

Volume

1

**BROSIS**

Engineering GmbH · Software und Industriesteuerungen



# **BROSIS – Com- mander Project Definition**

BROSIS-ENGINEERING GMBH

# **BROSIS Commander**

---

Version 3.2.2 dated 18/11/2008  
© BROSIS-Engineering GmbH  
Am Kirchbühl 9 • D-88099 Neukirch  
Phone (+49) 07528/9516 10 • Fax (+49) 07528/9516 21

---

# Table of Contents

<b>Introduction</b>	<b>1</b>	<b>Texts</b>	<b>34</b>
<b>The Project Definition Modules</b>	<b>2</b>	<b>Station Windows</b>	<b>35</b>
<b>Object Definitions</b>	<b>2</b>	<b>Settings</b>	<b>35</b>
<b>Object Library</b>	<b>2</b>	<b>Fault Visualisation</b>	<b>36</b>
<b>Window Definition</b>	<b>2</b>	<b>Fault Window</b>	<b>36</b>
<b>Skin Definition</b>	<b>2</b>	<b>Special Keys</b>	<b>36</b>
<b>Languages Import – Export</b>	<b>2</b>	<b>Texts</b>	<b>36</b>
<b>Programme Operation</b>	<b>3</b>	<b>Settings</b>	<b>37</b>
<b>The Menus</b>	<b>3</b>	<b>Article Input Window</b>	<b>37</b>
<b>Project</b>	<b>3</b>	<b>Special Keys</b>	<b>38</b>
<b>Languages</b>	<b>4</b>	<b>Texts</b>	<b>38</b>
<b>View</b>	<b>4</b>	<b>Settings</b>	<b>39</b>
<b>Tools</b>	<b>4</b>	<b>Article Data</b>	<b>42</b>
<b>Project Object Data</b>	<b>7</b>	<b>Machine Data Window</b>	<b>43</b>
<b>Shortcut Keys</b>	<b>7</b>	<b>Special Keys</b>	<b>43</b>
<b>Project Info</b>	<b>8</b>	<b>Texts</b>	<b>43</b>
<b>Information</b>	<b>8</b>	<b>Settings</b>	<b>44</b>
<b>PLC Variables</b>	<b>10</b>	<b>Machine Data</b>	<b>45</b>
<b>Texts</b>	<b>11</b>	<b>Measurement Data Window</b>	<b>46</b>
<b>Message Dialogue</b>	<b>14</b>	<b>Texts</b>	<b>46</b>
<b>Protocol</b>	<b>17</b>	<b>Settings</b>	<b>46</b>
<b>Print</b>	<b>18</b>	<b>Measurement Data</b>	<b>46</b>
<b>Station and Object Data</b>	<b>19</b>	<b>Manual Function Windows</b>	<b>47</b>
<b>The Object Library</b>	<b>20</b>	<b>Text</b>	<b>47</b>
<b>Images</b>	<b>20</b>	<b>Settings</b>	<b>48</b>
<b>Article Data and Machine Data</b>	<b>21</b>	<b>Manual Function Data</b>	<b>49</b>
<b>Faults</b>	<b>22</b>	<b>User Administration Window</b>	<b>50</b>
<b>Manual Functions</b>	<b>24</b>	<b>Special Keys</b>	<b>50</b>
<b>Measuring and Display Values</b>	<b>26</b>	<b>Text</b>	<b>50</b>
<b>Document and Code Generation</b>	<b>27</b>	<b>Browser Window</b>	<b>51</b>
<b>Window Data</b>	<b>28</b>	<b>Settings</b>	<b>51</b>
<b>Window (Menu Tree)</b>	<b>29</b>	<b>Browser Data</b>	<b>51</b>
<b>General Window Data</b>	<b>30</b>	<b>Protocol Windows</b>	<b>53</b>
<b>General</b>	<b>30</b>	<b>Special Keys</b>	<b>53</b>
<b>Key Allocation</b>	<b>31</b>	<b>Settings</b>	<b>53</b>

---

<b>Columns</b>	55	<b>Function: MessageOk</b>	95
<b>Print and Print Preview</b>	56	<b>Function: MessageYesNo</b>	95
<b>Examples</b>	58	<b>Function: ValueInputFloat</b>	96
<b>Step Function Window</b>	59	<b>Function: ValueInputInt</b>	96
<b>Text</b>	59	<b>Function: ValueInputBool</b>	96
<b>Settings</b>	59	<b>Function: ValueInputText</b>	96
<b>Stepper Data</b>	59	<b>Function: ValueInputSelection</b>	97
<b>Special Window (Delphi)</b>	60	<b>Function: ValueInputValueRange</b>	97
<b>Settings</b>	60	<b>Function: SetFuncKeyColourByID</b>	97
<b>Parameter</b>	60	<b>Function: SetFuncKeyVisibleByID</b>	97
<b>Display Elements</b>	62	<b>Reading and Writing of PLC Variables</b>	98
<b>The Image Area Editor</b>	62	<b>Miscellaneous</b>	99
<b>The Object Editor</b>	63	<b>Input of Variables</b>	99
<b>The Image Objects in Detail</b>	64	<b>Input of a Value Range</b>	100
<b>Text Objects</b>	64	<b>Input Mask</b>	102
<b>Input/Display Object</b>	65	<b>Opening an Image</b>	103
<b>Key Area Object</b>	66		
<b>Key Object</b>	67		
<b>Skin Key Object</b>	69		
<b>LED Object</b>	71		
<b>Image Object</b>	72		
<b>Analogue Display Object</b>	73		
<b>Progress Bar - Object</b>	75		
<b>LCD Display Object</b>	76		
<b>XY-Plotter Object</b>	77		
<b>Graph Object</b>	81		
<b>Skin Data</b>	83		
<b>Menus</b>	84		
<b>Settings</b>	84		
<b>Title Section</b>	86		
<b>Function Keys</b>	86		
<b>Images</b>	87		
<b>Customised Delphi-Programme</b>	89		
<b>Delphi Settings</b>	89		
<b>Creating a Dialogue</b>	89		
<b>Functions of TFoParentSpecialWindow</b>	91		
<b>Useful Utility Functions</b>	93		
<b>Function: RegisterVariable</b>	93		
<b>Function:</b>			
<b>Dat.Texts.LanguageTextByID</b>	94		
<b>Function: ShowWindow</b>	94		
<b>Function: GetTypeInformation</b>	94		
<b>Function: SetTypeInfoation</b>	95		

---



# Introduction

*Introduction: Project Definition of BROSIS-Commanders.*

The BROSIS-Commander offers a Windows-based User Interface for PLC-controlled plant operation via a convenient Project Administration. Project Administration is based on the *Project definition* and allows access to all the functions of the BROSIS-Commander. Special dialogues can be programmed separately in Delphi, and project-related information, such as object definitions for the object library, or skin definitions, may also be determined here.

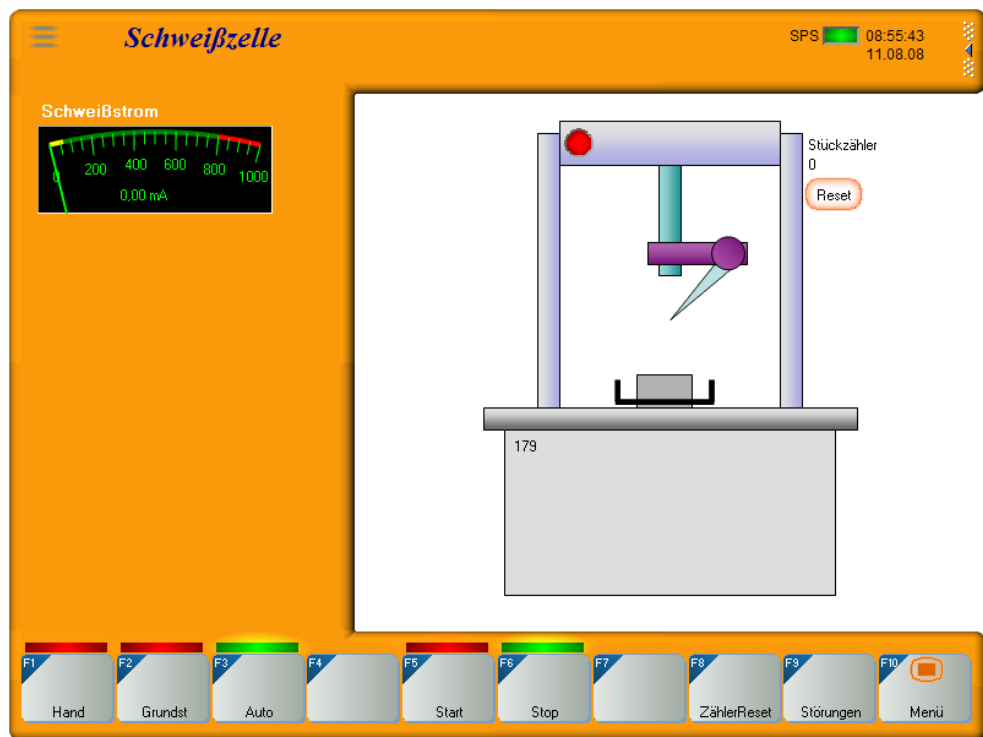


Figure 1 User Interface of BROSIS-Commander.



## The Project Definition Modules

Besides the general project settings, the Project Definition includes such modules as the Object Definition, Object Library, Skin Definition and Languages Import/Export. The individual modules are completely integrated in the Project Definition and assume the following tasks:

### Object Definitions

Objects are the smallest units used by BROSIS-Commander. Ideally, they can map modules within the control unit and take over the complete interaction with them. Objects are, for example, a cylinder, an axis, or a station. A station can be the only object and include further subordinated objects. The following information is maintained for an object:

- **Images** of the object. Display elements may be additionally embedded and shown in the Manual Functions or Station Window.
- **Article data** required by the object
- **Faults** that may be caused by the object
- **Manual functions** by means of which the object can be operated manually.
- **Measuring values** produced by the object.

### Object Library

Each object can be embedded in the object library and therefore, will be available for other projects. To ensure that objects are embedded as quickly as possible, all the names of PLC variables can be provided with wildcards, which will then be replaced with the object name.

### Window Definition

This is where the actual windows and dialogues of the User Interface are defined. Different window types are available for different tasks, e.g. Station windows, Fault windows, Article Data windows...

### Skin Definition

The Skin definition is used to adjust the visual appearance and size of the User Interface. Skins are saved in their own files and therefore, are project-independent.

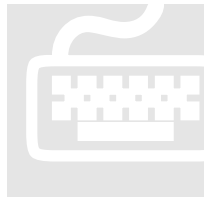
### Languages Import – Export

The entire text of the User Interface is language-independent. With a simple export function this text can be written into a text file and edited or translated, e.g. in Excel. After importing the new language it is available within the User Interface.

## Programme Operation

### *Operation of the Project Definition Interface - Description of Menu Items*

The Project Definition tool offers all the tools you require to design a user interface for BROSIS-Commander. These tools are used for parameterisation only. The controls, both of the plant and user interface, are embedded in the PLC. When opening the programme you will first be prompted to indicate the project to be opened. By aborting the programme you are reverted to the normal user interface, where a new project can be created via PROJECT|NEW.



### The Menus

The menus are subdivided into the main groups, i.e. Project, Edit, Languages, View, Tools and Help. The Edit group offers a selection of optional menu items. All other groups include fixed menu items.

### Project

Menu item	Description
New	Creates a new project.
Open	Saves an existing project.
Save	Saves the data of the currently opened project.
Save as...	Saves the data of the currently opened project under a new file name.
Close	Closes the currently opened project.
XML	Enables saving and loading of projects in XML format.
Create Server CFG	Creates the configuration file for the server. This file is required by the server to register variables, which require registration before the user interface starts running. The path and name of the file are configured in the server.
Terminate	Terminates the programme

**Note**

If the currently opened project includes any changes that are not yet saved, a prompt will appear before overwriting.

