy connector parameter
Copy connector comment
Change comment block
Edit comment block
Move comment
Delete comment
First '?'
Shift FB group
Copy FB group
Delete FB group

Diagram 1/26: Copy Block

This function permits a function block already displayed on a sheet to be selected and copied on-line. The same restrictions, as with an on-line insert, also apply to the block types and sampling types/interrupt events which can be copied. All block types that can be inserted on-line can also be copied on-line; i.e. those blocks with an appended _o at the type name in the block selection menu.

Copying a block will also copy all connector values and connector attributes (SCAL,MIN,MAXDIM, @TYP...).

An on-line copied block can also be deleted again via the UNDO function.

1.4.2.3 FB/CONN – Delete

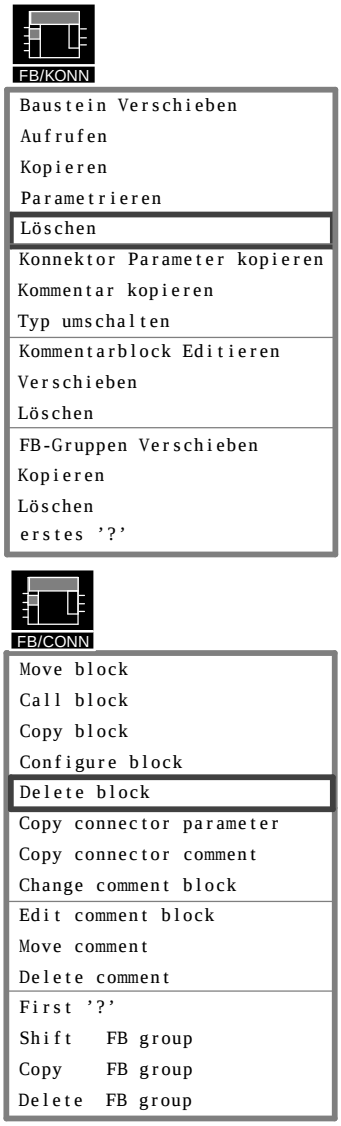


Diagram 1/27: Delete Function Block

This function will remove an existing function block on-line from the function packet. Only those block types can be deleted that can also be inserted on-line.

In order to delete a block on-line, all its connectors must be free of interconnections and no value windows must exist to a connector of this block.

A deleted block can be reactivated again via the UNDO function.

The island INSELE consists of the Master Programs MPA090, MPB100 and MPC110, whose utilities communicate via a peer to peer network and SINEC H1. The allocation of the Master Program to the islands has set the MPA090 with the reference channel number 0, the MPB100 the reference channel number 16 and the MPC110 the reference channel number 32. The US connector of the service FB for the processor 3 of the Master Program MPA090 then receives the character string '#4INSELE_2 " as the address stage 1. The US connector of the download FB for the processor 4 of the Master Program MPC110 receives the character string '#4INSELE_43 ". The CSH11 board of the MPE island must be

informed, via the program NML, that the transport connection name INSELE_2 and INSELE_43 must communicate with the same transport connection names of the STRUC G processor.

1.4.3 CONNECT: Connection Processing

If the function packet editor is currently in on-line operation, then modifications to the connections configuring are updated on-line in SIMADYN D. The dialog actions ‚CONNECT – Create' as well as ‚CONNECT – Delete', are available in the configuring mode and can be utilized for this purpose.

If the connection modifications are to be made permanent, then the corresponding setting can be implemented in the options dialog (see 1.4.1.3) .

1.4.3.1 CONNECT – Draw



Diagram 1/28: Create Connection

When connections are created in on-line operation, then these are reported to SIMADYN D and inserted into the running configuration. The connection then becomes active when acknowledged with the left-hand or right-hand mouse key.

Remark: Only connections **between function blocks** can be newly created on-line. Connections to edge strips ($ signals, virtual connections, ...) cannot be modified on-line.

1.4.3.2 CONNECT – Delete

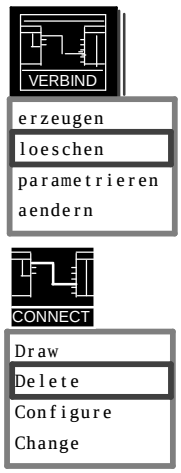


Diagram 1/29: Delete Connection

When connections are deleted, then these are reported to SIMADYN D and removed from the running configuration. The current value of an input at the timepoint of this removal is retained and possibly displayed at any subsequent actions. This value can be subsequently modified via the value modification (see section 1.8). A connection separation becomes active when acknowledged with the left-hand or right-hand mouse key.

Remark: Only connections **between function blocks** can be deleted on-line. Connections to edge strips ($ signals, virtual connection, ...) cannot be modified on-line.

1.4.4 OPTIONS: FP Re–Number, Routing, Forced Routing, Grid, Undo, Redo

1.4.4.1 OPTIONS: Undo

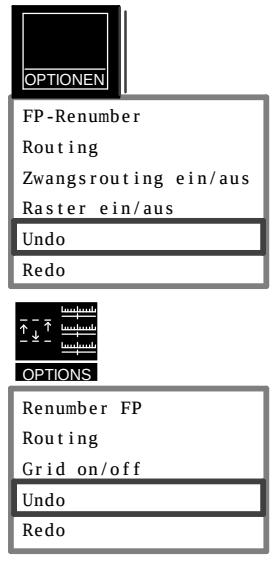


Diagram 1/30: Undo

The function OPTIONS – Undo ' permits the function packet editor in on-line operation to undo the last implemented configuring modification in SIMADYN D. A further undo following the previous undo will redo the first undo !!

The undo function has two basic user applications:

- Undoing Accidental Modifications:
  i.e. value changes at an incorrect connector, values changes with an incorrect value, separating the incorrect connection etc.
- Intentional Toggling between two SIMADYN D States:
  This can be applied to both values (i.e. toggling between two random values at a connector) and also for connections (i.e. temporary insertion of a connection with subsequent separation).

Remark: The undo function can also be executed via the hot key combination ‘Ctrl + u ”.

1.5 Processing a Function Drawing On-Line

If the function packet editor is currently implementing the basic function ‘Display Connector ’, then a series of dialog actions can be carried out on the function drawing with the mouse.

These dialog actions each lead to the on-line display of values from SIMADYN D on the function drawing.

1.5.1 Connector Selection

Connectors can be selected and displayed using different methods:

- Click on a **Connector**:
  A new **value display window** (Diagram 1/39) is opened next to the selected connector and the value of the connector displayed on-line.
- Click on the **Line Routing of an Analog Signal**:
  A new **value display window** (Diagram 1/39) is opened at the clicked on position and the connector value displayed on-line.

1.5.2 Line Routing – Binary Selection

- Click on the **line routing of a binary signal**:
  A binary signal is selected or deselected for the on-line display.

  Binary signals are displayed in the function drawing on-line dependent upon their status (Diagram 1/6):

- Not Selected: Black
- Selected, Logical 0: Red
- Selected, Logical 1: Green

You can recognize the states ‘Not Selected ’, ‘0 ”and ‘1 ”. Non selected binary signals are dash dot lines, ‘0 ’’ signals are red dotted lines and ‘1 ’’signals are green dashed lines.

A small flashing square is located at the end of the line routing and is known as the so called **life sign**. The display of the square is inverted every time a value arrives from SIMADYN D in order to detect activity even for a series of constant values.
If the life sign of a displayed signal **does not flash**, then no values have been received for this connector from SIMADYN D.

1.5.3 Define Connector Addresses

Any number of connectors can be displayed on-line on a function drawing in the form of value display windows. In order to display a connector from a non displayed sheet of the current function packet or another function packet, a **connector selection screen** must be opened:

- Click on an Empty Area with the Right-Hand Mouse Key:
  The connector selection screen (Diagram 1/31) appears, in which a connector address can be entered.

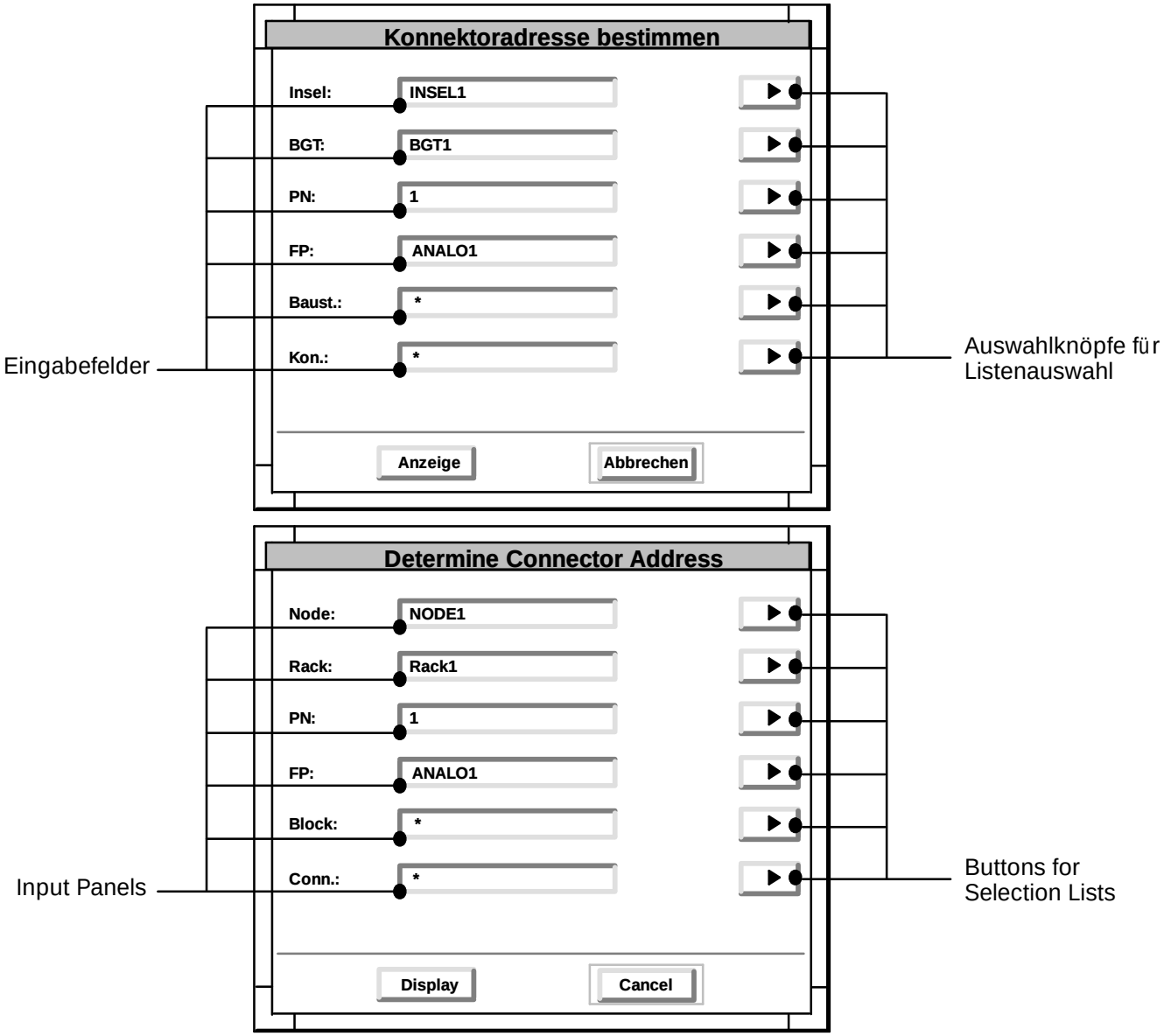


Diagram 1/31: Entering a Connector Address

The individual components of the connector address can be entered into the input panels either manually or via list selection. Pressing the button ‘Display ’will cause an attempt to open a new value display window that will service the desired connector.
An error message will appear with the possibility of repeating the entries if an error has been made (Diagram 1/32). If the registration of the connector in SIMADYN D finally ends in failure (by terminating the error dialog with ‘Cancel ’), then the value display window can once again be closed, since it did not receive any on-line data from SIMADYN D.

No value display window is generated if the selection screen is exited with ‘Cancel ’.

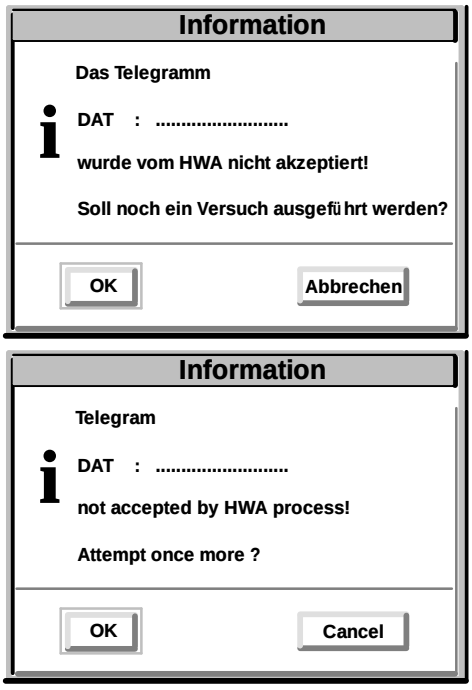


Diagram 1/32: Error Message for Unknown Connector

1.5.4 List Selection

Each component of the connector address has the possibility of selection via a list selection from the total amount of actual objects available.

Activating the button ‘> ”causes a selection list to appear, as shown in Diagram 1/33 to Diagram 1/38.

The connector address can be sequentially set-up **from above to below**. The list selection for the node contains all the network nodes defined in the system. The list system of the rack contains all the racks defined within the super-imposed network node. This is valid in the same manner for all available list selections, such that all the connector designators in the last list selection of the specified block are available.