**Languages**

The Languages menu allows changing between different languages. In addition, it allows importing and exporting language-dependent text to and from text files. The supported languages are defined by the file **Sprachen.dat**. This file also defines the selected language of the user interface.

Menu item	Description
Export	Allows exporting all texts to a text file, which is separated by tabs. This text file can be edited and translated, e.g. by means of MS-EXCEL™.
Import	Allows importing text from a text file, which is separated by tabs.
Language 1... Language n	Allows toggling between the available languages.

**Note**

The Sprachen.dat file may be built up as follows:  
 1;German  
 2;English  
 3;French  
 Syntax structure:  
 [Number of language];[Name of language]

**View**

Menu item	Description
Library Window	Opens and closes the object library.

**Tools**

Menu item	Description
Select colour	Opens a dialog by means of which the colour can be selected. The tool then determines the decimal and hexadecimal colour values. This can be useful, if the PLC has to send a colour value to the user interface.
Import TwinCAT Variables	Imports all the project icons from a Beckhoff .SYM_XML file and generates the file "Variable.txt". This function enables the Project Administration Tool to



	<p>enter the variables in the form of a shortlist.          To create the Beckhoff .SYM_XML file, select the menu item PROJECT  OPTIONS in TwinCat™ and, under ICON Configuration XML, highlight Create Icon Table. The appropriate file will be created for the next translation.</p>
Export Fault Variables	<p>This dialogue allows exporting the fault variables in any desired format. The following wildcards can be entered in the script:</p> <p><b>{INDEX}</b>          Consecutive index of the fault.</p> <p><b>{STOER_VAR}</b>          PLC variable containing the fault.</p> <p><b>{STOER_VAR OHNE PUNKT}</b>          PLC variable containing the fault, however, without the dots.</p> <p><b>{STOER_TEXT}</b>          Fault text.</p> <p><b>{STOER_TEXT OHNE STATION}</b>          Fault text, however, without a preceding station name.</p>
Skin Definition	<p>Opens the dialogue for editing Skin files (see skin data).</p>
Generate Docu/Code	<p>Generates</p> <ul style="list-style-type: none"> <li>the documentation, or respectively, the code for all project objects (see Documentation and Code Generation).</li> <li>a list showing the allocation of an Article Data entry and its ID. This allocation is required in order to interpret the type data.</li> <li>a list of used PLC variables. <b>At the moment, only the PLC variables of the stations and their objects are taken into account!</b></li> </ul>
Overview of Variables	<p>Creates a text description of all PLC variables used. The declaration is built up as follows:</p> <p>[ID= &lt; Variablen-ID &gt; ]          PLCNo.= &lt; PLC-Number within the Server &gt;          Name = " &lt; Name of the PLC-Variable &gt; "          Refresh= &lt; Refreshrate in ms &gt;          Type= &lt; Data Type &gt;          Default= &lt; Default Value of the Variable &gt;</p> <p>Valid data types are:          BOOLEAN, BYTE, WORD, DWORD, SHORTINT, INT, LONGINT, REAL, LONGREAL and STRING.</p>

	<p><b>Example</b></p> <pre>[ID = 722] PLCNo. = 0 Name = ".bChangeUser" Refresh = 0 Type = BYTE Default = ""</pre> <p>Variables may be changed within the overview of variables. These changes may then be saved by clicking the "Ok" key.</p>
Settings	<p>Allows setting the font types for the programme. This is particularly necessary if you wish to use UNI-CODE characters.</p>

## Project Object Data

*Project Object Data may be used to describe plant details and their functions and capabilities.*

Project-specific data can be displayed at the Project Definition level in the form of a directory tree. Detailed information on a selected element is shown on the right-hand side. To edit stations, objects, or dialogues (Create New, Delete, Copy, etc.) you may either use the appropriate entries in the EDIT menu, or the pop-up menu, which is opened after clicking the right mouse button, or the appropriate shortcut keys.

### Shortcut Keys

The following shortcut keys are used in almost all the modules:

Description	Shortcut Key
Create new entry	CTRL+INSERT
Delete entry	CTRL+DEL
Move entry up by 1	CTRL+U
Move entry down by 1	CTRL+D
Copy	CTRL+C
Insert	CTRL+V
Save Station or Object in Library	CTRL+B

---

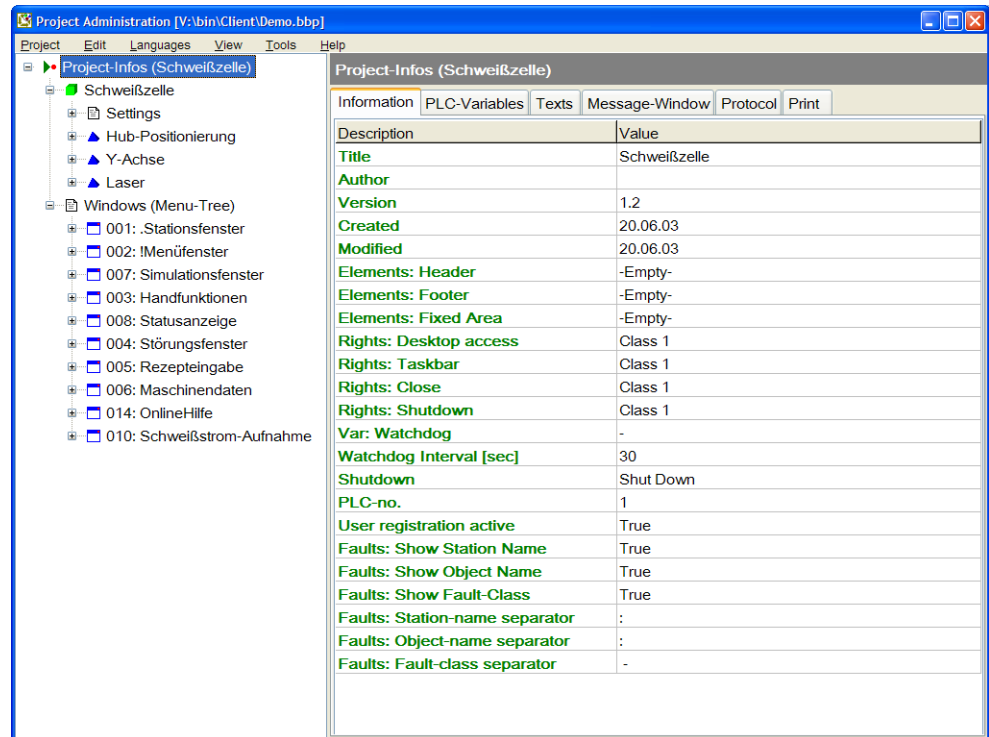
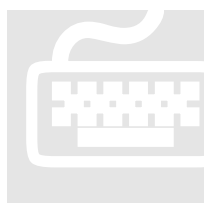


Figure 2 Project Administration with expanded directory tree



## Project Info

Project Info data include general project-related information and settings, e.g. user right definitions and general language-independent texts. By assigning user rights for the desktop and task bar, computer access by unauthorised users can be restricted very easily. Please note that only the desktop icons or respectively, the task bar, will be hidden. The Windows user rights will not be restricted in any other way. The start menu, for example, can be accessed with a double-click on the empty desktop.

Stations can be created under the project node; the node containing all the project windows is located here.

### Information

Includes general project-related information and settings.

Input	Description
Designation	Project designation.
Author	Name of author.
Version	Version number.
Created	Date of project creation.
Changed	Date of last change.
Display elements: Header	Allows positioning display elements in the header of the User Interface (see page 63).

**CHAPTER 3: PROJECT OBJECT DATA**

Display elements: Footer	Allows positioning display elements in the footer of the User Interface (see page 63).
Display elements: Fixed border	Positions display elements at the fixed border of the User Interface (see page 62).
Rights: Access to Desktop	Assigns the minimum user rights allowing access to the desktop of the computer on which the User Interface is running. Access will be restricted for all other users (i.e. the desktop will be hidden).
Rights: Taskbar	Defines the minimum user rights for access to the task bar of the computer on which the User Interface is running. Access will be restricted for all other users (i.e. the task bar will be hidden).
Rights: Terminate	Defines the minimum rights allowing the user to terminate the User Interface.
Rights: Shutdown	Defines the minimum rights allowing the user to shut down the User Interface.
Var: Watch-Dog	PLC variable informing the PLC at cyclic intervals that the User Interface is still running. The PLC is responsible for resetting the variable.
Watchdog-Interval [sec]	Interval (in seconds) at which the “Var: Watch-Dog“ has to be set.
Shutdown	Action to be carried out. Possible values are: <ul style="list-style-type: none"> <li>• Shutdown: The computer is shut down.</li> <li>• Logout: Performs a logout, followed by the Windows Login dialogue.</li> <li>• Restart: The computer is restarted.</li> </ul>
PLC No.	Number of the PLC to which the variables are registered. This is the last fallback level. The PLC numbers can be entered for the stations, objects and for each individual PLC variable.
User Login Active	This function can be used to deactivate the integrated User Login. The user login then has to take place within the programme or via the PLC (see PLC variable for user change, see further below in this Chapter).
Fault: Show Station Name	Indicates if the station name has to be included in the fault description.
Fault: Show Object Name	Indicates if the object name has to be included in the fault description.

Fault: Show Fault Class	Indicates if the fault class has to be included in the fault description text.
Faults: Delimiter Station Name	Defines delimiters between the Station name and Fault description text.
Faults: Delimiter Object Name	Defines delimiters between the Object name and Fault description text.
Faults: Delimiter Fault Class	Defines the delimiters between the Fault class and Fault description text.
Remark	Project-related remarks may be entered in the lower area.

### PLC Variables

Includes PLC variables to be used by the User Interface. All entries are optional. Each variable name may include the wildcard **{PANEL}**, which is substituted by the number of the panel. This allows registration of several panels for one control unit; the variable names have to include the panel number only.

Input	Description
Active Window	The User Interface writes the number of the currently active window into this variable.
Open Window	If a valid window number is written into this variable by the PLC, the appropriate window will be opened.
Active User	The User Interface writes the user right class of the registered user into this variable (right classes = 1...5 for normal users; class 6 for developers).
Active User Name	The User Interface writes the name of the currently registered (active) user into this variable.
Change User	The PLC can perform a user change by means of this variable, e.g: 1: The variable is registered in the form of a digit. In this case, a valid number has to be entered for the right class (1-6 (6 = developer)). 2: The variable is registered in the form of a string. In this case a user name <u>and</u> the right class, or the user name only (if User Administration is used) can be entered. a) Entry of User Right Class and Name In this case the User Right Class has to be separated from the name by a semicolon (Example: "3;Meier"). The name does not have to be created in the User Administration! b) Entry of a Name In this case the User Right Class is determined by means of the User Administration, i.e. the name has to be exist there. If it does not exist, the text of the

	variable will not be cleared! After successful re-registration, a 0 (or, respectively, and empty string) is written into the variable.
Active Language-Code	The User Interface writes the number of the currently selected language into this variable.
Set Language-Code	The variable enables the PLC to set the language.
Colour Status LED 1	This variable defines the colour for LED 1 of the Skin. If no variable is registered here, the LED will not be shown.
Colour Status LED 2	This variable defines the colour for LED 2 of the Skin. If no variable is registered here, the LED will not be shown.
PLC Protocol	This variable enables the PLC to use the Server Protocol utility. The entered text is copied into the protocol file and then overwritten with an empty string.
PLC version string	Makes it possible to display a string from the PLC in the Information dialogue of the Server and Client.