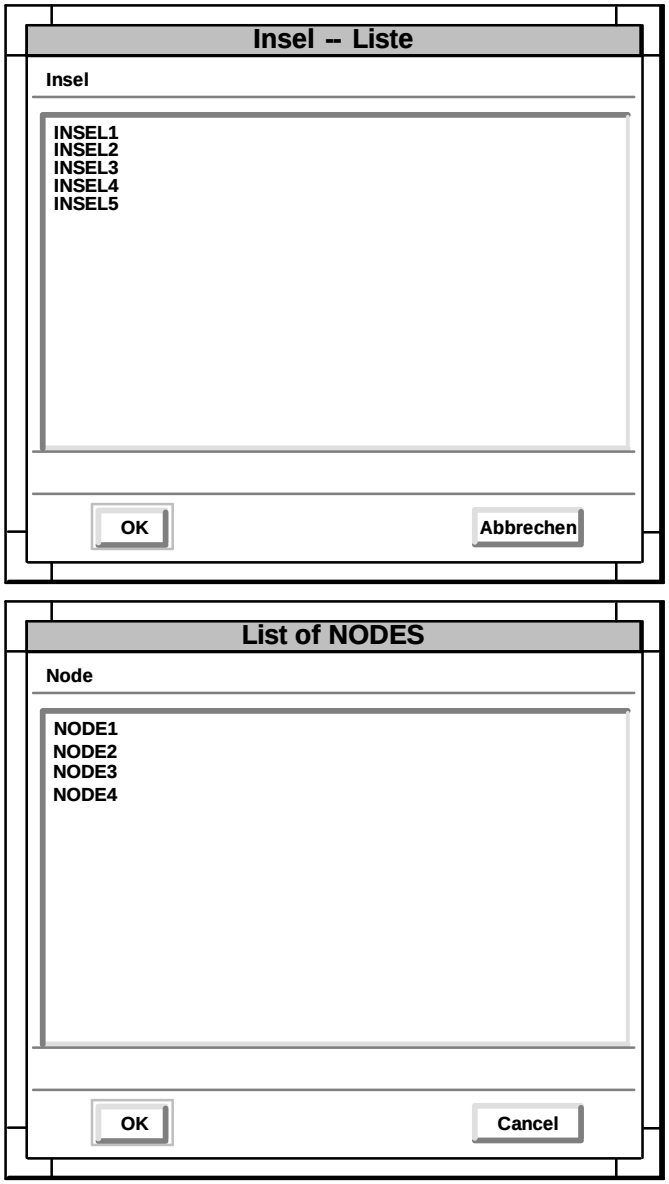


Diagram 1/33: Network Node List

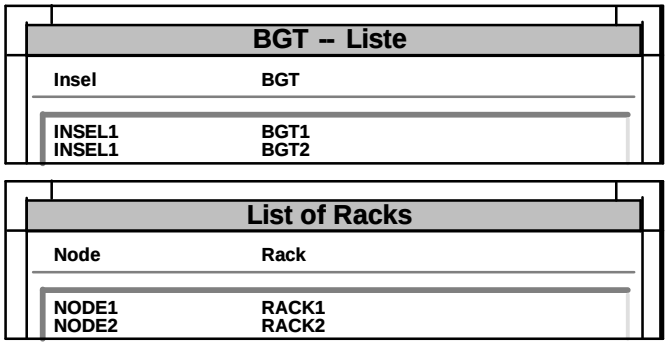
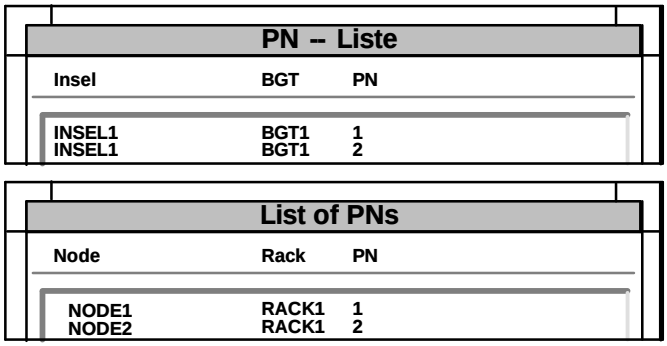


Diagram 1/34: Rack List (Section)



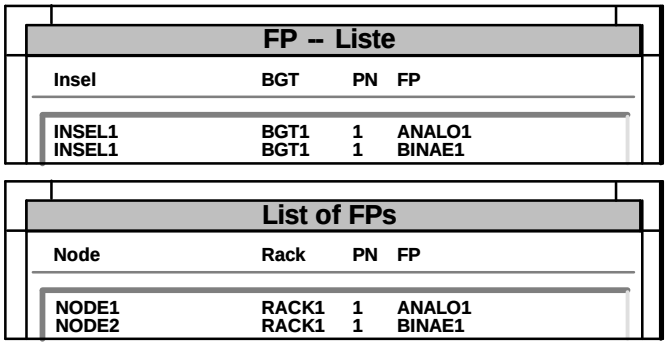
PN -- Liste

Insel	BGT	PN
INSEL1	BGT1	1
INSEL1	BGT1	2

List of PNs

Node	Rack	PN
NODE1	RACK1	1
NODE2	RACK1	2

Diagram 1/35: Processor List (Section)



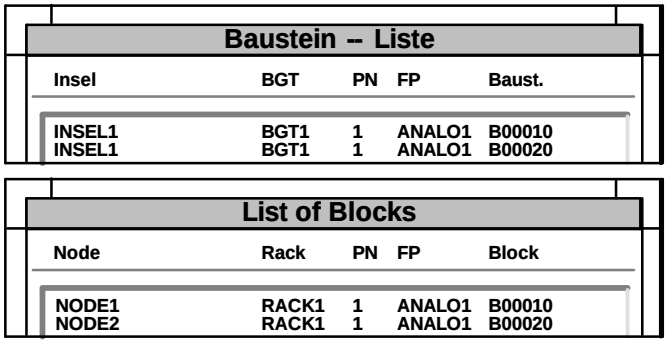
FP -- Liste

Insel	BGT	PN	FP
INSEL1	BGT1	1	ANALO1
INSEL1	BGT1	1	BINAE1

List of FPs

Node	Rack	PN	FP
NODE1	RACK1	1	ANALO1
NODE2	RACK1	1	BINAE1

Diagram 1/36: Function Packet List (Section)



Baustein -- Liste

Insel	BGT	PN	FP	Baust.
INSEL1	BGT1	1	ANALO1	B00010
INSEL1	BGT1	1	ANALO1	B00020

List of Blocks

Node	Rack	PN	FP	Block
NODE1	RACK1	1	ANALO1	B00010
NODE2	RACK1	1	ANALO1	B00020

Diagram 1/37: Block List (Section)

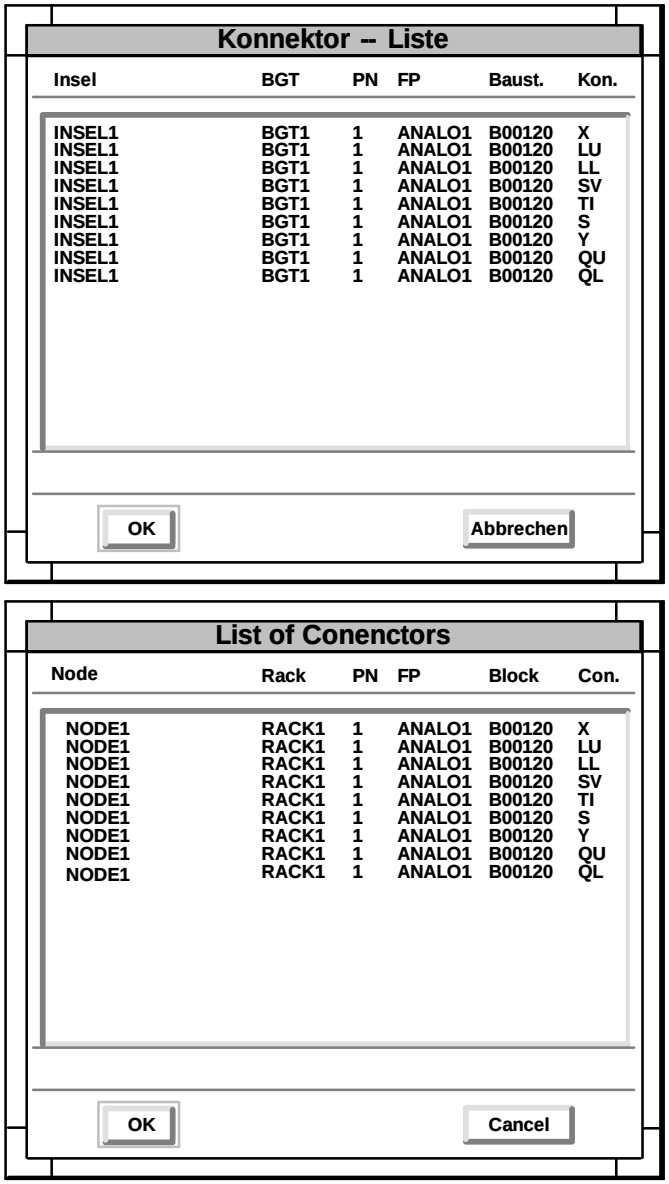


Diagram 1/38: Connector List

1.6 Processing Value Display Windows

The value display window is a flexible output element with which both binary and also analog signals can be displayed. Value display windows are located on the function drawing and assigned to a particular screen sub-division.

The value display windows are closed when a sheet is changed. The value display windows are re-opened if this sheet is reloaded during the same function packet editor session.

1.6.1 Display

Value display windows are displayed as a separate character area, which covers the information it is lying above. Value display windows are ‘Invisible ’’with regard to the magnifying glass function and are also not output when function drawings are plotted.

Each value display window consists of a frame, an optional title bar as well as the display area with life sign flag. The life sign flag is inverted every time a value arrives from SIMADYN D, thereby indicating activity even when a series of constant values is received. If the life sign within a value display window does not flash, then no values are currently been received for this connector from SIMADYN D.

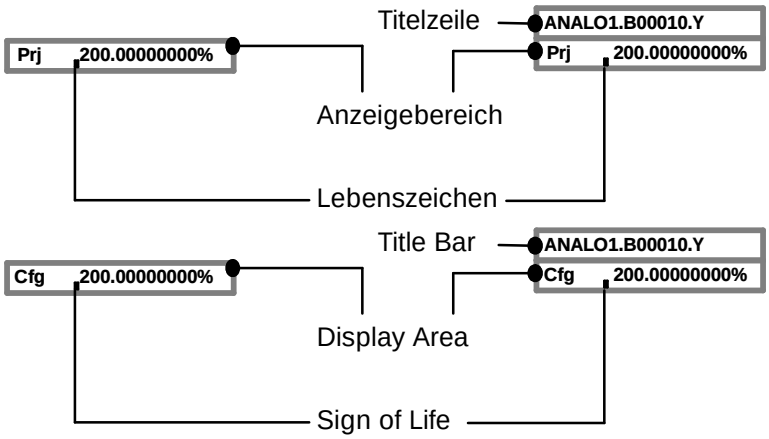


Diagram 1/39: Value Display Window With and Without Title bar

If value display windows are generated for a connector in the function drawing by means of a click, then they will receive a starting position directly next to the connector, pointing outwards and without title bar. Therefore all connectors of a block can be simultaneously displayed in value display windows without ov erlapping (see Diagram 1/40).

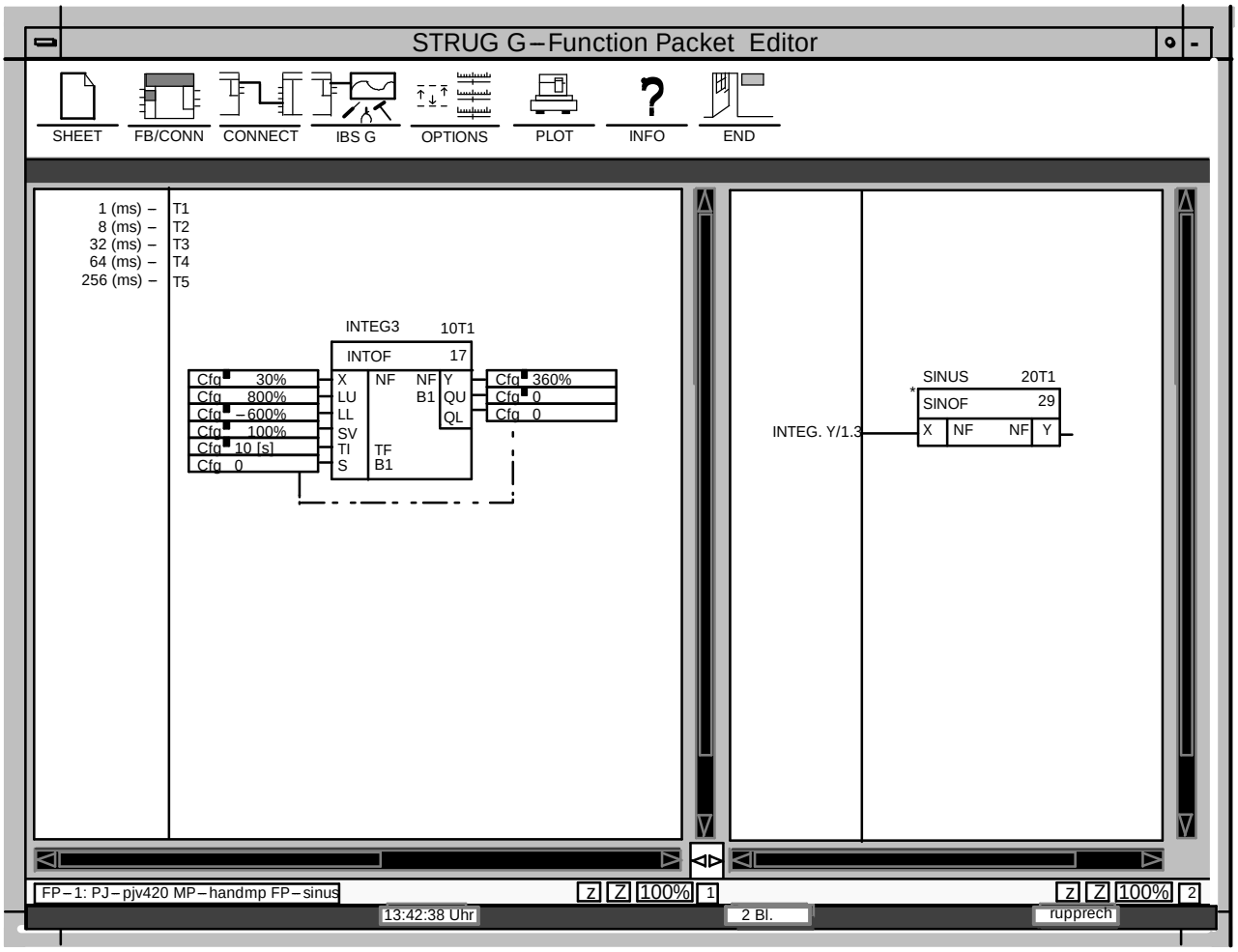


Diagram 1/40: Value Display Window in the Function Drawing

1.6.2 Selection Mechanism

Value display windows are selected using the following method:

- Shift + Click on a Value Display Window:
  The value display window is selected or deselected, dependent upon the previous status. Selected value display windows are displayed in the inverted form.

The basic function ‘IBS G – Display Connector ’’contains additional possibilities for selecting and deselecting value display windows:

- Shift + Click on a Block:
  Additionally select all value display windows that are associated with this block.
- Click on the Empty Area of a Screen Sub-Division:
  Deselects all value display windows located in this screen sub-division.
- Shift + Click on the Empty Area of a Screen Sub-Division:
  Select all value display windows that are located in this screen sub-division.

1.6.3 Shifting Value Display Windows

D Click on the value display window and subsequently move the mouse with held mouse key.
The value display window is shifted to a new position within the same screen sub-division. If a selected value display window is shifted, then all the other selected value display windows are shifted along with it.