**Note**

The use of the special variable **PLC\_STATUS** may be advisable for the LED variables. This variable returns the following colour values:

- Green: PLC running
- Red: PLC stop
- Yellow: PLC startup

**Example**

Active Window[{PANEL}] registers the variable .ActiveWindow[1] for the first panel, and .ActiveWindow[2] for the second panel.

.ActUser\_{PANEL} is used to register .ActUser\_1.

**Texts**

Includes all the language-independent texts required at the higher level. The “Kennung” (“ID”) column includes the value representing the text. The “Text” column includes the actual, language-independent text.

Input	Description
Input: False	Text for entry and display of the Boolean FALSE value.
Input: True	Text for entry and display of the Boolean TRUE value.
Format Date	Date (hour) format. The following entries are possible:

	<p>d Shows the day as a digit without leading zero.  dd Shows the day as a digit with leading zero.  ddd Shows the weekday as an abbreviation(Sun-Sat).  m Shows the month as a digit without leading zero.  mm Shows the month as a digit with leading zero.  mmm Shows the month as an abbreviation (Jan-Dec).  yy Shows the year as a two-digit value (00-99).  yyyy Shows the year as a four-digit value (0000-9999).  Example: dd.mm.yyyy</p>
Format Hour	<p>Time format (hours). The following entries are possible:  h Shows the hours without leading zero.  hh Shows the hours with leading zero.  n Shows the minutes without leading zero.  nn Shows the minutes with leading zero.  s Shows the seconds without leading zero.  ss Shows the seconds with leading zero. .  Example: hh:nn:ss</p>
Message: SelectArticleData	Message to be returned if no valid Article Data are selected, e.g. when activating Article Data.
Message: Yes Key	Designates the “Yes” key in the message field.
Message: No Key	Designates the “No” key in the message field.
Message: Ok Key	Designates the “OK” key in the message field.
Message: Select Type	Message to be returned if no valid type is selected (e.g. during the Type selection).
Menu Item: Logout	Menu text for User logout (only if the User Administration window is integrated).
Menu Item: Login	Menu text for User login.
Menu Item: Terminate	Menu text for terminating the User Interface.
Menu Item: Shutdown	Menu text for computer shutdown.
Menu Item: Info	Menu text: Information.
Menu Item: Log Output	Menu text: Log output.
Menu Item: Language	Menu text for changing the language.
Menu Item: Connection	Menu text: Connection.
Password: Login failed	Message text after a login has failed.
Password: Abort Key	Designates the Abort key in the Password dialogue.
Password: Change Key	Designates the Change key in the Password dialogue.



Password: Login Key	Designates the Login key in the Password dialogue.
Password: Text Name	Text that appears before the User selection name in the Password dialogue (only if the User Administration Dialogue is embedded).
Password: TextPassword	Text that appears before the Password entry box in the Password Dialogue.
Change Password: Password incorrect	Message text to appear after incorrect entry of the old password or incorrect confirmation of the new password.
Change Password Abort Key	Designates the Abort – key in the Change Password dialogue.
Change Password: Change Key	Designates the Change – key in the Change Password dialogue.
Change Password: Text old Password	Designates the text in the entry box for the old password in the Change – Password dialogue.
Change Password: Text new Password	Designates the entry box for the new password in the Change Password dialogue.
User Right Class 1 ... User Right Class 5	Text describing the individual User Right classes; if a Right Class does not include any text, it is not available.
Keyboard: Transparency	Controls the degree of transparency of the virtual keyboard.
Text: Header-Led 1	Text to be output for LED 1.
Text: Header-Led 2	Text to be output for LED 2.
Text: Menu Tree Title	Title to be output in the display of the menu tree (window selection).
SelectType: Abort Key	Designates the Abort – key in the Select Type dialogue.
SelectType From Article key	Designates the key for copying data from Article Data into the PLC in the Select Type dialogue.
SelectType: To Article key	Designates the key for copying data from the PLC to Article Data in the Select Type dialogue.
SelectType: OK key	Designates a key for copying the entered data.
Enter Value: Abort Key	Designates the Abort – key in the entry field.
Entry Value: Save Key	Designates the Save – key in the entry field.

**Note**

In addition to the default texts, other texts for the message dialogue

can be entered here. The key combinations CTRL+INSERT OR CTRL+DEL may be used to create or delete texts.

### Message Dialogue

The Message dialogue allows displaying a User Interface message from the PLC. This message is presented as a pop-up window. Three freely assignable keys are available to return messages from the user to the PLC.

The displayed text can be defined within the project text; this ensures language-independence. In this case, the PLC will only return the identification of the text within the texts of the project.

The PLC variables, which are required for the Message dialogue, can be entered here. Each variable name may include the wildcard **{PANEL}**, which is substituted with the number of the panel. This allows registration of several panels for one control unit; the variable names only have to include the panel number.

Input	Description
Message text	The PLC enters the message text to be shown, or the ID of language-independent project texts. <b>{PARAM}</b> may be used for any desired number of times as the wildcard for variable parameters.
Parameters	Includes the parameters in texts separated by semicolons. The parameters indicated here substitute the <b>{PARAM}</b> wildcards in the Message text.
Show Message	This variable enables the PLC to show the message. This should not be done until all other settings (message text, parameter, text of keys) are completed. To ensure that the settings before “Show message” are transferred to User Interface, all variables should be set to the maximum refresh rate! By resetting the variables the message can be masked out automatically.
Acknowledge	Indicates the number of the key used to acknowledge the dialogue.
Text Key 1	The PLC enters the designation (text) of the first key (optionally, an ID from the language-independent project texts may be entered). If no text is entered, the key will be masked out.
Text Key 2	The PLC enters the designation(text) of the second key (optionally, an ID from the language-independent project texts may be entered). If no text is entered, the key will be masked out.

Text Taste 3	This variable is used by the PLC to enter the designation (text) of the third key (optionally, an ID from the language-independent project texts may be entered). If no text is entered, the key will be masked out.
Close automatically	Indicates if the dialogue has to be closed automatically after the user has pressed a key (True), or whether the dialogue has to be closed from the PLC via the “Show Message” variable (False).

**Example**

The PLC assigns the following values to the variables:

- Message text = “Confirm Change to Type”{PARAM}/ {PARAM}“
- Parameter = „10;2“
- Text Key 1 = “Message Key OK“
- Text Key 2 = “Message Key Abort“
- Text Key 3 = „“

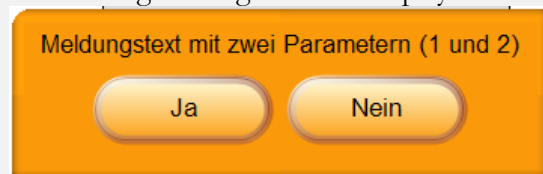
The following language-independent texts are defined for the project:

- Message Key OK = “Yes“
- Message Key Abort = “No“

The PLC now sets the “Show Message” variable to 2 (provided that this is the value to be shown).

The User Interface now proceeds as follows:

- Determine Message Text: “Confirm Change to Type {PARAM}/ {PARAM}“ is not found as an ID in the language-independent project texts; therefore the text is taken over by the PLC as it is.
- The parameters of the message text are replaced. The message text now appears as follows: “Confirm Change to Type 10/2”.
- Determines the texts for the keys (including language-independent project texts). Therefore, the text “Yes”, “No” and “Empty” is determined for the keys.
- The Message Dialogue is now displayed.



If the user presses the No key, the User Interface will write the value 2 (for key no. 2) into the “Acknowledge” variable and then mask out the Message dialogue.

## CHAPTER 3: PROJECT OBJECT DATA

### Protocol

These settings may be used to parameterise the protocol functions of the server. In this case, the server allows generating protocol files and error protocols.

Input	Description
Protocol Filename	<p>Path and filename of the server protocol files. The path may be entered depending on the server directory. The file name is assigned a date automatically.</p> <p>Example:                      .\Protocols\Protocol.txt                      e.g., generates the following protocol name:                      Protocol_2004_08_06.txt</p> <p>For each protocol entry, this will generate a line with the following columns (separated by [TAB]):</p> <ul style="list-style-type: none"> <li>• Date and hour of the entry.</li> <li>• Number of the panel that has generated this entry (-1 =Server).</li> <li>• Registered User (provided that the entry was generated by the panel).</li> <li>• Categorisation of the entry, e.g. “Key” for key presses, “Server” for server protocol entries, etc.</li> <li>• Protocol text.</li> </ul>
Delete after [Days]	Indicates the max. age of the protocol files in days.
Protocol Prog-Start/Stop	Indicates if a protocol (log) has to be prepared for the start and stop of the User Interface and server.
Protocol (Log) Faults	Indicates whether the occurrence or elimination of faults additionally have to be written into the protocol file
Protocol (Log) Dialogue Change	Indicates if a protocol has to be prepared after changing a dialogue in the User Interface.
Protocol Login	Indicates if a protocol has to be prepared at User login.
Protocol Article Data Changes	Indicates if a protocol has to be prepared after Changes of Article Data.
Protocol Machine Data Changes	Indicates if a protocol has to be prepared after changes in the machine data.
Protocol Key Presses	Indicates if a protocol has to be prepared of the key presses.
Faults Filename	<p>Path and filename of the Fault Protocol of the server. The path may be entered depending on the server directory. The filename is appended a date automatically.</p> <p>Example:                      .\Protocols\Faults.txt                      e.g. generates the following protocol filename:                      Faults_2004_08_03.txt</p>

	<p>For each fault, one line is created with the following columns (separated with [TAB]):</p> <ul style="list-style-type: none"> <li>• Date and hour of the start of a fault.</li> <li>• Date and hour of the end of a fault.</li> <li>• Fault category</li> <li>• Fault text</li> </ul>
Delete faults after [Days]	Indicates the max. age of the fault (in days).
Backup Directory	If a directory is entered here, the protocol files will not be deleted, but copied into this directory on expiration of the defined time period. Deleting is the User's responsibility!

**Note**

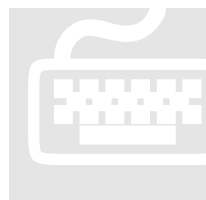
A protocol (log) will only be created for faults and key presses for which a protocol text was entered. Only fault texts of which a protocol text entry already exists will be saved automatically in the protocol texts during the CFG file creation for the server!

**Print**

These settings are used to define the printer settings for special print keys.

Input	Description
Header Title	<p>Text of header title. This text is shown centred in the header of the printout and may include the following wild-cards:</p> <p><b>&lt;PROJECT_TITLE&gt;</b>: Replaced by the title of the project.</p> <p><b>&lt;WINDOW_TITLE&gt;</b>: Replaced by the title of the current window.</p> <p><b>&lt;DATE:format&gt;</b>: Replaced by the current date.</p> <p><b>format</b> may be used to define the date format. Possible values are:</p> <p><b>dd</b> Day as a two-digit value</p> <p><b>ddd</b> Day as an abbreviation, Mon, Tue, ...</p> <p><b>mm</b> Month as a two-digit value</p> <p><b>mmm</b> Month as an abbreviation, Jan, Feb, ...</p> <p><b>yy</b> Year as a two-digit value</p> <p><b>yyyy</b> Year as four-digit value</p> <p><b>hh</b> Hour as a two-digit value</p> <p><b>nn</b> Minute as a two-digit value</p> <p><b>ss</b> Second as a two-digit value</p> <p>Example: <i>&lt;Date:dd.mm.yyyy hh:nn &gt;</i></p>

Footer	Text of the footer, appearing left-justified at the bottom of each page. This text may include the same wildcards as the header.
Left margin [mm]	Indicates the left margin in millimetres.
Right margin [mm]	Indicates the right margin in millimetres.
Top margin [mm]	Indicates the top margin in millimetres.
Bottom margin [mm]	Indicates the bottom margin in millimetres.
Font type: Header	Allows selecting the header typeface.
Font type: Footer	Allows selecting the footer typeface.



### Station and Object Data

Objects are the smallest units used by BROSIS-Commander. Ideally, they map modules within the control and take over the entire “correspondence” with them. In contrast to a normal object, a “Station object” may include subordinated objects (i.e. normal objects, no stations). The settings for the station objects may be selected directly from the Station Object input.

Input	Description
Description	Object description. This is an aptonym, which identifies the object. It is indicated in the event of faults (for manual functions, etc.).
Name	Name of the object instance. The content of this name can be integrated in the variable names of the object by means of the {NAME} wildcard.
Fault Prefix	The text entered here is placed before all faults of the objects.
Author	Name of the object Author (for information).
Version	Version number of the object (for information).
Generated	Date of creation of the object (for information).
Changed	Last date of object change (for information).
Documentation	Documentation remarks for the object (for information).
PLC Number	Indicates the name of the PLC to which the object variables have to be registered (unless defined otherwise).
Manual Function-ID	ID of the manual function of the object. This value is written into the variable defined in the Manual Function window, provided that Manual Function was selected.

Stations can be created by selecting the project node and pressing CTRL+INSERT. Similarly, an object can be added to the selected station by pressing CTRL+INSERT.

The order of the stations and of the objects within a station, can be changed by pressing the key combination CTRL+U or CTRL+D.

**Note**

All variable names should be referenced to the object as far as possible. By using the **{NAME}** wildcard, the object name can be inserted at any desired place at runtime.

**The Object Library**

The object library can be viewed after selecting the menu VIEW|LIBRARYWINDOW.

Stations and objects can be added to the object library by clicking CTRL+B. You will be prompted to enter the name under which the object is to be viewed in the library, and the filename under which it is to be stored. For an object to be shown in the object library, it has to be created in the default path (.\\ObjectLibrary).

To move an object from the library into the current project, you may simply drag it to the desired position with your mouse. Then the object name has to be adapted (if the object variables are provided with {NAME}-wildcards).

**Note**

If a station is moved to the object library and subsequently moved to another station as an object, all subordinated objects will be deleted!

**Images**

This function is used to maintain the object images. These images are displayed under Manual Functions for normal objects and in the Station window for the stations. To create an image you first have to create an **Image List (Bildliste)**. Image lists are used to activate a group of images. For a cylindrical object, for example, you may define the image lists “Installation Horizontal Left”, “Installation Horizontal Right”, “Installation Vertical Top”, “Installation Vertical Bottom” with the appropriate images. When inserting the object from the object library, you only have to activate the appropriate image list.

**Image List**

Input	Description
Active Image list	Highlights the image list including the images to be shown. Only one image list can be active at a time!



Elements	Allows editing the display elements. The first image of the image list is used for the preview.
----------	--

**Image**

Input	Description
ID (Kennung)	Number for unique identification of the image.
Var: Show image	Defines a PLC variable allowing the PLC to change the image currently displayed. This method allows easy simulation of movements (e.g. cylinder retracted, cylinder extended).
Image	This variable is used to assign the image. The following options are available: <ul style="list-style-type: none"> <li>• The image is embedded in the project file;</li> <li>• Saves a reference to the image only. In this case, the image has to be stored in the image directory of the User Interface!</li> </ul> Due to its small data volume compared to BMP, the JPEG format is the recommended image format in this case.
Elements	Allows editing the display elements. The image defined here is used as the background image for the preview.

**Article Data and Machine Data**

Used to enter the object-related article data and/or machine data. Article Data are type-dependent and stored by the User Interface in the server-defined type Information File (e.g. <Server-Directory>\TypeInfo\Types.dat). Machine data, in contrast, will be stored by the PLC itself. Which article data and machine data are to be used in the end is defined during the declaration of Article Data windows.

The list allows changing between the display of the Description and PLC variables (use the context menu (right mouse-click) and the View menu).

The menu items **Export / Import** are used to export or import the highlighted protocol entries into an ASCII file (separated by tabs) or Unicode file. (The first line of the file contains the column title)!

Bezeichnung→Typ→Kennung→SPS-Variable→notwendiges Recht→Kein Recht: Aktion→  
Einheit→Wertebereich: Typ→Wertebereich: Komma→Wertebereich: Min→  
Wertebereich: Max→Wertebereich: Maske→Wertebereich: Liste

Description→Type→ID→PLC Variable→Required Right→No Right: Action→  
Unit→Value Range: Type→Value Range: Comma→Value Range: Min→  
Value range: Max→Value Range: Mask→Value Range:List

Input	Description
Description	Description of the Article Data and/or Machine data entry. The following wildcards may be used for the object descrip-

	<p>tion:</p> <ul style="list-style-type: none"> <li>• <b>{OBJECT}</b> for the object name and <b>{STATION}</b> for the station name.</li> </ul>
Type	Defines either Article data or Machine data.
ID (Identification)	Using the ID allows simple access to an Article Data entry from special Delphi dialogues.
PLC Variable	<p>Indicates the variable including the PLC data.</p> <p>For <b>Machine Data</b> this is the variable the data of which is displayed and can be written into.</p> <p>For <b>Article Data</b> this is the variable used to transfer the Article Data of a specified type from the PLC to the User Interface or vice versa. The Article Data configuration takes place in the Article Data windows.</p>
Required Right	User Right required in order to edit or view an entry.
No Right: Action	<p>Indicates how to proceed if the user does not have the required rights. Possible values are:</p> <ul style="list-style-type: none"> <li>• <b>Window Settings:</b> Uses the settings of the Machine Data window.</li> <li>• <b>Hide:</b> The entry is not shown.</li> <li>• <b>Display:</b> The entry is shown, but cannot be edited.</li> <li>• <b>Display + Login:</b> The entry is displayed, but the Login dialogue will be shown if you attempt to edit the entry. The value can be edited after correct login.</li> </ul>
Unit	Used to enter the physical unit of an entry.
Value Range	Indicates the range of values for an entry. Allows you to define a list of admissible values, for example.
Image	<p>Allows assigning an image to illustrate the Article data or Machine data. Two options are available:</p> <ul style="list-style-type: none"> <li>• The image is embedded in the project file</li> <li>• Only the reference to the image is stored. In this case, the image has to be stored in the image directory of the User Interface!</li> </ul> <p>Due to its small data volume compared to BMP, JPEG is recommended for the image format.</p> <p><b>Cannot be displayed by the User Interface as yet!</b></p>
Description	<p>Allows a more detailed description of the entry.</p> <p><b>Cannot be displayed by the User Interface as yet!</b></p>

### Faults

Used to enter possible faults and warnings that may be caused by the object. Faults and warnings are displayed in different colours.

The overview (list) allows you to toggle between the display of the Description and of the PLC variables via the context menu (right mouse-click) and the View menu.

The menu items **Export / Import** are used to export or import the highlighted protocol entries from or into an ASCII file (separated by tabs) or Unicode file. (The first row of the file contains the column title)!

Fault text→Protocol text→PLC variable→Warning→Fault Category→Documentation

Input	Description
Description	<p>Text description of the fault, which is to be displayed on the User Interface. The User Interface adds the name of the object automatically, so that the fault can be assigned very clearly by the operator.</p> <p>This text can be changed dynamically via parameters. For this purpose, parameters can be embedded in the text via <b>{PARAM:&lt;Measuring Value-Name&gt;}</b>. The parameter is replaced by the appropriate measuring value during the display of a fault.</p>
Protocol text	<p>Text to be entered in the Fault/Error Protocol. This is necessary to maintain the protocols in one language.</p> <p>The menu item “Descr.-&gt; Protocol Text“ of the context menu is used to save all the highlighted fault description texts in the list.</p>
Warning	<p>Indicates if this is a warning or fault message. The display colour of the fault ticker differentiates between warnings and fault messages.</p>
Fault Category	<p>This parameter is used to categorise faults in any desired way.</p>
Documentation	<p>File name of a document containing further details or information concerning the fault message. This document has to be stored under &lt;Directory of User Interface&gt;\Docu.  <b>Cannot be displayed by the User Interface as yet!</b></p>
PLC-Variable	<p>Defines the PLC variable indicating the fault. This is a bit variable as a rule. If you wish to define several faults by means of the value of <u>one</u> variable, you will have to register an integer. The value describing the fault has to be entered for the variable under “Fault Active IF”.</p> <p>In this way, these faults can never occur at the same time. An AND linkage can be entered under “Fault Active IF”, e.g. “AND 2”. In this case, there is a fault if (&lt;Value&gt; AND 2) &lt;&gt; 0!</p>
Image	<p>Used to assign an image illustrating the fault condition.</p>

	<p>Two options are available:</p> <ul style="list-style-type: none"> <li>• The image is embedded in the project file;</li> <li>• Only a reference to the image is saved. In this case the image has to be stored in the image directory of the User Interface!</li> </ul> <p>Due to its small data volume compared to BMP, JPEG is recommended for the image format.</p>
Description	Allows a more detailed description of the fault.
Action	Allows describing corrective action in order to remedy the fault.

**Note**

The menu item “Fault List to Clipboard” in the context menu of the object tree allows you to copy a list of all faults (separated by tabs) to a clipboard. This list can be inserted easily, e.g. using MS-Excel™.

**Example**

A parameterised fault text might appear as follows:  
 “Robot Fault No. {PARAM:RobotFault} when picking up the part”  
 {PARAM:RobotFault} will then be replaced with the contents of the “RobotFault” value.

**Manual Functions**

Manual functions enable manual control of an object (during manual operation of the plant). The manual functions of an object are defined by the keys and display options in the Manual Function windows. More complex manual functions, e.g. axis controls, can be defined via **Display Elements in Images (Anzeigelemente in den Bildern)**. This function is useful to define entries, LEDs, switches, etc. for operating an object. If the manual functions are shown in the form of a list in the User Interface, only one row will be available for each object to visualise its status. The function key bar can be used for operation.

In the overview (list) you can toggle between the display of the Description and PLC-variables by right-clicking the context menu and selecting the View menu.