1.6.4 Pop-Up Menu

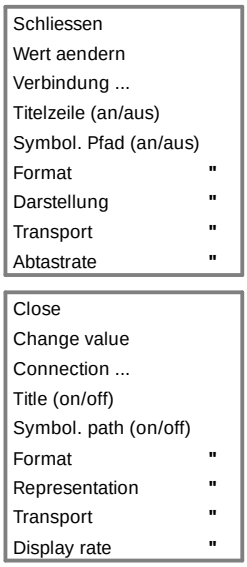


Diagram 1/41: Pop-up Menu of a Value Display Window

A pop-up menu appears on screen by **clicking with the right-hand mouse key** on a value display window. Processing functions can then be activated for the value display window. The processing functions are relevant to either one single value display window or to all selected value display windows, dependent of whether the pop-up menu was invoked for an deselected or a selected value display window.

1.6.4.1 Close

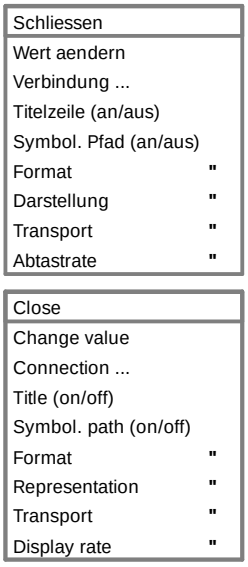


Diagram 1/42: Pop–Up Menu – Close

The clicked on value display window is closed.

If the pop-up menu was invoked on a selected value display window, then this function closes all selected value display windows.

Remark: Close all value display windows of a screen sub-division by selecting all the value display windows of the screen sub-division in the basic function with Shift + Click ’on an empty space of this screen sub-division. All value displayed windows are subsequently removed by ‘Closing ’.

1.6.4.2 Change Value

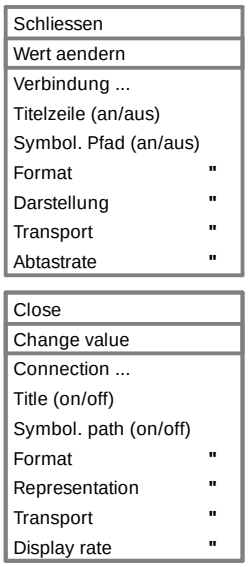


Diagram 1/43: Pop–Up Menu – Change Value

A change value window (see section 1.8, Diagram 1/59) is popped onto screen for the connector of the value display window upon which the pop-up menu was invoked. If the change value window is already opened then it will be loaded with the new connector and brought to the foreground.

1.6.4.3 Connection

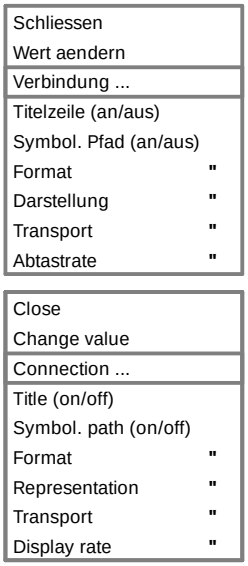


Diagram 1/44: Pop-Up Menu – Connection ...

The connection table (see section 1.9, Diagram 1/64) is popped onto screen for the connector of the value display window upon which the pop-up menu was invoked. The connector of the value display window is entered as the reference connector. If the connection table is already open, then it is loaded with the new reference connector and brought to the foreground.

1.6.4.4 Title Bar

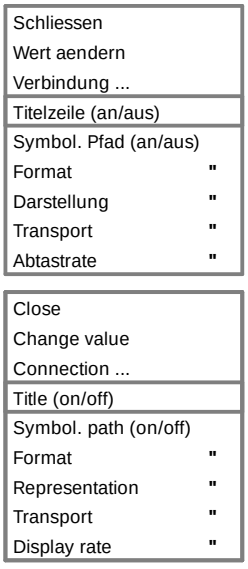


Diagram 1/45: Pop-Up Menu – Title Bar (On / Off)

The clicked on value display window is displayed with or without title bar. The title bar is utilized for the identification of the connector should the value display window be shifted away from its original position or when value display windows originating from other sheets or function packets are displayed.

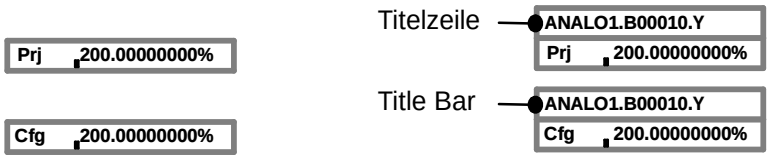


Diagram 1/46: Value Display Window with and without Title Bar

Remark: The associated value display window **toggles** the status of the title bar with this function.

If the pop-up menu is invoked on a selected value display window, then this function is implemented on all selected value display windows.

1.6.4.5 Symbolic Path

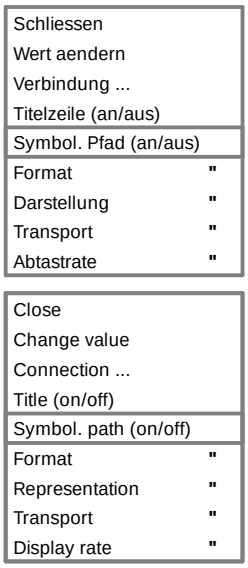


Diagram 1/47: Pop-Up Menu – Symbolic Path (On / Off)

This menu item is utilized to toggle the display in the title bar of the clicked on value display window. An activated symbolic path display shows the designator contained in the configuration file 'Symbolic Path Designator '' instead of the multi component connector address (see section 1.13.1).

Remarks: The associated value display window **toggles** the representation type of the title bar using this function. If the pop-up menu has been invoked on a selected value display window, then this function is implemented on all selected value display windows.

1.6.4.6 Format

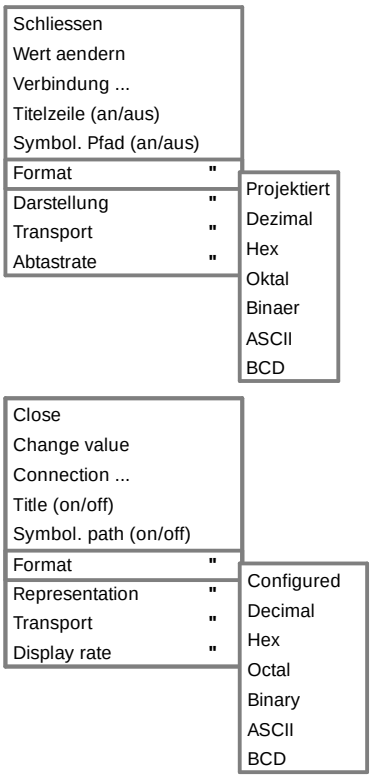


Diagram 1/48: Pop-Up Menu – Format

The menu item ‘Format ”possesses a sub-menu from which a particular output format can be set for each value display window. Activation of this function will switch the format in the value display window and the current value will be output in the new format at the next display cycle.

If the pop-up menu is invoked on a selected value display window, then this function is executed for all selected value display windows.

The following output formats are available:

- Configured Format (Default):
  The output is based upon the connector type and the specifications defined during configuring.
- Decimal:
  The value of the connector is interpreted and output as a signed decimal number.

  Remark: A maximum of the first four bytes of the connector value is utilized.
- Hexa–Decimal:
  The value of the connector is interpreted and output as a hexa number without sign.

  Remark: A maximum of the first four bytes of the connector value are utilized.
- Octal:
  The value of the connector is interpreted and output as an octal number without sign.

  Remark: A maximum of the first four bytes of the connector value is utilized.

- Binary:
  The value of the connector is interpreted and output as a binary number without sign.

  Remark: Due to the many output digits required for binary numbers, this display type should only be utilized for connector types with a maximum length of two bytes.
- BCD:
  The value of the connector is interpreted and output as a BCD number (Binary Coded Decimal). BCD numbers will have two digits of a number coded into binary and displayed as one byte.

  Example:
  The decimal number 385 is represented in its 2 byte BCD form:

3	8 5
0000 0011	1000 0101
Byte 1	Byte 2

  Bit positions that do not correspond to a BCD digit, are output as '?'.
- ASCII:
  The value of the connector is interpreted and output as an ASCII character string.

1.6.4.7 Representation

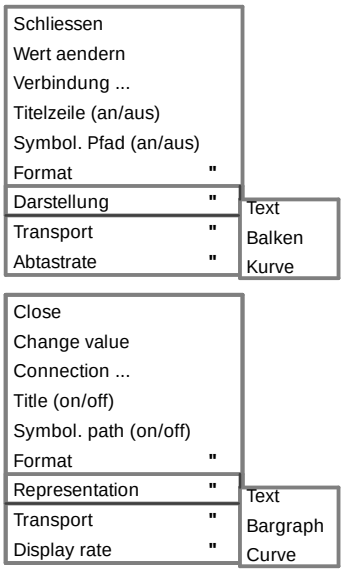


Diagram 1/49: Pop-Up Menu – Representation

The menu item 'Representation ''possesses a sub-menu from which various forms of display can be set for each value display window. Activation of this function causes the representation type of the value display window to be switched and the current value to be output in the new representation form at the next display cycle.
If the pop-up menu is invoked on a selected value display window, then the function is implemented for all selected value display windows.

The following representation types are available:

- Text:
  The value of the connector is output as a character string in the display area of the value display window in the selected format. The output format is indicated in front of the value.

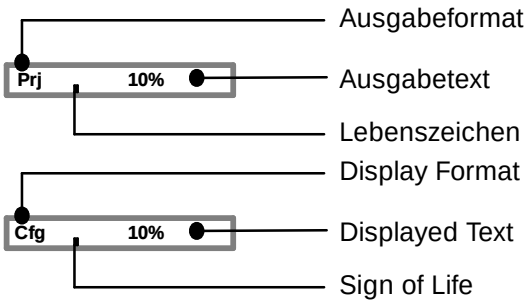


Diagram 1/50: Representation Type 'Text" in a Value Display Window

D Bar:
The value of the connector is drawn in the display area as a black bar, stretching from the zero point to the current value. The previously achieved maximum value is displayed above the display area, the minimum value is displayed below the display area; both displayed in the set output formats. The zero point is marked by a small vertical line in the display area. The left-hand edge of the display area coincides with the minimum value, the right-hand edge with the maximum value.
Exceeding the previous maximum value or falling below the previous minimum value will cause the maximum and minimum value indicators to be correspondingly adapted and the zero point marker to be reset. This procedure is known as **Auto Scaling** and is active whilst the value display window is opened in the bar representation form.

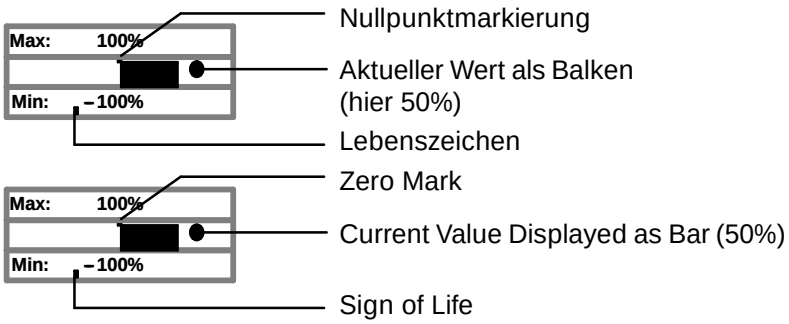


Diagram 1/51: Representation Type 'Bar" in a Value Display Window

D Curve:
As opposed to the representation types 'Text" and 'Bar", the 'Curve''representation type is not a **sampled value representation** but rather an **integral response value representation**.
The display area of the value display window is enlarged such that a polygon curve can be displayed, which shows the current value of the connector as well as a series of past values. The current value will be located on the right-hand edge of the display area. The previously achieved maximum value will be indicated above the actual display area and the minimum value below the display area; both displayed in the set output formats. A current value is drawn with a dot located between the upper edge (maximum value) and the lower edge (minimum value). Exceeding the previous maximum value or falling below the previous minimum value will cause the maximum and minimum value indications to be correspondingly adapted as well as the current value and the past values to be redrawn.

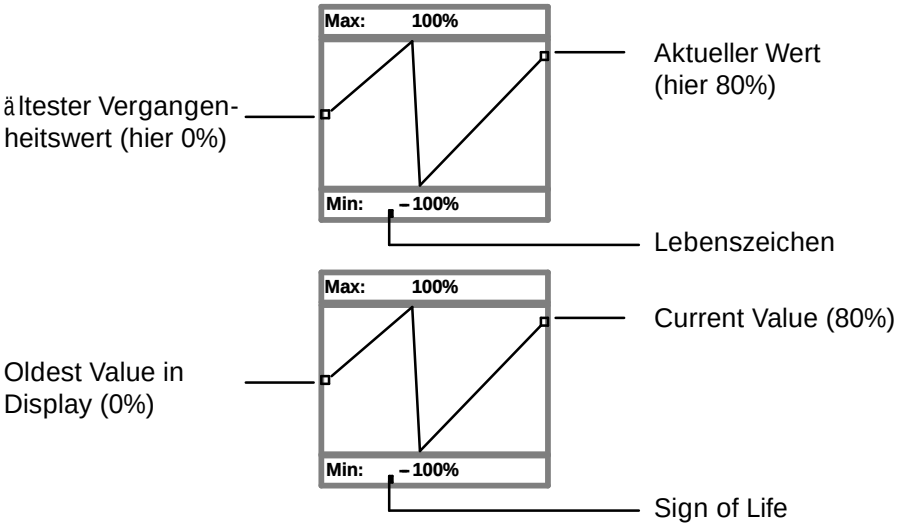


Diagram 1/52: Representation Type 'Curve ''in a Value Display Window

Remarks: Reviewal of the curves drawn in the value display window requires the following to be understood:

- The values are drawn in the speed with which the system is capable of refreshing the screen. Values can be possibly left out of the polygon curve when many value display windows are open or other high system loading occurs, since the next value is then supplied before the previous value has been displayed. This can cause curve deformation.
- Rapidly occurring value sequences can possibly be incorrectly displayed due to the 'Sub-Sampling Rate '' effect. The sub-sampling rate can occur both in SIMADYN D (mainly due to the configured sampling time of the monitor or service block) and in the function packet editor (due to the already mentioned system loading or to a slow display cycle).