Input	Description
Key no.	Number of the key to be assigned to the function. If the value entered here is higher than the number of function keys, the key will not be displayed. (It has to be obtained from the Image display elements. This option is useful if many different actions are possible.
Key Row 1	Key designation of row 1

Key Row 2	Key designation Row 2
Key Protocol Text	Text to be logged in a protocol after pressing the key.
Input Title	If you wish a value to be entered (Var. Input-Value), this text will define the window title.
Text: Status	If the status of this manual function is reached (Var: Reached), the list will show this value in the Status field. If the object is approaching this status, the text of the key designation will be shown.
Use in List	If this button is active, the Manual Function in the displayed list will be used to determine the status and colour.
Level-List	A manual function (within the <b>List Display</b> ) can only be used if the level of the manual function is greater or equal to the level of the manual function window. In this way, it is possible to configure a manual function window, e.g. for monitoring purposes, or for restricted operation in the automatic mode.
Level-Image	A manual function (within the <b>Image Display</b> ) can only be used if the level of the manual function is greater or equal to the level of the manual function window. In this way, it is possible to configure a manual function window, e.g. for monitoring purposes, or for restricted operation in the automatic mode.
Rights	Defines the minimum User Rights required to operate this Manual Function.
Open Type Selection	If this button is active, the operator is prompted to select a type before pressing the key. This allows writing entries from linked Article Data into “Var: Input Value”, or respectively, importing them from there to the Article Data. In this case, however, the “Var: Input-Value” has to be assigned and an Article Data link has to be entered.
Var: Action Press	Defines the PLC variable into which the defined value has to be entered after pressing the key.
Var: Action Release	Defines the PLC variable into which the defined value has to be entered after releasing the key.
Var: Input Value	If this variable is defined, an input dialogue is opened after pressing the key. The user can enter a value, which is confirmed and then written into this variable. The “Var: Action Press” will not be executed until this process is completed.
Value Range	Indicates the range of values for entry. This can be a list of admissible values, for example.
Var: Release	This variable defines if the manual function is released or

	locked. If the variable does not show the default value, the displayed list will show the key designation in the “Locked” column, and the LED of the key will be lit up in red!
Var: Activated	This variable defines if the action initiated by this manual function is approaching the final position. If this is the case, the yellow LED will be lit up (or assume the colour corresponding to the value of the variable). This is only valid for non-bit variables.
Var: Reached	This variable defines if the action initiated with this manual function has reached its end position. If this is the case, the green LED will be lit up (or assume the colour corresponding to the value of the variable). Again, this is only valid for non-bit variables).
Image	Allows assigning the image to be shown on the function key. Two options are available: <ul style="list-style-type: none"> <li>• The image is embedded in the project file;</li> <li>• Only the reference to the image will be stored. However, it has to exist in the image directory of the User Interface!</li> </ul> The BMP image format should be used here, so as to overlay the keys with transparent images.
Description	Allows describing an entry in more detail. <b>Cannot be displayed by the User Interface as yet!</b>
Article Data Linkage	Indicates all the Article Data of the object. Using this list you can now link one or several Article Data entries with “Var: Input Value”. The linkage will not be active until <b>Open type Selection (Typauswahl öffnen)</b> is activated.

**Note**

- During the key designation the Skin defines which row is to be displayed in which position! It is quite possible that row 2 appears above row 1.
- Complex manual functions can be created via the display elements for the Manual Function image.
- The User Interface either shows several manual functions in the form of a list, or one function in the form of a graph.

**Measuring and Display Values**

Measuring and display data can be displayed either in the Measuring Data window or as display values (for stations and special dialogues) in the Overview. In the Measurement Data window the values are listed in the form of a table.

In the overview, you may toggle between the display of the description and the display of PLC variables (right-click the context menu and select the View menu).

Input	Description
Description	Description of the value.
Factor	Factor by which the value returned by the PLC has to be multiplied before displaying.
Unit	Physical unit of a value.
Decimal Places	Number of decimal places to be displayed for the value.
Name	Name of the measuring value. This name can be used to embed the measuring value in fault description texts by means of parameters (see Faults).
PLC Variable	Defines the PLC variable including the measuring value.
Description	Allows a more detailed description of the manual function. <b>Cannot be displayed by the User Interface as yet!</b>

### Document and Code Generation

The document and code generation function allows entering any desired type of text to the object. The texts may include the following parameters:

Parameter	Description
{DESCRIPTION}	Description of the object.
{NAME}	Name of the object.
{AUTHOR}	Name of the author.
{VERSION}	Version number of the object.

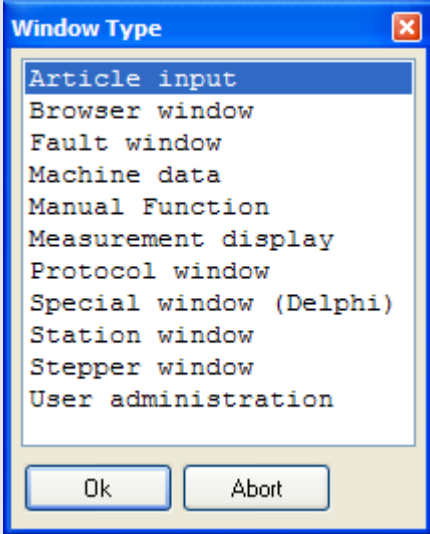
All the texts of the Document and Code Generation within the project may be summarised by the menu item: TOOLS|GENERATE DOCU /CODE.

## Window Data

*While the physical components of the plant are described in the objects, the behaviour and visual appearance of the User Interface is defined by the Window Data.*

Within the window data, dialogues can be created, deleted or changed. These dialogues or windows determine the behaviour of the User Interface and the display of the objects previously generated. The Window Data function is used to define which type of data you wish to display, and where and in which format you wish to display it. In addition, you define who is authorised to view and use your data. Numerous predefined windows are available to meet most of your requirements. For special adaptations, windows can be completely freely designed in Delphi and then embedded.

Windows can be created, moved or deleted by means of the context menu. The following actions are possible:

Menu Item	Description
New Window CTRL+INSERT	Creates a new window after prompting you to define the type of window to be created. 
Delete Window CTRL+DEL	The highlighted window is deleted after a prompt.



Copy CTRL+C	The highlighted window is copied to the clipboard.
Insert CTRL+V	Allows you to insert a window that has been copied to the clipboard.
Up CTRL+U	Moves the highlighted window UP by one position.
Down CTRL+D	Moves the highlighted window DOWN by one position.

**Note**

The position of the window within the list has no influence on the behaviour of the User Interface. The possibility of moving a window has been integrated to facilitate the operation of the Project Definition.



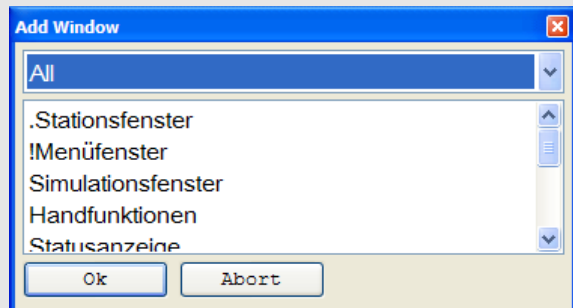
**Window (Menu Tree)**

The menu tree is the shortest way to get to any desired window of BROSIS-Commander. It defines which windows can be accessed directly via the menu tree. Within the menu tree configuration, windows are grouped together, or windows can exist several times. The tree structure can be manipulated via the context menu as follows:

Menu Item	Description
Rename F2	Allows renaming folders and windows. If a window is renamed, this will not affect the actual window title, but only the display in the menu tree.
Copy Window Title F5	Copies the original title of the window.

Add  
CTRL+INSERT

Adds a window to the menu tree. Shows a dialogue from which the window to be added can be selected.



This list is used to highlight and copy one or several windows. The filter function only shows the windows which are not yet in the directory tree.

At the same level	This button defines if a window to be added has to be entered at the same level as the active window, or if it has to be subordinated.
Add Folder CTRL+O	Adds a new folder. This folder is created at the same level as the highlighted entry.
Add Sub-folder CTRL+P	Adds a new folder. This folder is appended and appears below the highlighted folder.
Delete CTRL+DEL	Deletes the highlighted entry.
Up CTRL+U	Moves the highlighted entry UP by one position (within its level). To move the entry to another level, you may drag it with the mouse.
Down CTRL+D	Moves the highlighted entry DOWN by one position (within its level). To move the entry to another level, you may drag it with the mouse.

**Note**

The menu tree should only include windows that are available at any time in the automatic mode. Otherwise there is the risk that the User may be guided to the Manual Functions while in the automatic mode.



**General Window Data**

You will find the General Window Data by clicking directly on the window entry in the menu tree. The details depending on the type of window are at the next lower level. Apart from the general settings, the General Window Data include function keys and language-independent window texts.

**General**

Includes general information concerning the window.

Input	Description
Window Number	Number of the window. As this number can be used to open windows from the PLC (see Project Info), this number has to be unique. A valid number is assigned automatically when the window is generated.
Width - Overview	This button may be used to adjust the width of the overview (list), provided that your window uses the Overview (list) field. If you define 0, the Skin setting will be used.

Title	<p>Title of the window. The title is indicated in the title bar of BROSIS-Commander. The display can be determined by the following prefixes:</p> <ul style="list-style-type: none"> <li>• <b>4Dot:</b> Shows only the title of the project; the window title is masked out.</li> <li>• <b>Exclamation mark:</b> Shows only the title of the window; the project title is masked out.</li> </ul> <p>If none of these signs are used as a prefix, the title will appear as follows:          &lt;Project title&gt; [&lt;Window title&gt;]</p>
Start Window	<p>Indicates if this window is the start window. This window is the first one to be opened after starting the BROSIS-Commander. Only one window may be highlighted; and it must be available in all panels.</p>
Required Right	<p>Defines the minimum right the User must have in order to open this window.</p>
Available in the Panels	<p>Used to enter a list of numbers separated by commas. This list indicates in which panels the window has to be available.</p>

**Example: Window Title**

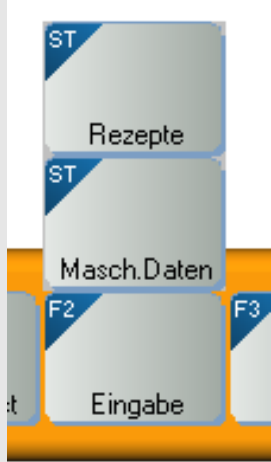
- Project Title = “Assembly Cell“ Window Title = “Main Window“  
 Displayed Title = Assembly Cell [Main Window]
- Project Title = “Assembly Cell“ Window Title = ”Main Window“  
 Displayed Title = Assembly Cell
- Project Title = “Assembly Cell” Window Title = “!Main Window“  
 Displayed Title = Main Window

**Key Allocation**

Includes the definition of function keys. The list of defined function keys is located on the left side. Function keys can be newly created and deleted via the context menu.

Menu Item	Description
Add CTRL+INSERT	Creates a new function key.
Delete CTRL+DEL	Deletes the highlighted function key(s).
Copy CTRL+C	Copies the highlighted function key(s) to the clipboard.
Insert CTRL+V	Inserts function keys from the clipboard.

The following data can be entered for each function key:

Input	Description
Text Line 1	Designates the first text line (row) of the key.
Text Line 2	Designates the second text line of the key.
Protocol text	Logs text in a protocol when the key is pressed.
External Programme	Allows accessing an external programme. Use this button to enter the programme including the path name. If the programme should prompt you to enter access parameters, enter these parameters in quotes, e.g.: C:\Windows\notepad.exe „c:\temp\Hallo.txt“
Linked Keys	 <p>A list of key numbers separated by commas can be entered here. The keys corresponding to these numbers are built up with each key operation. It is useful to define keys that are not within the visible range (usually these are the numbers &gt; 12).</p>
Key Number	Number of the key. Usually the keys 1 to 10 or 12 are visible. The Skin defines how many keys are displayed. Keys that are beyond this range may be used by means of the “linked keys”. If a key number is within the range of the function keys, this key may also be operated via the hardware function keys.
Required Right	Defines the minimum right authorising the User to operate the key.
No Right: Action	<p>Defines the behaviour of the key, if the user should not have the required User Right to operate it. The following actions are possible:</p> <p><b>Hide:</b> The key or, respectively, its designation and image, will be masked out.</p> <p><b>Display Key:</b> The key is displayed in its normal form.</p> <ul style="list-style-type: none"> <li>• <b>Display (without image):</b> The key designation is displayed; the image is masked out.</li> <li>• <b>Display key + Message:</b> The key is displayed in its normal form; after pressing it again a message will appear stating that there is no User Right.</li> <li>• <b>Display key + Message (without image):</b> The key des-</li> </ul>