1.6.4.8 Transport

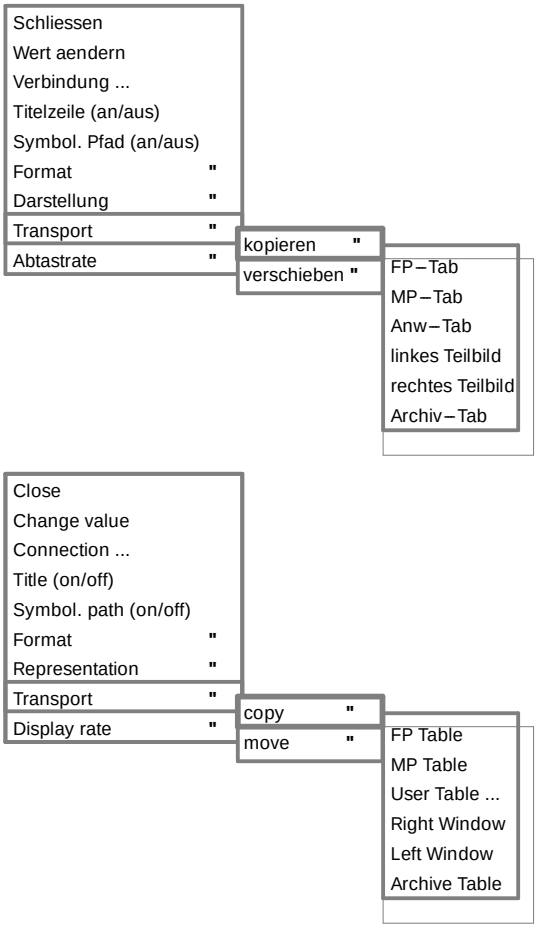


Diagram 1/53: Pop-Up Menu – Transport

Selection of the menu item ‘Transport ’invokes a sub-menu containing the menu items ‘Copy ’and ‘Move ’. These menu items each contain further sub-menus required for the selection of the destination for the value display window copy and move operations.

‘Copy ’causes the corresponding connectors to be copied to the destination and additionally to retain the value display windows.

‘Move ’causes the corresponding connectors to be copied to the destination and the value display windows to be simultaneously closed.

The following destinations are available for the ‘Transport ”function:

- FP Table:
  The corresponding connectors are copied or moved in the FP table in front of the current table line. If the FP table is not opened, then this function has no effect.
- MP Table:
  The corresponding connectors are copied or moved in the MP table in front of the current table line. If the MP table is not opened, then this function has no effect.
- User Table ...:
  Selection of this menu item causes a list of all opened user tables to be popped onto screen (Diagram 1/54), from which a table can be selected. The corresponding connectors are copied or moved into the selected user table. If no user table has been opened, then this function has no effect.

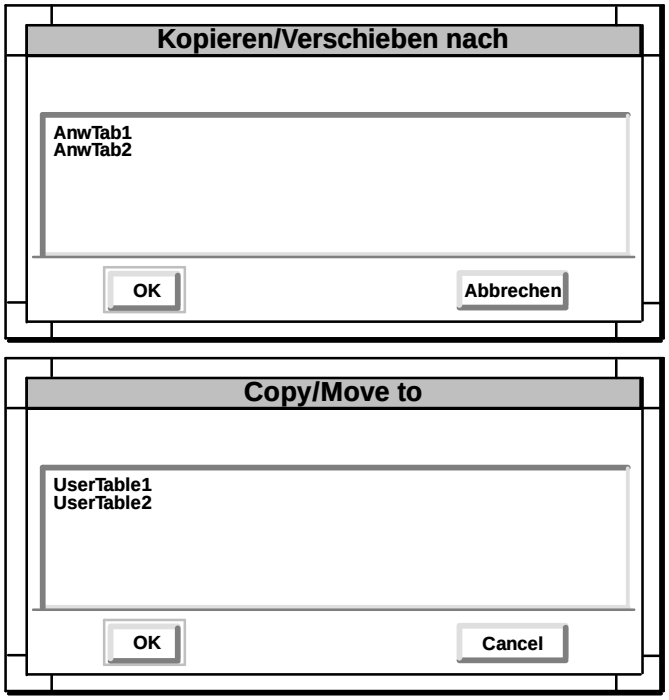


Diagram 1/54: Destination Selection for Transporting in a User Table

- Left-hand Screen Sub-Division:
  The corresponding connectors are copied or moved into the left-hand screen sub-division of the function packet editor, as long as it is loaded with a sheet for processing.
  This menu item is not available in the pop-up menu for value display windows located in the left-hand screen sub-division.
- Right-hand Screen Sub-Division:
  The corresponding connectors are copied or moved into the right-hand screen sub-division of the function packet editor, as long as it is loaded with a sheet for processing.
  This menu item is not available in the pop-up menu for value display windows located in the right-hand screen sub-division.
- Archive Table:
  The corresponding connectors are copied or moved into the current archive of the archive table. If the archive table is not opened, then this menu item has no effect.

1.6.4.9 Display Rate

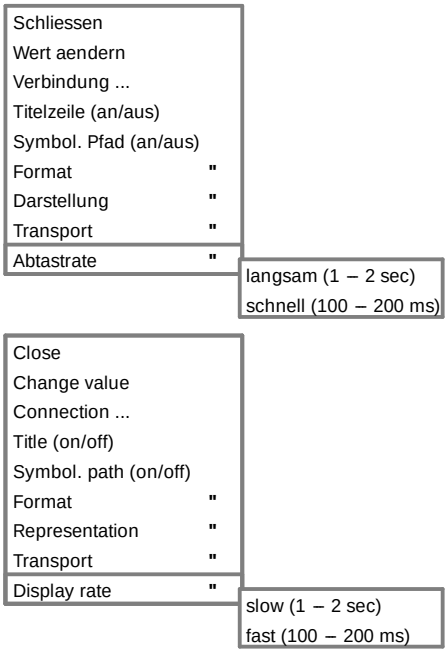


Diagram 1/55: Pop-Up Menu – Display Rate

Connector values can be displayed with two different refresh rates in the function packet editor. The **slow** display rate outputs a new value approximately every 1 – 2 seconds. The **fast** display rate (**default** for a newly opened value display window) will refresh the display at much shorter intervals. Whereby a new cycle is started approximately 100 ms after a completed display cycle (the display of all desired values in the various forms).

Remark: The sampling rate of the function packet editor strongly influences the communication rate between the HWA and the data preparation It is therefore recommended to restrict only a few connectors to the fast display rate. If many connector values on a sheet are displayed in the **fast** refresh rate, then it will possibly take a long period until all value display windows can be displayed with their values after a change of sheet.

1.7 Processing Connector Tables

1.7.1 Table Categories

Three various categories of connector tables exist for the on-line operation of the function packet editor:

- **An MP Table** for each Master Program:
  An MP table automatically possesses the names of the corresponding master program and can be popped onto screen via the menu item ‘IBS G – Open MP Table ”.
  Connectors representing the total inter-relationships within a rack should be displayed with an MP table.

  Remark: A total of one common MP table exists for all function packets in a master program.
- **An FP Table** for each Function Packet:
  An FP table automatically possesses the names of the corresponding function packets which prefixes the associated master program name separated by a ‘: ’. It can be popped onto screen via the menu item ‘IBS G – Open FP Table ”.
  Connectors representing the total inter-relationship within a function packet should be displayed in an FP table.

  Remark: A total of one FP table exists for each function packet.
- **Any Number of User Tables**:
  An unrestricted name can be defined by the user for a user table.
  The menu item ‘IBS G – Open User Table ... ”permits a user table to be selected and displayed from those available.

  Remark: User tables can be accessed in the same manner from all function packets. A practical method of defining names for user tables is therefore essential.

1.7.2 Representation

FP, MP and user tables are differentiated only in the representation of the title bar. The following is therefore the description of an example based upon a function packet table.

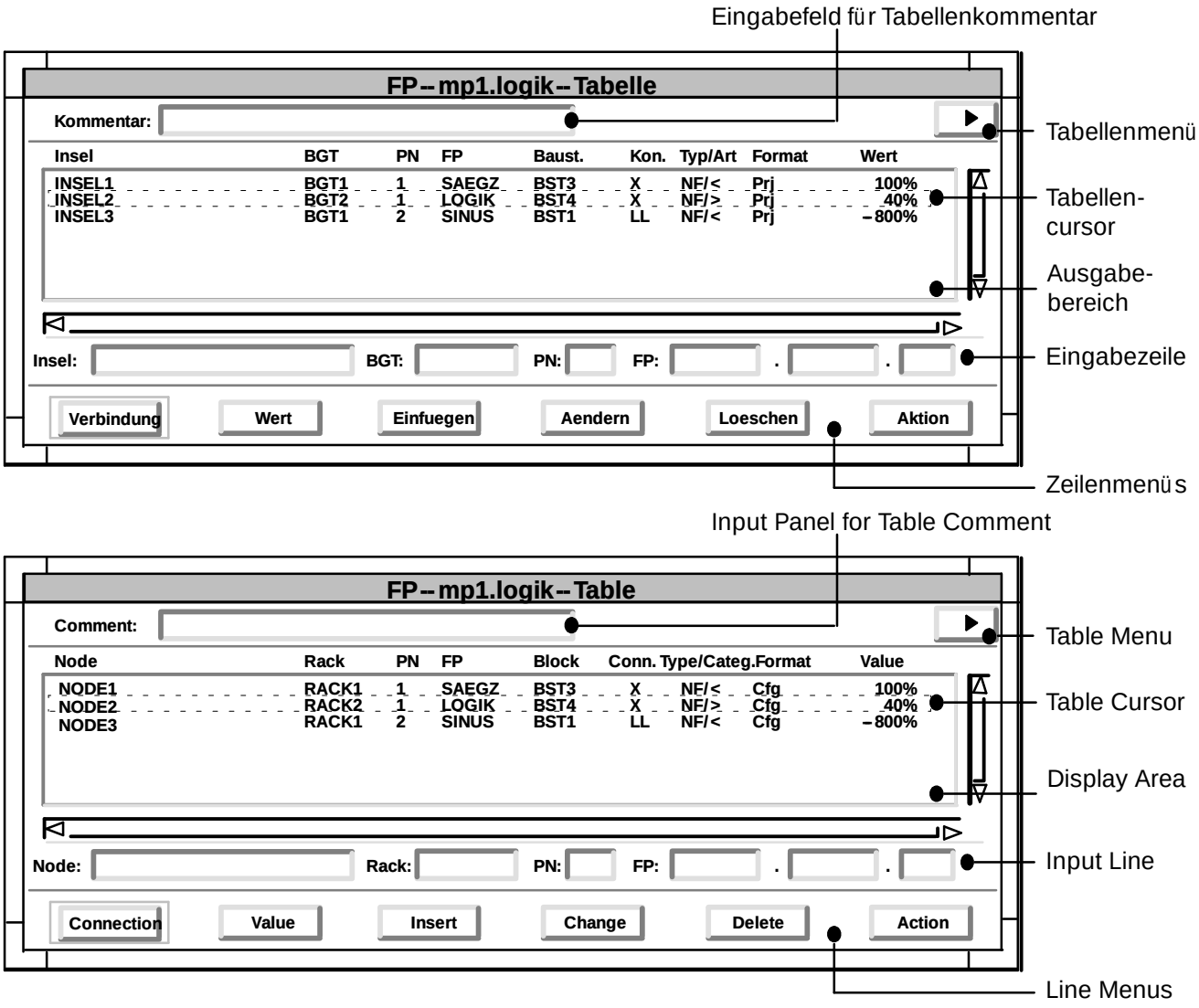


Diagram 1/56: Representation of a Connector Table

1.7.3 Functions for Processing the Complete Table

A connector table contains the following elements for processing the total table:

- Input Panel for Table Comments:
  Each connector table possesses an input panel in the first line, into which a comment for describing this table can be entered.

- Table Menu:
  Pressing the button ‘> ”causes a pop-up menu to appear with the following selection possibilities:

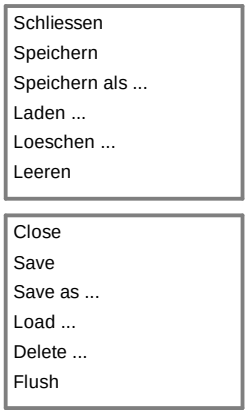


Diagram 1/57: Table Menu

- Close:
  The table is closed, modifications are **not** saved.
- Save:
  The table is saved into its own name and remains open.
- Saved as ...:
  The table is saved under an unrestricted name. Enter the corresponding name into the file selection. The file name extension should be exclusively ‘:asl ’’(for identifying it as a user table).
- Load ...:
  The contents of the table is replaced by the contents of another table. The table to be loaded can be extracted from the file selection.
- Delete ...:
  The **file** of the loaded table is erased. The table is initially retained on the screen and can be saved under another name.
- Flush:
  The **contents of a table** is deleted. This will cause all table lines to disappear. The function ‘Flush ’’has no influence upon the contents of the file.

1.7.4 Connector Input Line

The connector input line is located at the bottom edge of the table. The address components for a connector can be entered via the keyboard into the six input panels. The following keyboard or mouse actions are necessary:

- TAB or Shift + TAB key switches to the next or previous input panel.
- Click with the **right-hand mouse key** on an input panel to open a **list selection** for this input panel. The list selection is described in section 1.5.4.

1.7.5 Selection of Table Lines

The following processing functions are available for influencing individual or multiple lines of a table:

- Click on a table line to set the table cursor (dashed frame) at this line and deselect all lines.
- Shift + click on the table line to select or deselect this line. All table lines are selected if the last (always empty) line of a table is clicked on with shift.
- Move arrow upwards or move arrow downwards to correspondingly also move the table cursor.

Changing the position of the table cursor will cause the connector address in the current table line to be automatically entered as a default into the input line.

1.7.6 Processing Functions of a Table Line

The following functions can be executed with the aid of the buttons at the lower edge of the table:

- Connection:
  The connection table (section 1.9, Diagram 1/64) for the connector of the current table line is popped onto screen. If the connection table is already opened, then it is brought into the foreground with the new connector as a reference connector.
- Value:
  The value modification window (section 1.8, Diagram 1/59) for the connector of the current table line is popped onto screen. If the value modification window is already opened, then it is loaded with the new connector and brought to the foreground.
- Insert:
  The connector at the input line is inserted into the table in front of the current table line. This is also simultaneously registered with SIMADYN D for on-line display.
- Change:
  The connector in the input line replaces the connector at the current table line.
- Delete:
  The connector in the current table line is removed from the table.
- Action:
  A pop-up menu (Diagram 1/58) appears on screen when this button is activated. It contains a series of menu items that can also be found in value display windows (see section 1.6.4).
  All actions from these menus are related to either the **selected** table lines or the current table line, if no table line is selected.

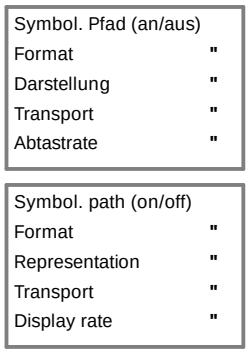


Diagram 1/58: Action – Pop-Up Menu

The following actions can be executed for the desired table lines:

- Symbolic Path (On / Off):
  The symbolic designation or the six digit connector address is displayed as the connector address.
- Format:
  Each table line can be represented in various formats.
- Representation:
  Even in tables, it is permitted to switch from the text representation to bar or curve representation.

– Transport:
  The function permits connectors to be copied or moved from table lines to another table or a screen sub-division.

– Display Rate:
  The display rate can be toggled between ‘Slow ’’and ‘Fast ’’for the selected table lines.

1.8 Value Modification of Input Connectors

A special input window (Diagram 1/59) has been designed for the on-line modification of constant values. This window can be popped onto screen from the basic function FB/CONN – Configure ’as well as from the value display windows and the connector tables.