	<p>ignation is displayed; the image is masked out. If the key is pressed again, a message will appear stating that there is no User Right.</p> <ul style="list-style-type: none"> <li>• <b>Display key + Login:</b> The key is shown in its normal form. If it is pressed, the login dialogue will be opened in case there is no User right:</li> <li>• <b>Display key + Login (without image):</b> The key designation is displayed; the image is masked out. If the key is pressed, the login dialogue will be opened in case that no User Right has been assigned.</li> </ul>
True/False LED	<p>If this entry is “True”, the colour of the function key LED will not be determined by the colour value of the PLC variable “Var: Colour”, but by the properties “LED colour True” and “LED Colour False”</p> <p>In this case, an AND linkage can be entered for the PLC variable under “Colour or True”, e.g. “AND 2”. This effects that “LED Colour True” will be displayed, if: (&lt;Value&gt; AND 2) &lt;&gt; 0!</p>
LED-Colour True	Colour to be shown by the LED, if the PLC variable “Var: Colour” corresponds to the default value.
LED-Colour False	Colour to be shown by the LED, if the PLC variable <u>does not</u> correspond to the default value.
Var: Press Key	PLC variable, which is set to the entered value as soon as a key is pressed. Entering two values separated by a <b>Slash</b> allows you to toggle between two numerical values.
Var: Release Key	PLC variable, which is set to the entered value as soon as the key is released.
Var: Colour	PLC variable defining the colour of the keyboard LED. If the PLC returns 0, or if the variable is not linked, the LED will be masked out.
Var: Visible	PLC variable defining the visibility of the key. This enables the PLC to show or hide a key.
Var: Row 1	PLC variable defining the first text line (row) of the key.
Var: Row 2	PLC variable defining the second text line (row) of the key.
Press Delta-Value for Var.	This setting is used to increase the value of the PLC variable “Var: Press Key” by the value entered here.
Special ID	<p>Special ID of the key. The number of possible special IDs differs depending on the window. The following special IDs are possible for each window:</p> <ul style="list-style-type: none"> <li>• <b>Display Keyboard:</b> Shows the virtual keyboard.</li> <li>• <b>Print Screenshot (Detail):</b> Prints a detailed area of the User Interface.</li> </ul>

	<p>A prompt asking for the printer will appear first.</p> <ul style="list-style-type: none"> <li>• <b>Print Screenshot (Overview + Detail):</b> Prints the overview and detailed area of the User Interface after a printer prompt.</li> <li>• <b>Print Screenshot (All):</b> Prints the complete window of the User Interface after a printer prompt.</li> <li>• <b>Print Screenshot (Detail) / Standard Printer:</b> Prints the detailed area of the User Interface on the standard printer.</li> <li>• <b>Print Screenshot (Overview + Detail):</b> Prints the overview and detailed area of the User Interface on the standard printer.</li> <li>• <b>Print Screenshot (All):</b> Prints the complete window of the User Interface on the standard printer.</li> <li>• <b>Cleanup Monitor:</b> Opens a window, which blocks the operation for 30 seconds. (This duration may be adjusted in the INI-file of the Client via ( [Settings] CleanUp duration=30 ) ).</li> <li>• <b>Close Program:</b> Closes the user interface.</li> </ul>
ID for Delphi-Programme	If a value of > 0 is defined here, the Delphi programme created by you will receive information after pressing or releasing a key, and will be able to respond.
Open Window	Selects a window to be opened after pressing the key. In addition, this key may be used to return to the previous window.
Image	Defines the image to be shown on the key.

**Note**

As the position of the two text lines can be freely defined in the Skin definition, Line 1 may not necessarily be the top line (row)!

**Texts**

Includes all the texts required for the selected window. The text language depends on the entry selected from the LANGUAGE menu.



## Station Windows

A Station window generally displays an overview or detailed view of a station. The image of the station object and of course, its display elements, serve as the background. Active faults can be highlighted within the image, thus enabling the operator to locate the fault position.



### Settings

Input	Description
Station	A Station window has to be allocated to a station. This button is used to define a station.
Var: Current Type	PLC variable including the ID of the current type. This variable is needed to start a type prompt within the display element.
Used Image List	Defines the image list to be used as a standard. The PLC variable defined in this list determines the image to be displayed.
Elements	This button is used to edit the elements of the image list currently being used.
Type Overview	Defines what has to be displayed in the Overview area. the following options are available: <ul style="list-style-type: none"> <li>• <b>No Margin:</b> The overview is masked out, leaving more space for the image.</li> <li>• <b>Display values:</b> Allows displaying Measuring and Display values from the objects of the station.</li> <li>• <b>Display elements:</b> Allows placing freely definable display elements in the Overview area.</li> <li>• <b>ID (Delphi):</b> If this item is selected, the Overview area will be defined by a customised Delphi application.</li> </ul>
Display Values	Defines the display elements to be shown in the Overview area.
Display Elements	Allows positioning display elements within the fixed margin (border); (see Display Elements).
Delphi ID	Number used as identification in the customised Delphi programme.
Fault Display Colour	Colour of fault displays.
Warning Display Colour	Colour of warning displays.

Fault Display Transparency	Indicates the degree of transparency used for the visualisation of warnings and faults. If this value is less than 255, the display will be shown as a transparent area surrounded by a border.
----------------------------	---

### Fault Visualisation

The Fault Display (Visualisation) allows highlighting possible faults within the image of a station. These highlights are displayed in the BROSIS-Commander as soon as the fault has occurred. The Fault window is opened directly by a mouse-click on the fault.

To highlight a fault you need the position and the image to be displayed. Press the  button and select the fault to be highlighted from the list, followed by a mouse-click on the position within the image. To delete the highlighted fault, select the fault from the list and press the -button.



### Fault Window

The Fault window shows the currently active faults of one or several stations in the form of a table. This window usually includes a key which is used to acknowledge a fault. This key can be implemented by setting a PLC variable. The PLC should reset the faults after setting the variable.

### Special Keys

Input	Description
Toggle between Details	An image, description or corrective action may be entered for each fault = <b>(Details)</b> . This button allows you to toggle between these fault details.
Hide Details	The details will be masked out.
Show Image	Shows the image of the fault being highlighted.
Show Description	Shows a description of the currently active (highlighted) fault.
Show Action	Shows corrective action for the currently active (highlighted) fault.

### Texts

The texts can be found in the General part. The following texts are required for the Fault window:

ID	Description
----	-------------



Title: Description	Text of the title to be displayed above a detailed fault description.
Title: Image	Text of the title to be displayed above the fault image.
Title: Actions	Text of the title to be displayed above the corrective action to remedy the fault.
Title: Fault	Title of the column including the fault text.
Title: Time	Title of the column including the time at which the fault has occurred.

### Settings

Input	Description
Station	This button is used to allocate one or several stations by means of a selection box. The faults of these stations are displayed in the Fault window.
Text Colour: Faults	Defines the colour used to display faults.
Text Colour: Warnings	Defines the colour used to display warnings.
Var: Fault Ticker	If a variable is entered here, the text of the active faults will be cyclically written into this variable. Via this variable, a fault ticker may also be provided in another position, i.e. in the header of the User Interface.
Var: Warning Ticker	If a variable is entered here, the text of the active warnings will be cyclically written into this variable. Via this variable, a fault ticker may also be provided in another position, i.e. in the header of the User Interface.

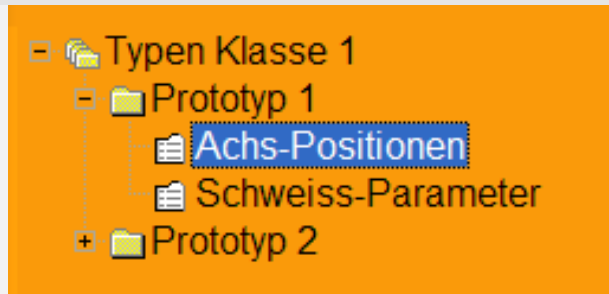


### Article Input Window

The Article Input window is used to store type-dependent information. Article Data may be retrieved by the PLC; the ID of the respective type is used for unique Article Data identification.

For reasons of clarity the window subdivides the types by main types to create a tree structure. Within this structure, Article Data may be subdivided further.

**Example**



In our example the main type “Main Type A” was created. This main type includes “Type 1”. Type 1 includes information that is required by the PLC. This information again is subdivided into material info, speeds, .....

**Special Keys**

Input	Description
Main Type New	Creates a new Main Type
Type New	Creates a new type under the highlighted Main Type.
Type New (Copy)	Creates a new type under the highlighted Main Type. Data will be copied from a selected type.
Delete Type or Main Type	Deletes the highlighted Type or Main Type.
Change Value	Changes the value of the highlighted entry.
Change Type /Main Type	Changes the data of the highlighted type (description and unique ID) or Main Type (description).
Save text as Article Data	Saves the Article Data as a text file for documenting purposes.
Enter desired Type before Key Action	Enters the highlighted type in the variable “Var: Activate Type” before the actual key action, followed by a prompt asking if the type is to be activated (text with ID “Prompt: Select Type”). If the answer is “Yes”, the key action will be executed.

**Texts**

The texts can be found in the General part. The following texts are required for the Article Data window:

ID	Description
Prompt: Select Type	Prompt text to be displayed before activating a type with the special key “Enter selected Type

	before Key Action”.
	This prompt may be answered with Yes or No.
Prompt: Delete Type	Prompt text to be displayed before deleting a Type or Main Type.
Title: Unit	Text of the title for the Unit column in the table of Article Data.
Title: Text	Text of the title for the Text column in the table of Article Data.
Title: Value	Text of the title for the Value column in the table of Article Data.
Title: Type Selection	Title of the dialogue for Type Selection (e.g. for copying a Type).
TypeInput: Abort Key	Designation of the “Abort” key for entering the data of a Type or Main Type.
TypeInput: OK Key	Designation of the “OK” key for entering the data of a Type or Main Type.
TypeInput: Title Type ID	Designation of the entry field for the Type ID (used when entering the Type data).
TypeInput: Title TypeName	Designation of the entry field for the Type name (used when entering the data of a Type or Main Type).

### Settings

Input	Description
Filter for Type	Used to enter a filter designating the Types and Main Types created in this window. Types with this designation can only be displayed in this window. In this way, different groups of type-dependent information can be maintained, e.g. production data and tool data.
Mask for Type IDs	This mask defines which types are to be displayed. The mask has to match the ID of the Type. Masks may include the wildcard * for several characters and ? for one character.
Query before Activating a Type	Defines if a query has to be presented before activating a type (see special keys). The query uses the text with the des-

	ignation: “Query: Select Type“. This text may include the wildcard %s, which is then replaced by the type name.
Assign Type ID Automatically	Automatically enters a unique ID for the type, which can be hidden from the operator.
Width Column Text [%]	Defines the width of the text column as a percentage of the available space.
Width Column Unit [Pixel]	Defines the width of the text column in pixels. The remaining width is used for an input value.
No Rights: Colour	Defines the text colour of input for which the user has no entry rights. (The required rights are set under Article Data).
No Rights: Action	Allows setting a standard action in case that the user should not have any rights for entry. (The required rights are set under Article Data). The following settings are possible: <ul style="list-style-type: none"> <li>• <b>Hide:</b> The entries will not be shown.</li> <li>• <b>Display:</b> The entries are displayed in the colour defined under “No Rights: Colour”, but cannot be edited.</li> <li>• <b>Display + Login:</b> The entries are displayed in the colour defined under “No Rights: Colour”. If the user should attempt to change the entry, he will first be prompted to authenticate and will not be allowed to change the value without the appropriate User rights.</li> </ul>
Var: Current Type	Variable including the ID of the current type. This variable is needed by the Article Data window in order to highlight the current type.
Var: Select-Type	Variable including the ID of the type to be transferred to the PLC or User Interface.
Var: Article Data in PLC	Starts loading the Article Data from the User Interface to the PLC (see example). The following values may be defined: [Start Article Data in PLC];[Article Data written];[Fault]
Var: Article from PLC	Starts loading the Article Data from the PLC to the User Interface (see example). The following values may be defined: [Start Article Data from PLC];[Article Data read];[Fault]
Var: Activate Type	Variable that includes the ID of the type to be transferred to the PLC via the special key “Enter Desired Type before Key Action”.
Var: Current TypeName	This variable is used to send the type name of the current Type to the PLC.