Value modifications are possible in all number based connector types (B1, N2/N4/NF, E2, I2/I4, O2/O4, V1/V2/V4, T2/TF, D2, R2/R4) as well as for strings (connector type NS).

Remark: Only strings, that are **not longer in length** than the original configured string, are permitted as a new value for connectors of type **NS**.

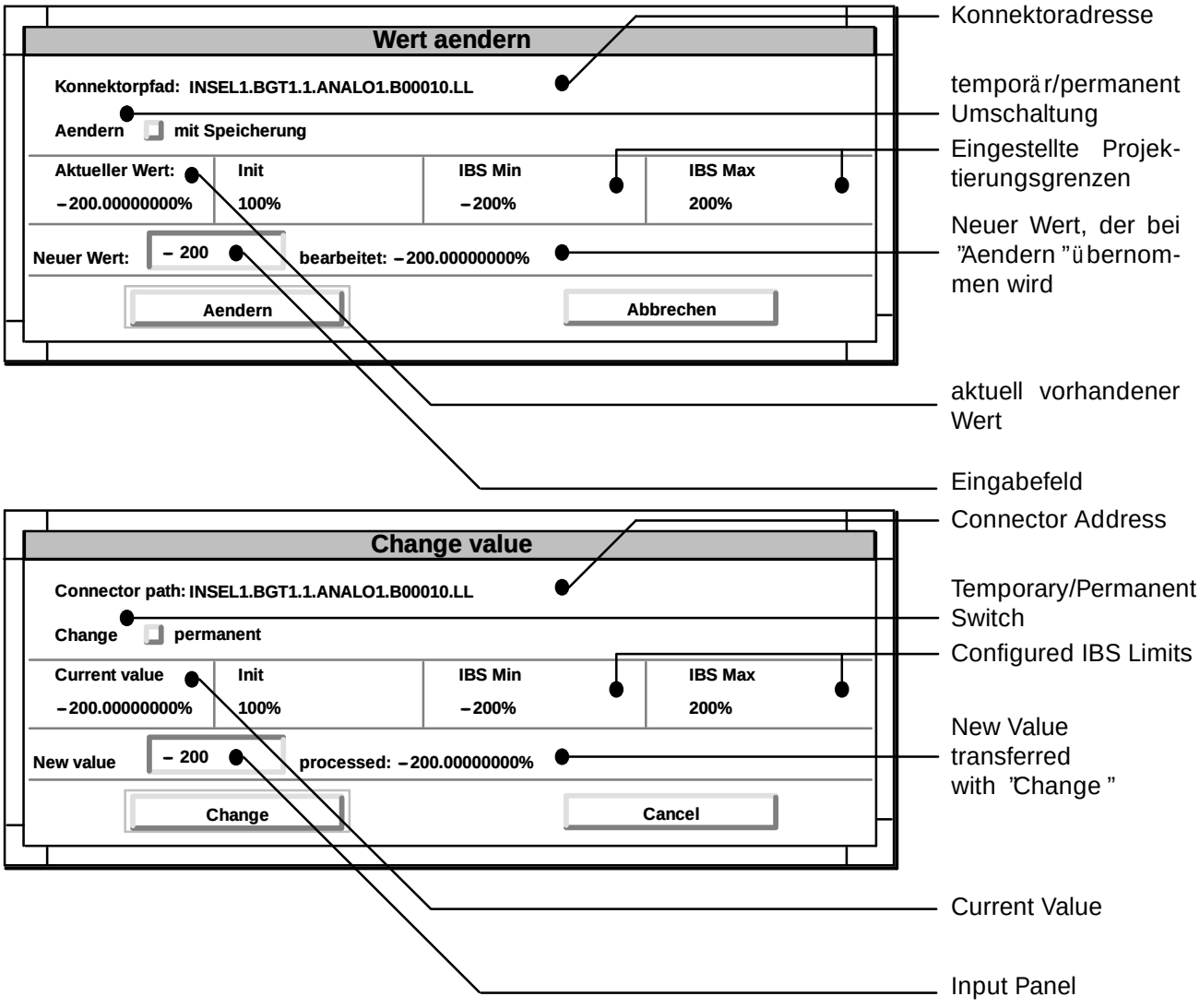


Diagram 1/59: Value Modification Window

The specification of the connector, whose value is to be modified, is located below the title bar. The button ‘Permanent ’’defines whether the modifications are of permanent or temporary nature (see page 6–1). The current value present at the connector is cyclically displayed in the panel ‘Current Value ’’. The configured values for INIT, IBS Min as well as IBS Max are available as further information.

A new value can be defined for this connector in the input value ‘New Value: ’’. During the input of digits, the output panel ‘Processed: ’’indicates which value has been detected as a result of the entries in the text panel.

The input of invalid characters or numbers is acknowledged with an acoustic sound; the previously valid value in the panel ‘Process: ’’is retained.

The following buttons are located in the last line:

- Change:
  The value displayed in ‘Process: ’’is entered into the specified connector on-line. After a short reaction period, the display ‘Current Value: ’’switches to this value. The window remains open. This function can also be activated by the return key.
- Cancel:
  The window is popped off screen. This function can also be activated by Alt + F4.

Remark: It is sufficient to activate the switch ‘Permanent ’’and subsequently to repeat the ‘Change ’’button activation for the permanent acknowledge of a temporary modification.

An automatic update of temporary modifications is possible using the function ‘EEPROM Update ”
(see section 1.12.2).

If the connector, whose value is to be modified, is displayed in the right-hand or left-hand screen sub-division, then the value modification is also completed in the configured system.
If the connector to be modified is not displayed, i.e. for modifications originating from a connector table, then a warning window (Diagram 1/60) is initially popped onto screen before implementation.

If the warning window is acknowledged with ‘OK ’, then the change is implemented on-line but not in the configured system. The **consistency** between the configured system and on-line states is therefore **no longer guaranteed** !

Abort the on-line modification with ‘Cancel ’in order not to lose the consistency, load the sheet onto which connector to be modified is displayed into the left-hand or right-hand screen sub-division and subsequently repeat the value change.

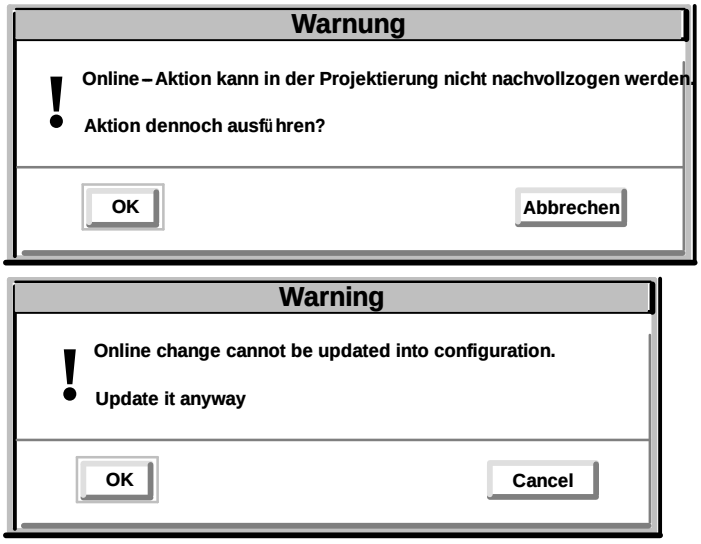


Diagram 1/60: Warning Window for On-Line Modifications, that cannot be Updated in the Configured System

If you want to change a connected input connector, then the window ‘Error Message ’is popped onto screen. The lead back to the value modification window is released using the button ‘Cancel ’.

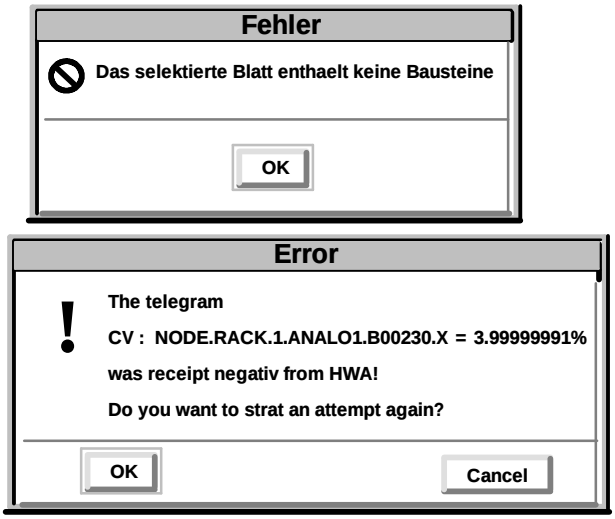


Diagram 1/61: Error By Connected Input Connectors

If you want to change a connected output connector, then the window Error Message ' is popped onto screen as in diagram 6:60 . The lead back to the value modification window is released using the button OK '.

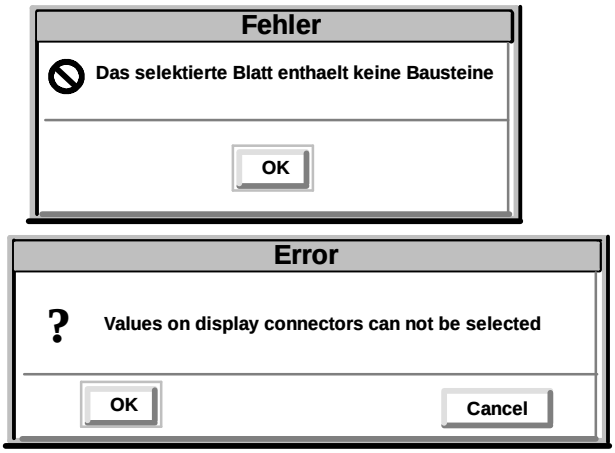


Diagram 1/62: Error By Output Connectors

If you want not to change an initialisation connector permanently, then the window Error Message ' is popped onto screen as in diagram 6:60. The lead back to the value modification window is released using the button OK '.

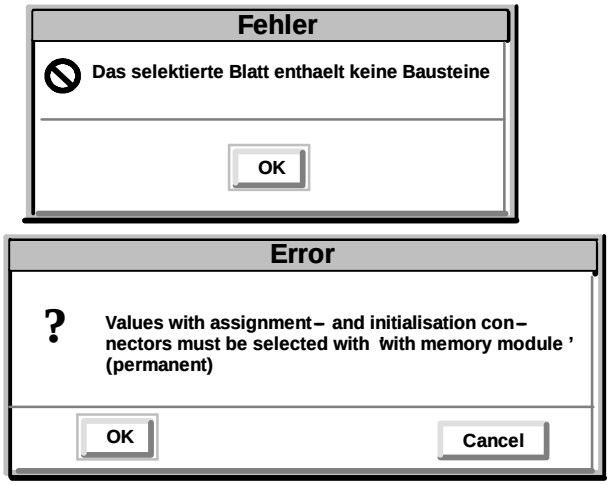


Diagram 1/63: Error By Initialisation Connectors

1.9 Processing the Connection Table

Connections from input or output connectors can be modified, with the aid of the connection table, without actually loading the sheet of the corresponding function packet.

The connection table is popped onto screen in the following manner:

- Menu item ‘Connection ’’in the pop-up menu of a value display window (review section 1.6.4.3):
  The connection table is opened with the connector indicated in the value display window.
- The button ‘Connection ’’in a connector table (see section 1.7.5):
  The connection table is opened with the connector of the current table line.

If the connection table is already opened, then it is loaded with the desired connector and brought to the foreground.

1.9.1 Representation

The connection table generates a connection to a so called **Reference Connector**, which can either be an input or output. If the reference connector is chosen to be an input, then a maximum of **one** further connector can be located in the display area of the connected connectors. If the reference connector is an output, then several inputs can appear as connected connectors.

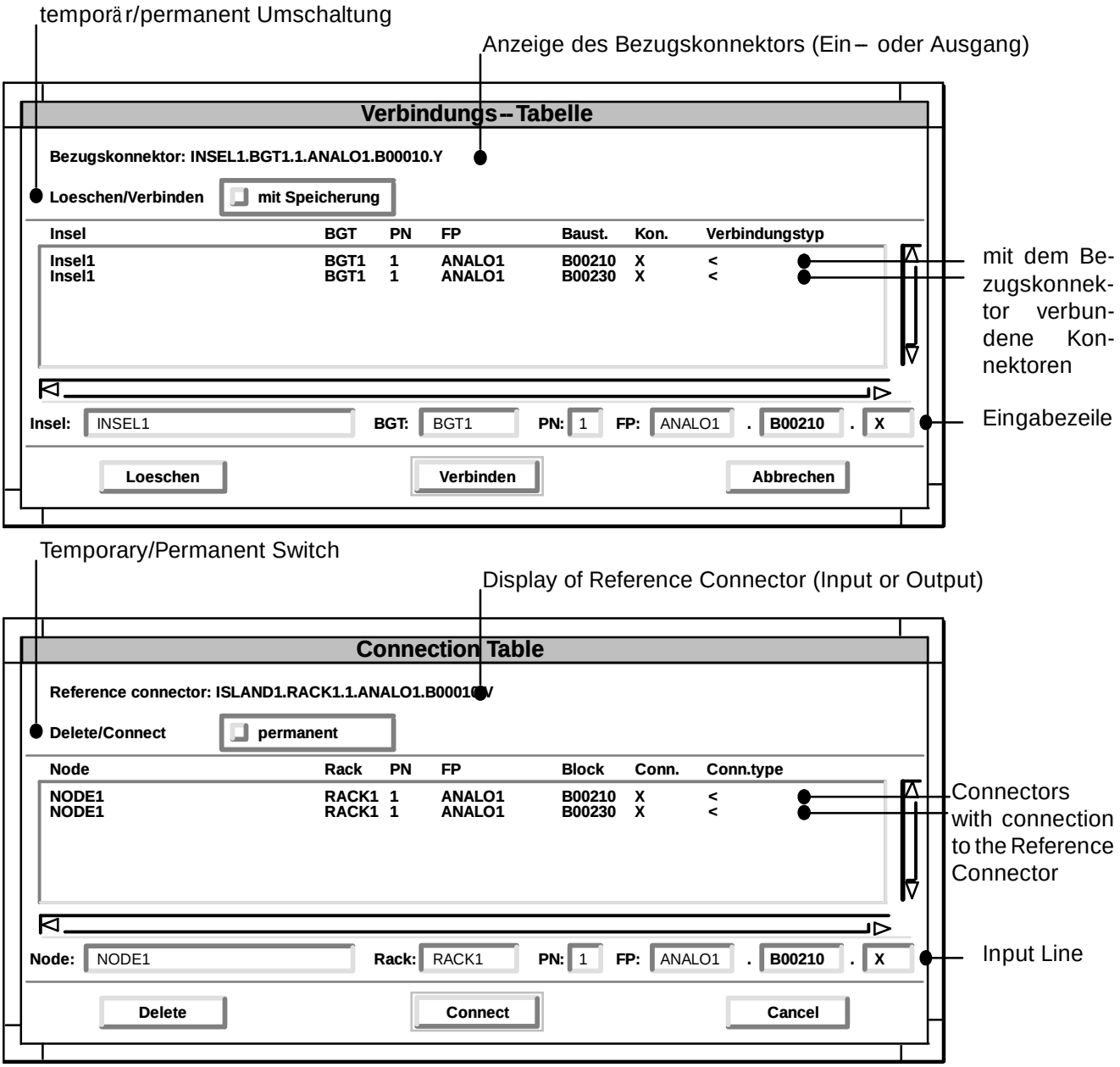


Diagram 1/64: Connection Table

1.9.2 Processing Functions

The following functions are available for processing the displayed reference connectors in the table:

- The switch ‘Permanent ’’specifies whether the subsequent modifications (connections inserted or deleted) are of a permanent or temporary nature.
- Clicking a table line will cause it to be selected and have it s connector address transferred to the input line.
- The panel s node, Rack, PN and FP are disabled for entries. Modifications are only permitted in the block and connector designator input panels. List selections are available for these panels by clicking the right-hand mouse key (see section 1.5.4).

The following functions can be activated by the buttons located at the lower edge of the table:

- Delete
  The connector at the current table line is removed from the list of the connections with the reference connector. The location of the switch ‘Permanent ’’defines whether this is a temporary or permanent modification.
  Remark: Only connections **between function blocks** can be deleted on-line. Connections to the edge strips ($ signals, virtual connections, ...) cannot be modified on-line.
- Connections
  The connector in the input line is connected with the reference connector and inserted into the connection table. The location of the switch ‘Permanent ’’defines whether this is a temporary or permanent modification.
  Remark: Only connections **between function blocks** can be created on-line.
  Connections to edge strips ($ signals, virtual connections, ...) cannot be modified on-line.
- Cancel
  The connection table is closed.

1.10 Processing the Formula Table

The simple modification of several connector values is possible by defining a so called formula table.

1.10.1 Generating a Formula Table

The formula table is inserted into the file ‘online.rzt ’in the directory ‘lnk_onl ’. The precise structure of this file, that can be created with any type of text editor, is described in detail in section 1.13.2.

The specification

```
NODE1.Rack1.3.SAWTTH1.B040.X = 200
```

signifies, for example, that the specified connector is to be set to the value of 200.

The value specification in the formula table must appear in the same form in which it will also be entered in the input panel ‘New Value: ’of the input window ‘Change Value ’(Diagram 1/59). In particular, the unit or dimension text must be omitted.

1.10.2 Representation

Invoking the menu item ‘IBS G – Open Formula Table ’causes the formula table to be popped onto screen with all entries contained in the file ‘online.rzt ’.

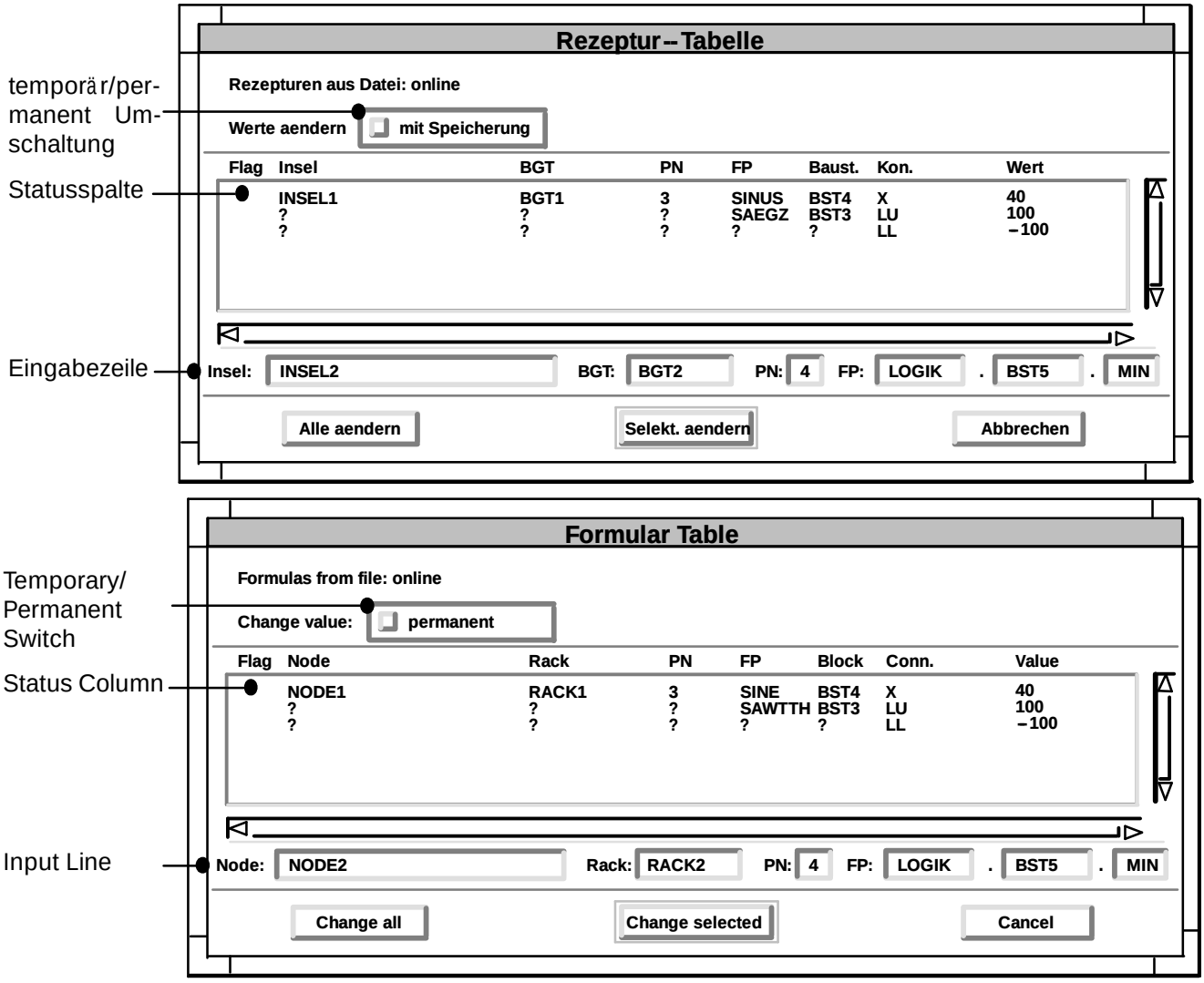


Diagram 1/65: Formula Table

If components of the connector address are missing in the formula file, then these components are indicated with a ? 'in the table.

1.10.3 Selecting Table Lines

Each line in the formula table can be individually selected or deselected. Therefore it is possible to define areas of connectors whose values are to be modified at the same time.

Pressing the left-hand mouse key selects the line and simultaneously deselects previously selected lines.

Pressing Ctrl + left-hand mouse key will select or deselect additional lines.

Pressing Shift + left-hand mouse key selects or deselects an area between the previously processed line and the current line.

1.10.4 Input Line

A connector address can be entered into the input line, that can then be utilized for the components marked with a ? ’in the formula table.

Example:

The formula table contains the lines as shown in Diagram 1/65:

```
NODE1.Rack1.3.SINE.   BST4.X    =  40
?      .?.    .?.SAWTTH. BST3.LU  =  100
?      .?.    .?.?       ?    .LL  = –100
```

The connector line contains the specification:

```
NODE2.Rack2.4.LOGIK.BST5.MIN
```

Therefore the following connector addresses are generated:

```
NODE1.Rack1.3.SINE.BST4.X       =  40
NODE2.Rack2.4.SAWTTH.BST3.LU =  100
NODE2.Rack2.4.LOGIK.BST5.LL     = –100
```

Line 2 will therefore receive 3 components and line 3 will receive 5 components of the connector line.

This mechanism can therefore be utilized to load parameter sets into similar configuration sections. If a series of similarly structured function packets exist in the system, then a complete range of value modifications can be loaded by simply exchanging the function packet name in the connector line of several function packets.

The individual components of the input line can be determined via list selection. Therefore pressing the right-hand mouse key in the corresponding input panel. The list selections will appear on screen as described in section 1.5.4.

1.10.5 Processing Function

The switch ‘Permanent ’’specifies whether modifications are of a temporary or permanent nature (review page 6–1).

The following buttons are located in the last line of the formula table:

- Change All:
  All lines of the formula table are transmitted with their values to SIMADYN D for modification. The modifications are transferred to SIMADYN D **in the specified sequence**, but will be processed independently of one another at that location.
  Each registered value modification is flagged in the formula table with a ? ’in the flag column. This signifies that the value modification is currently being implemented.
  If a positive acknowledge is received from SIMADYN D with respect to the value modification, then a ‘+ ’will appear in the flag column. If the modification cannot be implemented on-line (i.e. because the specified connector does not exist), then a ! ’will appear in the flag column.

  Remark: If the **connector address is not fully specified**, since an address stage has not been defined in the formula file, then the input line will also contain no value. Therefore the associated connector **will not be flagged in the flag column** and no modification is sent to SIMADYN D.
- Selecting Changes:
  As opposed to ‘Change All ’’, only **the selected** elements of the formula table are transferred for modification to SIMADYN D with this function. The sequence is otherwise identical to the function described above.
- Cancel:
  The formula table is closed. Reactivation of the formula table will cause the file ‘bnline.rzt ’’to be reloaded. Therefore the formula table can be modified and reloaded during a function packet editor run.

1.11 Processing Archives

The archive table is intended for saving value responses at connectors. Several archives can be processed within the archive table. In each archive, it is also possible to save several connectors in so called **Connector Channels** and to load these again at a later date.

Archiving values is implemented in the data processing function, which allocates arriving values to a connector channel and saves them on file.

All values of a connector arriving from SIMADYN D are archived at the rate that they are transferred via the connection HWA – > DA. This signifies that even when high system loading occurs, where the graphic display areas of the value display windows sometimes miss out on individual values, all values are always archived.

1.11.1 Representation

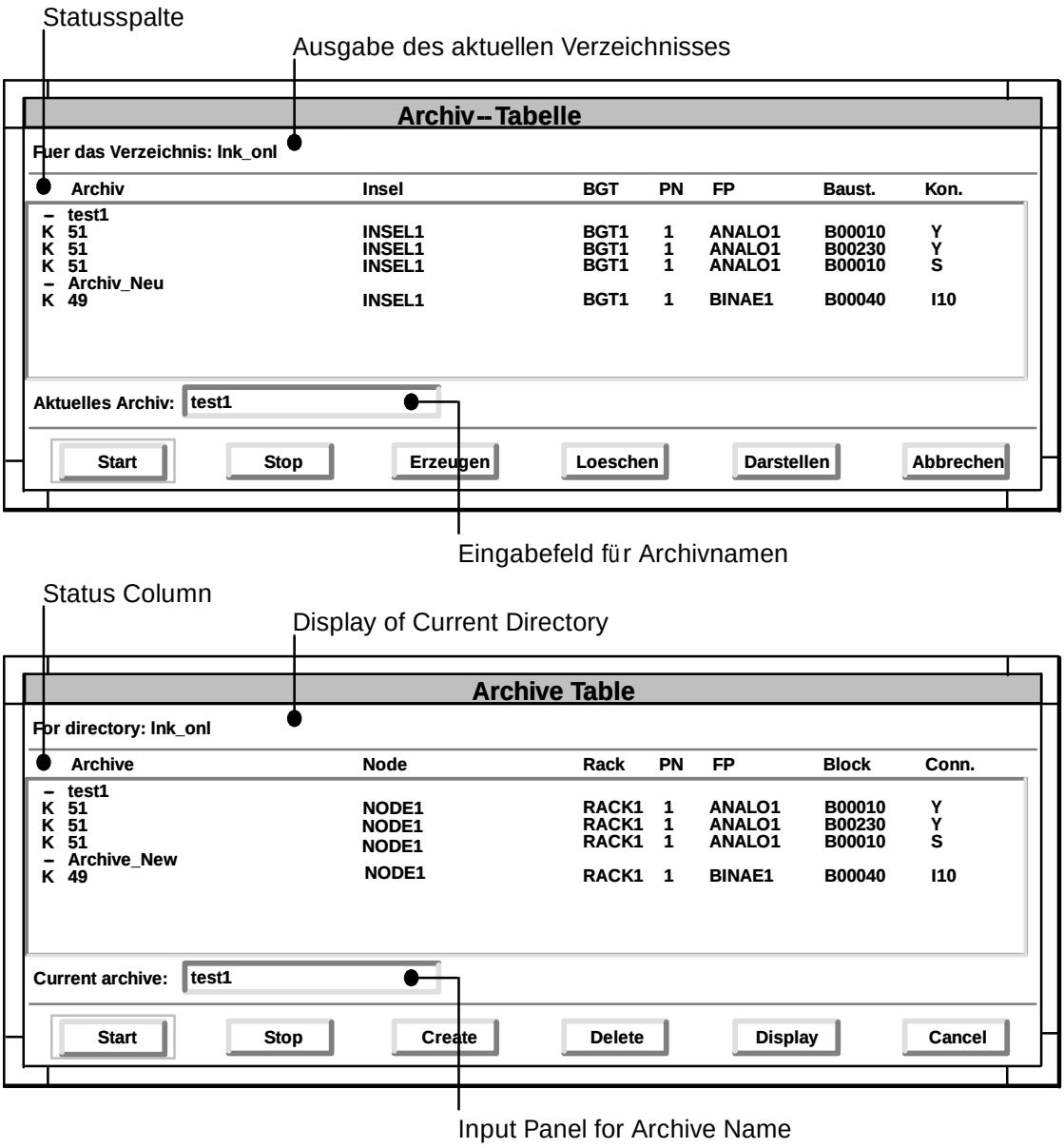


Diagram 1/66: Display of the Archive Table

The archive table displays, after being popped onto screen, all archives located in the directory ‘lnk_onl ”. The archive files are given names in the archive with the extension ‘‘:arc ~ ”.

A status column is located to the left of the archive name and indicates the current processing status of this archive in the form of two characters.

The archive can have the following **processing states**:

- 1st. Character ‘+ ’
  The archive is closed. It s contents is not shown in the archive table.
- 1st. Character ‘– ’
  The archive is open. The subsequent lines in the archive table show the contents of the archive.
- 2nd. Character ’ ’
  The archive is currently in a neutral state.

- 2nd. Character 'C'
  The archive has been modified without a recording having taken place. This occurs when connectors are generated, inserted or deleted.
- 2nd. Character 'R'
  Loading is currently taking place from the archive. This archive can then not be closed.
- 2nd. Character 'W'
  Saving is currently taking place in the archive. The archive can then not be closed.

The lines in a opened archive, following these characters, contain the **connector channels** located in the archive. Each connector channel is displayed with a complete address as well as the number of values saved in the connector channel.

A status column is located to the left of each connector channel, showing the current processing status of each connector channel in the form of two characters.

A connector channel can have the following processing states:

- 1st. Character 'K'
  This identifies a connector channel.
- 2nd. Channel ' '
  The connector channel is not being read.
- 2nd. Character 'R'
  The connector channel is being read, i.e. a so called **archive value display window** has being opened.