**Example: PLC requests Article Data**

Preconditions:

- Variable “Var: Select Type“ exists.
- Variable “Var: Article Data in PLC“ is a bit-variable and includes the default values 1;0

Procedure:

- PLC: Write ID of the type into the variable “Var: Select Type“.
- PLC: Set the variable “Var: Article Data in PLC“ to “True”.
- User Interface: Determine data of the desired type.
- User Interface: Write data into the appropriate variables (Article Data of objects).
- User Interface: If data of this type has been found, the variable “Var: Article Data in PLC“ will be reset to “False“; otherwise it will remain set to “True“

**Example: PLC writes data into a Type**

Preconditions:

- Variable “Var: Select Type” exists.
- Variable “Var: Article Data from PLC” is a bit-variable and includes the default values 1;0

Procedure:

- PLC: Write data of the type into the appropriate variables (Article data of the objects).
- PLC: Write ID of the type into the variable “Var: Select Type”.
- PLC: Set the variable “Var: Article Data in PLC“ to “True”.
- User Interface: Copy data into the desired type.
- User Interface: If this type exists, the variable “Var: Article Data from PLC“ will be reset to “False“; otherwise it will remain set to “True“

**Example: Activating a Type from the User Interface**

Preconditions:

- Variable “Var: Select Type“ exists.
- Variable “Var: Article Data in PLC“ is a bit-variable and includes the default values 1;0.
- The variable “Var: Activate Type” exists.
- Special key “Enter desired Type before Key Action” is defined and linked to a bit-variable. The assumed default value is 1.

Procedure:

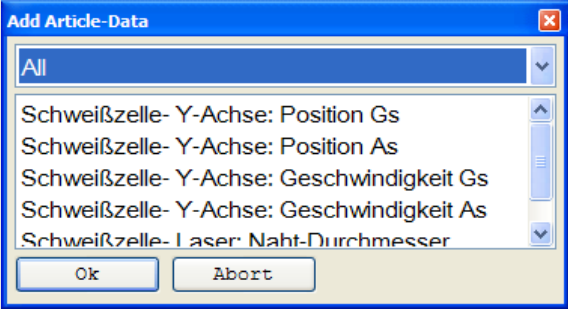
- User Interface.: Highlight the desired type and press the special key.

- User Interface: Prompt if the type has to be activated.
- User Interface: If the prompt is answered with “Yes”, the ID of the type will be written into the variable “Var: Activate Type”, and the bit-variable of the key will be set to 1.
- PLC: Reset the bit variable of the key to 0.
- PLC: Copy the contents of the variable “Var: Activate Type” into “Var: Select Type”.

Proceed as shown in our example “PLC requests Article Data”.

### Article Data

Allows defining the Article Data of the objects displayed in this window. Article data may be grouped in folders for enhanced clarity. The context menu is used to manipulate the list of Article Data to be displayed.

Menu Item	Description
Rename F2	Allows renaming folders
Add CTRL+INSERT	<p>This menu item is used to add Article Data to the window. A dialogue will appear, showing the list of Article Data to be added.</p>  <p>Using this list, one or several Article Data can be highlighted and copied. Only Article Data that are not yet in the window will be displayed (filter function).</p>
At the same level	This button defines if an entry to be added is at the same level as the active one, or if it has to be entered at a subordinated level.
Add Folder CTRL+O	<p>Adds a new folder, which is created at the same level as the highlighted entry.</p> <p>Besides the folder designation, you should also enter its type and associated entry mask for the Article Data. The following types are possible:</p> <ul style="list-style-type: none"> <li>• Display all sub-entries: If the folder is selected, the entries of this folder and of its sub-folders will be displayed.</li> </ul>

	<ul style="list-style-type: none"> <li>• Show only these entries: After selecting the folder only the entries in this folder will be shown.</li> <li>• Mask: Shows an input mask instead of an input table. This mask can be created under Objects (see page 62).</li> <li>• Open window: Allows opening another window for entering Article Data (especially special Delphi windows).</li> <li>• Invisible: Hides the folder. The Article Data in this folder remain stored!</li> </ul>
Add Sub-folder CTRL+P	Adds a new folder, which is appended below the highlighted folder.
Change folder	Allows changing the folder settings.
Delete CTRL+DEL	Deletes the highlighted entry.
Up CTRL+U	Moves the highlighted entry UP by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.
Down STRG+D	Moves the highlighted entry Down by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.



### Machine Data Window

The Machine Data Window allows manipulating the type-independent data of the machine. The window only allows displaying and manipulating PLC variables. The actual data must be held remanently in the PLC.

#### Special Keys

Input	Description
Change Value	After pressing this button the window for entering the currently active value will be opened. The same will happen after a double-click on the entry.
Save Values in File	If this button is pressed, all the Machine Data will be saved in an ASCII file.
Load Values from File	If this button is pressed, all the Machine Data will be read from an ASCII file.

#### Texts

The texts can be found in the General part. The following texts are required for the Machine Data window:

ID	Description
Title: Unit	Text of the title for the “Unit” column in the Machine Data table.
Title: Text	Text of the title for the “Text” column in the Machine Data table.
Title: Value	Text of the title for the “Value” column in the Machine Data table.

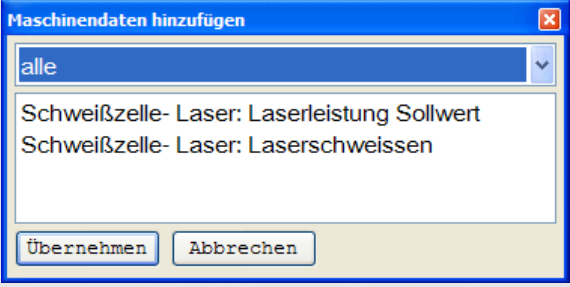
### Settings

Input	Description
Width Column Text [%]	Defines the width of the text column as a percentage of the available space.
Width Column Unit [Pixel]	Defines the width of the text column in pixels. The remaining width is used for an input value.
No Rights: Colour	Defines the text colour of input for which the user has no entry rights. (The required rights are set under Article Data).
No Rights: Action	<p>Allows setting a standard action in case that the user should not have any rights for entry. (The required rights are set under Article Data). The following settings are possible:</p> <ul style="list-style-type: none"> <li>• <b>Hide:</b> The entries will not be shown.</li> <li>• <b>Display:</b> The entries are displayed in the colour defined under “No Rights: Colour”, but cannot be edited.</li> <li>• <b>Display + Login:</b> The entries are displayed in the colour defined under “No Rights: Colour”. If the user should attempt to change the entry, he will first be prompted to authenticate and will not be allowed to change the value without the appropriate User rights.</li> </ul>
Var: Save in File	<p>String-Variables: As soon as a text is entered by the PLC, the User Interface will start saving the machine data into a file with the defined filename. The filename has to be saved together with its filename extension. The pathname references the path of Brosis-Commander-Client. The following values may be defined:</p> <p>[Returned text on successful completion];[Returned text in the event of an error]</p>
Var: Read from File	<p>String-Variables: As soon as a text is entered by the PLC, the User Interface will start loading the machine data from a file with the defined filename. The filename has to be saved together with its filename extension. The pathname references the path of Brosis-Commander-Client.</p> <p>The following values may be defined:</p> <p>Returned text on successful completion];[Returned text in the event of an error]</p>



### Machine Data

Allows defining the machine data of the objects to be displayed in this window. The machine data may be grouped together in folders for enhanced clarity. The list of machine data to be displayed can be manipulated by means of the context menu.

Menu item	Description
Rename F2	Allows renaming folders
Add CTRL+INSERT	<p>This menu item is used to add machine data to the window. A dialogue will be presented, showing a list of machine data to be added.</p>  <p>Using this list, one or several machine data entries can be highlighted and copied. Only machine data entries which are not yet in the window will be displayed (filter function).</p>
At the same level	This button defines if an entry to be added is at the same level as the active one, or if it has to be entered at a subordinated level.
Add Folder CTRL+O	Adds a new folder, which is created at the same level as the highlighted entry.
Add Sub-Folder CTRL+P	Adds a new folder. This folder is appended and appears below the highlighted folder.
Delete CTRL+DEL	Deletes the highlighted entry.
Up CTRL+U	Moves the highlighted entry UP by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.
Down STRG+D	Moves the highlighted entry Down by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.



## Measurement Data Window

This window allows displaying the measuring values of the objects in the form of a table.

### Texts

The texts are found in the General part. The following texts are required for the Measurement Data window:

ID	Description
Title: Unit	Text of the title for the Unit column in the Machine Data table.
Title: Measurement	Text of the title for the Measurement column in the table of Machine Data.
Title: Value	Text of the title for the Value column in the table of Machine Data.

### Settings

Input	Description
Width Column Text [%]	Defines the width of the text column as a percentage of the available space.
Width Column Unit [Pixel]	Defines the width of the text column in pixels. The remaining width is used for an input value.

### Measurement Data

Used to define the measurement data of the object to be displayed in this dialogue. The measurement data may be grouped together in folders for enhanced clarity. The list of measurement data to be displayed may be manipulated by means of the context menu.

Menu Item	Description
Rename F2	Allows renaming folders.
Add CTRL+INSERT	This menu item is used to add measurement data to the window. A dialogue will be presented, showing a list of measurement data to be added. One or several machine data entries may now be highlighted and copied by means of this list. Only machine data which are not yet included in the dialogue will be displayed.

At the same level	This button defines if an entry to be added is at the same level as the active one, or if it has to be entered at a subordinated level.
Add Folder CTRL+O	Adds a new folder, which is created at the same level as the highlighted entry.
Add Sub-Folder CTRL+P	Adds a new folder. This folder is appended and appears below the highlighted folder.
Delete CTRL+DEL	Deletes the highlighted entry.
Up CTRL+U	Moves the highlighted entry UP by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.
Down STRG+D	Moves the highlighted entry Down by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.



### Manual Function Windows

The Manual Function window is used to show the Manual Functions of the objects in the form of a table or graph (depending on the selection), and to manipulate these functions. When allocating the function keys it should be noted that the function-keys defined for the Manual Function objects have to be displayed.

#### Text

The text is given in the General Part. The following texts are required for the Manual Function window:

ID	Description
Title: Text	Text of the title for the Text column in the table. This column includes the names of the individual objects.
Title: TypeSelection	Text of the Window title for the Type selection (for graph elements requiring this prompt).
Title: Locked	Text of the title of the Locked column in the table. This column shows all the locked (latched) functions.
Title: Status	Text of the title for the Status column in the table. This column shows the current status of the object.