1.11.2 Processing Functions

Movement within the tables lines of an archive list is possible with the aid of cursor keys or by clicking on a table line. Modifications of the current table line will cause the associated archive name to be entered into the input panel as a default.

The following functions can be executed using the buttons located on the lower table edge:

- Start
  The opened archive of the current table line is written. Therefore all the connectors present in the archive are registered in SIMADYN D and all the connector channels are supplied with the arriving values. The values previously present in the connector channels are replaced by the new values. The display in the status panel of the archive toggles to 'W'.

  Remark:
  It is neither possible to attach values to a connector channel nor to write to one or several of **certain** connector channels. The **total** archive is always loaded with all connector channels.
  Any number of archives can be written simultaneously.
- Stop
  The current table line archive is no longer written. All the connectors of the connector channels in this archive are de-registered in SIMADYN D. The display in the status panel of the archive is toggled back to ' '.

  Remark
  After writing, the archives connector channels will not immediately be displayed with their current values. In order to review the number of values written, the archive must be temporarily closed in the archive list and then re-opened.
- Create
  A new archive with the name from the input panel is created in the list of archives. The display in the status panel of the new archive shows '–C' for 'Open' and 'Modified'. No archive file is however created at this moment. This is only implemented when writing to the archive after activation using 'Start'.
- Delete
  If the current table line is indicating an archive, then all connector channels of this archive are removed from the table.

If the current table line indicates a connector channel, then this is removed from the table.
The status line of the archive is switched to 'C'. The contents of the file will not at this moment be modified.

- Representation
  If the current table line is an archive, then the archive is opened (or closed), when previously closed (or open). The status display switches from "+" (or "–") to "–" (or "+"). An archive cannot be closed if it is currently being written.
  If the current table line is a connector channel, then an archive value display window (see section 1.11.3) is opened for this. The status display of the connector channel, as well as the associated archive, is switched to 'R'. The function 'Representation' can also be activated by a double click on the corresponding table line.
- Cancel
  The archive table is closed. This then has the subsequence that the archives been currently written can no longer be written into. All opened archive value display windows are also closed.

1.11.3 Archive Value Display Window

An extended form of the value display window exists for the representation of connector values in a connector channel; known as the so called **Archive Value Display Window** (Diagram 1/67). An archive value display window is open for this connector by double clicking on a connector channel or by activating the selection button 'Representation' with selected connector channel. If this connector is already displayed in an archive value display window, then it will be brought to the foreground.

If an archive value display window is opened, then the connector channel status column is flagged with 'R' in the archive list.

1.11.3.1 Representation

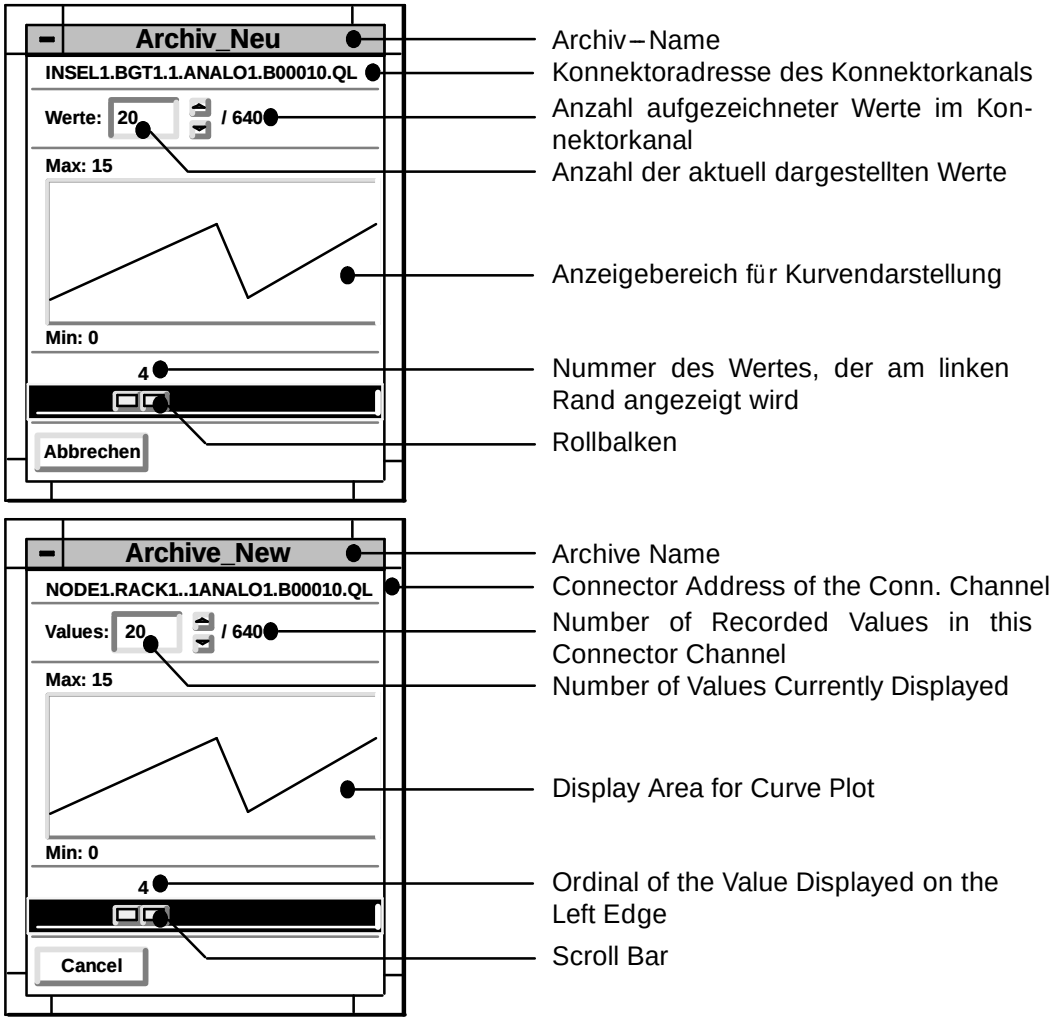


Diagram 1/67: Archive Value Display Window

An archive value display window contains a value display window with curve plot of a variable size. An archive value display window is not attached to a function drawing and can therefore be moved across the complete screen area.

An archive value display window can be closed via the system menu (menu item ‘Close ”of the menu located in the left-hand upper corner of the title bar) or via the selection button ‘Cancel ’. The status column of the associated connector channel changes from R ’to ’ ’in the archive list.

The name of the archive from which the connector channel originates is displayed in the title bar. The connector address is located in the upper most line of the archive value display window.

1.11.3.2 Set-Up Display Area

Any random **section** of the recorded value sequence can be plotted in the curve window of the archive value display window.

The **length of the section** (number of displayed values) can be modified via the text input panel as well as via the arrow buttons above the curve window.
Modifications in the text input panel become active as soon as the arrow buttons are pressed. A standard of 20

values are displayed when the archive value display window is activated; the maximum is equal to the amount contained in the connector channel.

The **position of the section** (ordinal number of the first displayed value) can be set via the scroll bars below the curve window.

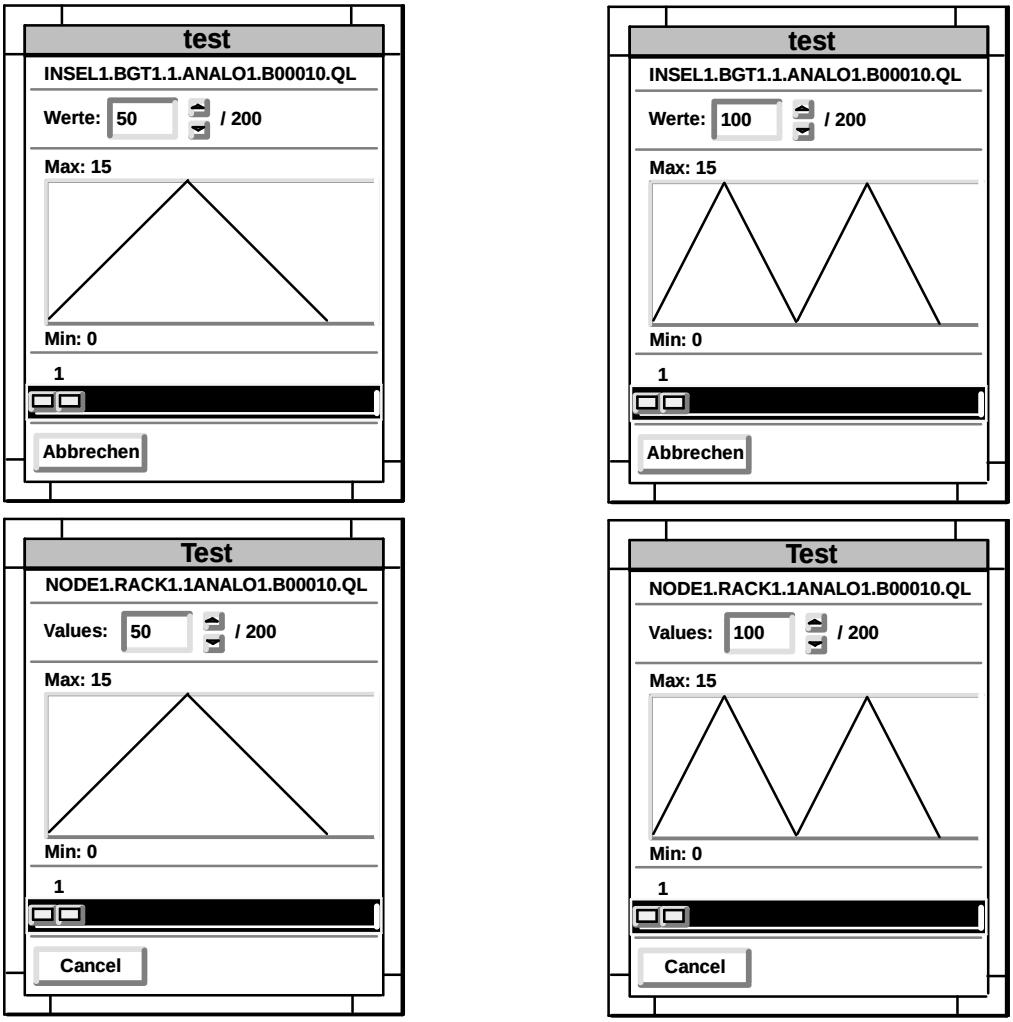


Diagram 1/68: Display Area of the Archive Value Display Window

1.11.3.3 Display Values Off-Line

The curve window of the archive value display window can be shifted across the area of available connector values with the aid of the scroll bar. A ‘Search ’’can be implemented either rapidly forwards or backwards via the scroll bar. The maximum and minimum value indicators are set during the first run through, such that the scaling can be adjusted (auto scaling function).

The size of the display window determines how far the scroll bars can be moved to the right. The scroll bars will only accept values that lie between 1 and (1 + maximum value number – scroll area).

The scroll bar can be moved with the aid of the mouse or the cursor keys. If all the values of a connector channel are displayed in the display window, then the scroll bar indicates ‘1 ’’and cannot be operated.

The size of the archive value display window can be modified, which will cause stretching and compression of the display area.

1.12 The Log Book

All on-line actions from the function packet editor, causing **modifications** in SIMADYN D, are recorded in a log book. The log book is saved in the file ‘online.log ”in the directory ‘lnk_onl ”and is a pure ASCII file that can be processed with a normal editor (**plain text -- log book**).

The log book can therefore be utilized to clarify which on-line actions where implemented during a function packet editor session. This can be carried out both during a function packet editor run as well as after completion of this run. The log book file is over-written after a renewed start of the function packet editor in the IBS G mode.

1.12.1 Layout

The recording starts after the function packet editor is invoked in the IBS G mode. A start message is inserted into the log book. The initialization telegrams transmitted by the function ‘Start On-Line ”to the ‘HWA ”process are then logged (see section 1.4.1.1). Subsequently all important actions transmitted to SIMADYN D are logged.

Exiting the function packet editor will cause an end message to be inserted into the log book. The actions recorded into the log book are coded with the scheme described in the section 1.13.3.

Example of a log file after the completion of a function packet editor session:

```
; Beginn Online-Projektierung: Mon Dec 12 14:31:54 1994
; Einlesen der 'online.ini'-Datei gestartet
DEV :  dev_nr:2 dev_name:/dev/tty1a
SRV :  Insel.BGT.PN DUST1 SERVIC Ein_Kanal-Name
TI  :  150 ms
; Einlesen der 'online.ini'-Datei beendet
CV  :  INSEL1.BGT1.1.ANALO1.B00010.X =  10%
CV  :  INSEL1.BGT1.1.ANALO1.B00010.X =  20%
CC  :  INSEL1.BGT1.1.ANALO1.B00130.Y -> B00010.QU
RC  :  INSEL1.BGT1.1.ANALO1.B00020.Y
; Ende Online-Projektierung: Mon Dec 12 14:49:02 1994
; Start Online-Configuring: Mon Dec 12 14:31:54 1994
; Reading file 'online.ini' running
DEV :  dev_nr:2 dev_name:/dev/tty1a
SRV :  Node.Rack.PN DUST1 SERVIC A_Channel_Name
TI  :  150 ms
; Reading file 'online.ini' completed
CV  :  NODE1.RACK1.1.ANALO1.B00010.X = 10%
CV  :  NODE1.RACK1.1.ANALO1.B00010.X = 20%
CC  :  NODE1.RACK1.1.ANALO1.B00130.Y -> B00010.QU
RC  :  NODE1.RACK1.1.ANALO1.B00020.Y
; End Online-Configuring: Mon Dec 12 14:49:02 1994
```

1.12.2 EEPROM Update

The possibility exists to make temporary modifications of value changes at connectors to be automatically permanent. The so called EEPROM update therefore requires the information from an additional log book, the binary log book (‘online.log ~ ”). This log book is automatically maintained by the system in parallel to the plain text log book (review section 1.12).

1.12.2.1 Invocation

The EEPROM update is invoked via the menu item ‘IBS G – EEPROM Update ”. The automatic update is then immediately started.

1.12.2.2 Working Screen

A working screen (Diagram 1/69) appears during the update procedure. Each value modification is displayed in this window during its processing, so that the progress of the processing can be tracked.

Activating the ‘Cancel ’’button will abort the procedure. All update actions displayed up to this timepoint have been completed in SIMADYN D.

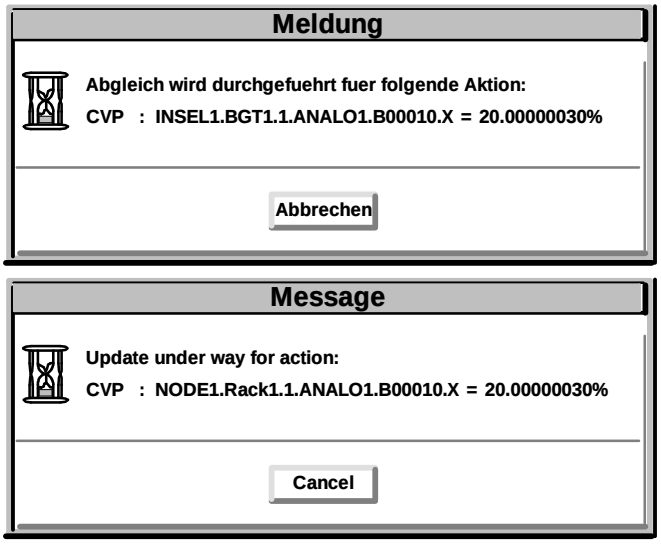


Diagram 1/69: EEPROM Update – Working Screen

A remark window is popped onto screen when errors occur and must be acknowledged by the user. The function is subsequently terminated.

A window (see Diagram 1/70) appears on screen after completion of the total function. It indicates the end of the action and must be acknowledged.

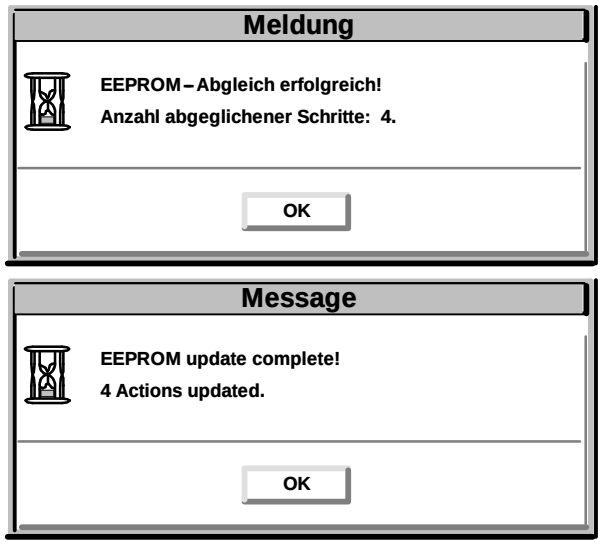


Diagram 1/70: EEPROM Update – End Message

If the function is invoked immediately after a successful completion of an EEPROM update, then no further update steps will be undertaken and a window is popped onto screen (review Diagram 1/70) indicating 0 'updated steps.

Remark: The automatic EEPROM update is only intended for value modifications and is not available for connection modification.

1.12.2.3 Effects on the Log Book

The start of the EEPROM update is inserted into the plain text log book as a comment. Subsequently all necessary actions required for the update are entered The completion of the EEPROM update is also recorded in the log book.

Example of a log book with EEPROM update:

```
; Beginn Online-Projektierung: Mon Dec 12 14:31:54 1994
; Einlesen der 'online.ini'-Datei gestartet
DEV :  dev_nr:2 dev_name:/dev/tty1a
SRV :  Insel.BGT.PN DUST1 SERVIC Ein_Kanal-Name
TI  :  150 ms
; Einlesen der 'online.ini'-Datei beendet
CV  :  INSEL.BGT.PN.FP.BST.KON1 = 10%
; ---- EEPROM-Abgleich START ----
CVP :  INSEL.BGT.PN.FP.BST.KON1 = 10%
; ---- EEPROM-Abgleich ENDE ----
; Ende Online-Projektierung: Mon Dec 12 14:49.02 1994
; Start Online-Configuring: Mon Dec 12 14:31:54 1994
; Reading file 'online.ini' running
DEV :  dev_nr:2 dev_name:/dev/tty1a
SRV :  Node.Rack.PN DUST1 SERVIC A_Channel_Name
TI  :  150 ms
; Reading file 'online.ini' completed
CV  :  NODE.RACK.PN.FP.BST.CON1 = 10%
; ---- EEPROM update start ----
CVP :  NODE.RACK.PN.FP.BST.CON1 = 10%
; ---- EEPROM update end ----
; End Online-Configuring: Mon Dec 12 14:49:02 1994
```

1.13 Appendix

1.13.1 File Structure 'Symbolic Path Designator"

The configuration file 'Symbolic Path Designator'' is located in the directory 'lnk_onl'' with the name 'online.spb'. The file extension ':spb" stands for symbolic path binary.

This file contains a definition line per connector according to the following scheme:

Connector Address = Symbolic_Name

Connector address can be specified with up to six components, whereby the individual address components are separated with ':'. The connector addresses with fewer address components will be handled in the following manner for symbolic outputs.
Extracted from the total amount of all connector addresses from the ':spb "file, a determination is made by which the most number of conforming address components are found.

Example:

```
NODE.Rack1.1.SAWTTH.B00010.LU = Upper Limit_Vent7
SAWTTH.B00010.LU = Upper Limit_Vent
LU = Upper Limit
1.SAWTTH.B00020.LL = Lower Limit_Vent7a
```

This symbolic name 'Upper Limit_Vent7 ''has been defined, in this example, for **exactly one** connector in the system. All **other** connectors from function packets with the name 'SAWTTH ''from the block 'B00010 ''with the connector designation 'LU ''were designated symbolically with 'Upper Limit_Vent ''. All further connectors with the connector designator 'LU "were displayed with the general term 'Upper Limit ''.

Important: The specifications Node 'and Rack 'are currently not evaluated. The connector addresses should therefore not contain specifications referring to nodes and racks (as shown in the fourth line of the example above).

Remark: Connectors that cannot be allocated a symbolic name conforming with the above described scheme, are displayed symbolically as '??? ''. It is therefore recommended to specify symbolic names in the ':spb "file for all occurring connector designators.

Additional empty lines as well as commentary lines (initiated by '// ') can be present in the file.

The file is evaluated **at the start** of the function packet editor. Modifications to this file therefore only have an effect at the next invocation of the editor.

1.13.2 File Structure 'Formula Table"

The file for the formula table is located in the directory 'lnk_on ''and has the name 'online.rzt ''.

The file has a similar structure to that of the file for 'Symbolic Path Designator ''in the section 1.13.1. It consists of lines of format

Connector Address = Value

connector addresses can be inserted with up to six components. If the initial components of the connector address is missing, then these components can be supplemented within the formula table with the aid of the ? " mechanism (review section 1.10.3).

Values for connectors must be specified in the same format **without dimension text**, just as they are to be entered in the value modification window (review section 1.8).

Important: Specifications for Node 'and Rack 'are currently not evaluated. The connector addresses should therefore not contain specifications for the node and rack.

Example for the formula file from Diagram 1/65:

```
INSEL1.BGT1.3.SINUS.BST4.X = 40
SAEGZ.BST3.LU = 100
LL = -100
NODE1.RACK1.3.SINE.BST4.X = 40
SAWTTH.BST3.LU = 100
LL = -100
```

The ‘‘.rzt ”file can also contain empty lines and comment lines (initiated by ‘// ’). This lines are not displayed in the formula table.

The file is evaluated at each invocation of the function IBS G – Open Formula Table ’. Modifications to the file will therefore only take effect at the next invocation of this function.

1.13.3 File Structure ‘Plain Text Log Book ”

The plain text log book is located in the file ‘bnline.log ”. This file consists of individual lines that either contain comments or log an on-line instruction. Comment lines start with ‘;’.

The on-line instructions are coded according to the following scheme:

```
CV  :  INSEL.BGT.PN.FP.BST.KON = Value
Wert ändern - temporär (CV = Change value)

CVP :  INSEL.BGT.PN.FP.BST.KON = Value
Wert ändern - permanent (CVP = Change value permanent)

CC  :  INSEL.BGT.PN.FP.BST.KON1 -> BST2.KON2
Verbindung erzeugen - temporär (CC = Create connection)

CCP :  INSEL.BGT.PN.FP.BST.KON1 -> BST2.KON2
Verbindung erzeugen - permanent (CCP = Create connection permanent)

RC  :  INSEL.BGT.PN.FP.BST.KON
Verbindung löschen - temporär (RC = Remove connection)

RCP :  INSEL.BGT.PN.FP.BST.KON
Verbindung löschen - permanent (RCP = Remove connection permanent)

DEV :  dev_nr:0 dev_name:/dev/tty2a
Geräteadresse für SIMADYN D-Anschluß definieren (DEV = Device)

SRV :  INSEL.BGT.PN dev_nr DUST1 SERVIC Kanal_Name
Dienstedefinitionen für SIMADYN D (SRV = Service)

TI  :  Zeit ms
Definition der schnellen Zykluszeit
CV  :  NODE.RACK.PN.FP.BLOCK.CONN = Value
Change value - temporarily

CVP :  NODE.RACK.PN.FP.BLOCK.CONN = Value
Change value - permanently

CC  :  NODE.RACK.PN.FP.BLOCK1.CONN1 -> BLOCK2.CONN2
Create connection - temporarily

CCP :  NODE.RACK.PN.FP.BLOCK1.CONN1 -> BLOCK2.CONN2
Create connection - permanently

RC  :  NODE.RACK.PN.FP.BLOCK.CONN
Delete connection - temporarily

RCP :  NODE.RACK.PN.FP.BLOCK.CONN
Delete connection - permanently
DEV :  dev_nr:0 dev_name:/dev/tty2a
Define device address for connection to SIMADYN D

SRV :  NODE:RACK:PN dev_no DUST1 SERVIC Channel_Name
Define Service for SIMADYN D

TI  :  Time ms
Definition of the fast display rate
```

1.13.4 IBS G Configurations

It is not necessary for the computer on which the configuring software is running (i.e. the function packet editor) to also be the computer that forms the interface to SIMADYN D. Two cases are differentiated between, dependent upon configuration.

1.13.4.1 Configuration ‘Local ’’

The local configuration has the three processes ‘FPE ”, ‘DA ”and ‘HWA ” running on the same processor. The default setting ‘Local Host ’’as the processor name of the ‘HWA ’’process can be acknowledged in the ‘Start On-Line ”screen. The processes ‘DA ’’and ‘HWA ’’are automatically generated by the function packet editor at the start.

1.13.4.2 Configuration ‘Remote ’’

It is possible to create a link, via TCP/IP, to the processor on which the connection to SIMADYN D is installed (known as computer 2 in the following). Therefore various configuring computers can utilize this interface. The configuring computer is known as computer 1 in the following.

The following steps are therefore necessary for the link:

- Start the ‘HWA ’’Process on Computer 2:
  This can be implemented via ‘rlogin ’’with a subsequent start of the HWA by ‘lnk_hwa ’’from the STRUC directory.

  Example:
  ```
  rlogin computer2
  cd struc
  lnk_hwa
  ```

  The ‘HWA ”process is now started. The terminal window is disabled for entries until this process is exited.
- Start the function packet editor on computer 1
- Implement connect to computer 2:
  Enter ‘Computer 2 ’’as the computer name of the ‘HWA ’’process in the ‘Start On-Line ”screen of the function packet editor. The connection is created after acknowledging with ‘OK ”.

1.13.5 Connection Cable PC – SIMADYN

Dependent upon the design of the serial interface at the PC, a connection cable according to either Diagram 1/71 or Diagram 1/72 is required.

The following pin assignments are necessary for a cable from the SS4 module to a **9 pin** PC connector:

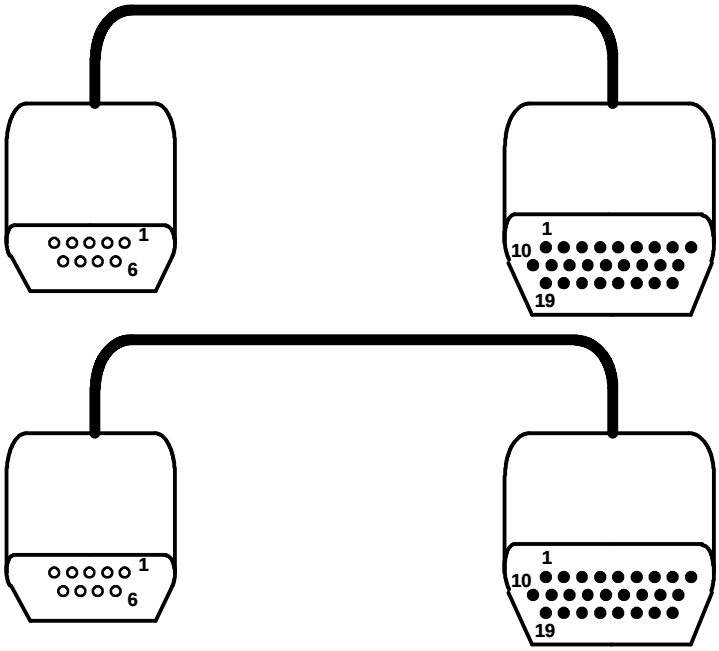


Diagram 1/71: Connection Cable PC (9 Pin)

PC Sub–D, **9 Pin**, Socket			**SS4** Sub–D, Three Row, 26 Pin, Plug	
Pin	Signal		Signal	Pin
2	RxD		TxD	4
3	TxD		RxD	16
5	GND		GND	9
	Screen		Screen	Connector Housing

The following pin assignments are valid for a cable from the SS4 module to a **25 pin** PC connector:

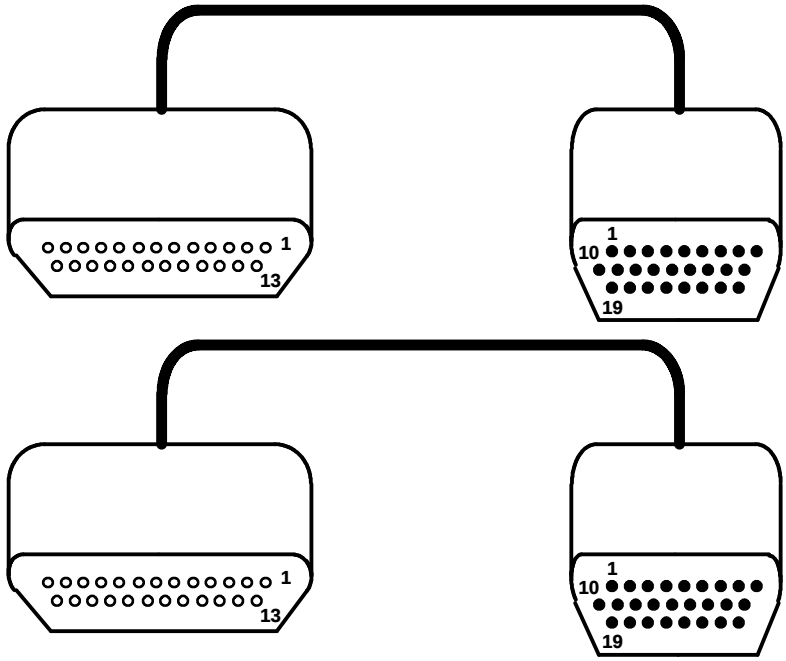


Diagram 1/72: Connection Cable PC (25 Pin)

PC Sub–D, **25 Pin**, Socket			**SS4** Sub–D, Three Row, 26 Pin, Plug	
Pin	Signal		Signal	Pin
3	RxD		TxD	4
2	TxD		RxD	16
7	GND		GND	9
	Screen		Screen	Connector Housing