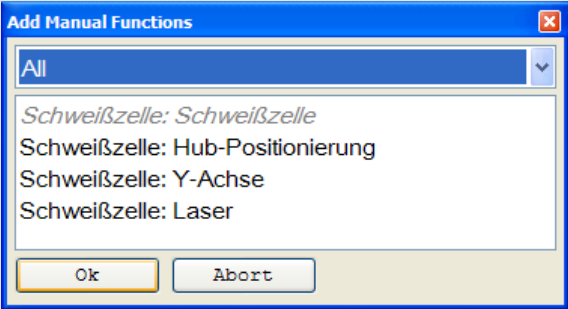
## Settings

Input	Description
Register All Variables	Defines if all required variables are to be registered at the programme start already (True). By activating this button, the changeover (toggle function) between the manual functions will become faster. If it is not activated, only the required variables will be registered. This saves memory space and is especially advisable for projects with a great number of objects.
Var: Current Type	This is the variable that includes the ID of the current type. This variable may be required by graph (image) elements of an object.
Var: Active Manual Function ID	The ID of the active manual function is written into this variable.
Key - Level	Indicates the level of the display of keys. The level determines which keys of the manual functions of the objects are to be displayed in this window. This allows, for example, creating a second Manual Function window and assigning a key level which is high enough to allow viewing only (but no operation).
Width – Status Column [Pixel]	Indicates the width of the Status column in pixels.
Width – Locked Column [Pixel]	Indicates the width of the Locked column in pixels. The remaining width available is used for the Object Names column.
Colour: Reached (Table)	Colour in which the Manual Function appears in the Table view after reaching the defined status. This setting is valid for all Manual Function windows.
Colour: Activated (Table)	Colour in which the Manual Function appears in the Table view after being activated. This setting is valid for all Manual Function windows.
Colour: Locked (Table)	Colour in which the Manual Function appears in the Table view after being locked. (This setting is valid for all Manual Function windows).
Colour: Reached (Key)	Colour in which the function key LED is shown after the Manual Function has reached the defined status. (This setting is valid for all Manual Function windows).
Colour: Activated (Key)	Colour in which the function key LED is shown after the Manual Function is activated.

	(This setting is valid for all Manual Function windows)!
Colour: Locked (Key)	Colour in which the Function Key LED is shown after the Manual Function is locked. (This setting is valid for all Manual Function windows).

### Manual Function Data

Allows defining the Manual Functions of the objects displayed in this window. For enhanced clarity, the Manual Function data may be grouped together in folders. The list of displayed Manual Function data can be manipulated by means of the context menu.

Menu Item	Description
Rename F2	Allows renaming objects
Add CTRL+INSERT	This menu item is used to add Manual Function data to the window. A dialogue will be presented, showing the list of Manual functions to be added. 
At the same level	This button defines if an entry to be added is at the same level as the active one, or if it has to be entered at a subordinated level.
Add Folder CTRL+O	Adds a new folder, which is created at the same level as the highlighted entry.
Add Sub-Folder CTRL+P	Adds a new folder. This folder is appended and appears below the highlighted folder.
Delete CTRL+DEL	Deletes the highlighted entry.
Up CTRL+U	Moves the highlighted entry UP by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.
Down STRG+D	Moves the highlighted entry DOWN by one position (within its level). If you wish to move the entry to another

level, you may drag it with your mouse.



## User Administration Window

If a User Administration window has been created, not only the password, but the User name and password will be used for User login. In this way, the registered User can be identified very clearly in the protocol. No further settings are required besides the function key entries for creating, changing or deleting a User. The User information is saved in the file "User.xml" in the server directory.

### Special Keys

Input	Description
New User	Opens a dialogue for creating a new user; this requires entering the name, password and User Right class.
Change User	Opens a dialogue for changing the User data.
Delete User	Deletes the highlighted User.

#### Note

The User password may also be changed in the login dialogue. In this case, the old as well as the new password must be entered.

### Text

The text is given in the General part. The following texts are required for the Manual Function windows:

ID	Description
Prompt: DeleteUser	Text of the prompt to be displayed before deleting a User. The name of the user may be inserted by means of the wildcard %s. Example: "Do you want to delete User "%s" ?"
Input: Password	Title for password entry.
Input: Name	Title for User name entry.
Input: Rights	Title for entering the User Right Class.
Input: Abort key	Designation of the Abort key in the input dialogue.
Input: OK key	Designation of the OK key in the input dialogue.
Message: User Already Exists	Message text displayed if a user is created whose name already exists.

Message: Different passwords	Message text displayed if the entered passwords are not identical.
Title: Name	Designation of the “Name” column in the User list.
Title: Rights	Designation of the “Rights” column in the User list.



### Browser Window

The Browser window allows displaying any desired files and web contents. This is convenient method of displaying such documents as, for example, circuit diagrams, pneumatic plans, operating instructions, etc., in a PDF format, Word format, etc. The respective programme (Acrobat Reader™, MS-Word™, ...) has to be installed on your computer. In this way, it is also possible to embed Intranet pages or web-cams without difficulty.

### Settings

Input	Description
Show Tree	Defines if a tree is to be displayed in addition to the browser, e.g. to select various other contents. This tree may be configured within the Browser data.
Var: Show Input	This variable allows the PLC to display an entry defined in the tree by simply returning the numerical ID of the entry.

### Browser Data

Allows defining data for the contents to be displayed. For enhanced clarity this information can be grouped together in sub-folders. The list of data to be displayed can be manipulated by means of the context menu.

Menu Item	Description
Add CTRL+INSERT	This menu item is used to add a new entry to the tree of Browser data.
Add Folder CTRL+O	Adds a new folder. This folder is created at the same level as the highlighted entry.
Add Sub-Folder CTRL+P	Adds a new folder, which is appended below the highlighted folder.
Delete CTRL+DEL	Deletes the highlighted entry.
Copy CTRL+C	Copies the highlighted entry to the clipboard.
Insert CTRL+V	Inserts an entry from the clipboard. If a folder is highlighted, the entry will be added to this folder.

Insert At the same Level CTRL+E	Inserts an entry from the clipboard. If a folder is highlighted, the entry will be added at the folder level.
Up CTRL+U	Moves the highlighted entry UP by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.
Down STRG+D	Moves the highlighted entry DOWN by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.

The following information may be entered for each input item:

Input	Description
Description	Name of entry
Address	Address or file to be displayed, e.g.: <ul style="list-style-type: none"> <li>• .\Help\Home.pdf</li> <li>• .\Infos\CircuitDiagram.pdf</li> <li>• <a href="http://www.brosis-engineering.de">www.brosis-engineering.de</a></li> <li>• .\Docu\Content.doc</li> </ul>
Show in Browser	If this value is True, data will be displayed as an integral part of the window; otherwise it will be opened externally, e.g. by means of Acrobat-Reader™ for PDF files.
ID	ID of the entry, enabling the PLC to display it via the variable “Var: Show Entry“.
Required Right	Defines the minimum User right required for a User to open an entry.
Width	Defines the width of the Browser area in pixels. If a 0 is entered here, the entire space available will be used.
Height	Defines the height of the Browser area in pixels. If a 0 is entered here, the entire space available will be used.
X-Position	Defines the X-position of the Browser area in pixels. The 0 position is at the far left.
Y-Position	Defines the Y-position of the Browser area in pixels. The 0 position is at the far left.

The definition of the height, width and position of the Browser area is necessary in order to mask out the scroll bars, for example. In addition, several web-cams do not always display their images at the upper left, but instead, use a wide margin. This can be compensated for by moving the X and Y-positions to the negative area.





## Protocol Windows

The Protocol Window allows displaying any desired ASCII text files separated by tabs. The Overview shows a list of existing files. The Detail area shows the file in the form of a table. The display can be restricted by means of the filter settings!

### Special Keys

Input	Description
Open Protocol	Opens the file highlighted in the shortlist. The same effect is obtained with a double-click.
Refresh Protocol List	Re-imports the list of existing protocol files.
Print	Prints the displayed protocol. The printer type can be selected previously.
Print (Standard Printer)	Prints the displayed protocol on the standard printer, or respectively, on the printer that was last used by the programme.

### Settings

Input	Description
Filename	Filter for the filenames to be displayed within the list. Besides „*“, this filename may also include the wildcard <DATE>.
Load Data Automatically	If this parameter is “True”, the protocol file will be loaded as soon as the file has been selected from the list.
Default-Directory	Directory in which the protocol files are stored.
Data from Server	If this parameter is True, data will be requested from the server. The directory will then reference the server directory position!
Force Unicode-Format	If this setting is True, the protocol files will be formatted and interpreted as UNICODE; otherwise the programme will attempt to determine the format automatically.
List / Date Format	Format in which the date of the protocol file is to be displayed in the Overview list. This function is only possible if the <DATE> exists in the file filter.
Filter +	Filters the displayed lines (rows). Only the lines corresponding to the filter settings will be displayed. In addition to text, the wildcard “*” may also be entered. Only the column dis-

	<p>played under “Filter Position” will be filtered.</p> <p>Example:          Contents of text file:</p> <table border="1"> <thead> <tr> <th>Date</th> <th>User</th> <th>Prog</th> <th>Text</th> </tr> </thead> <tbody> <tr> <td>28.06.2006 07:35:15</td> <td></td> <td>Server</td> <td>Server Start</td> </tr> <tr> <td>28.06.2006 07:37:33</td> <td></td> <td>Client</td> <td>Start Panel</td> </tr> <tr> <td>28.06.2006 07:37:56</td> <td>Developer</td> <td>Client</td> <td>Stop Panel</td> </tr> <tr> <td>28.06.2006 07:58:14</td> <td></td> <td>Client</td> <td>Start Panel</td> </tr> <tr> <td>28.06.2006 07:58:50</td> <td>Developer</td> <td>Client</td> <td>Stop Panel</td> </tr> </tbody> </table> <p>Filter Settings:          Only show rows including the column Prog “Server”!          (Instead of the server, it is also possible to define “v” for the Filter +):</p> <p><b>Filter +</b>            Server  <b>Filter-Position</b> 3</p> <p>Result:</p> <table border="1"> <thead> <tr> <th>Date</th> <th>User</th> <th>Prog</th> <th>Text</th> </tr> </thead> <tbody> <tr> <td>28.06.2006 07:35:15</td> <td></td> <td>Server</td> <td>Server Start</td> </tr> </tbody> </table>	Date	User	Prog	Text	28.06.2006 07:35:15		Server	Server Start	28.06.2006 07:37:33		Client	Start Panel	28.06.2006 07:37:56	Developer	Client	Stop Panel	28.06.2006 07:58:14		Client	Start Panel	28.06.2006 07:58:50	Developer	Client	Stop Panel	Date	User	Prog	Text	28.06.2006 07:35:15		Server	Server Start												
Date	User	Prog	Text																																										
28.06.2006 07:35:15		Server	Server Start																																										
28.06.2006 07:37:33		Client	Start Panel																																										
28.06.2006 07:37:56	Developer	Client	Stop Panel																																										
28.06.2006 07:58:14		Client	Start Panel																																										
28.06.2006 07:58:50	Developer	Client	Stop Panel																																										
Date	User	Prog	Text																																										
28.06.2006 07:35:15		Server	Server Start																																										
Filter -	<p>Filters the displayed lines (rows).          Only the rows that do NOT correspond to the filter settings will be displayed. In addition to text, the wildcard “*” may also be entered. Only the column displayed under “Filter Position” will be filtered.</p> <p>Example:          Content of the text file:</p> <table border="1"> <thead> <tr> <th>Date</th> <th>User</th> <th>Prog</th> <th>Text</th> </tr> </thead> <tbody> <tr> <td>28.06.2006 07:35:15</td> <td></td> <td>Server</td> <td>Server Start</td> </tr> <tr> <td>28.06.2006 07:37:33</td> <td></td> <td>Client</td> <td>Start Panel</td> </tr> <tr> <td>28.06.2006 07:37:56</td> <td>Developer</td> <td>Client</td> <td>Stop Panel</td> </tr> <tr> <td>28.06.2006 07:58:14</td> <td></td> <td>Client</td> <td>Start Panel</td> </tr> <tr> <td>28.06.2006 07:58:50</td> <td>Developer</td> <td>Client</td> <td>Stop Panel</td> </tr> </tbody> </table> <p>Filter Setting:          Only the rows NOT showing “Server” in the Prog Column are to be displayed! Instead of the server, it is also possible to define “v” for Filter -):</p> <p><b>Filter -</b>            Server  <b>Filter-Position</b> 3</p> <p>Result:</p> <table border="1"> <thead> <tr> <th>Date</th> <th>User</th> <th>Prog</th> <th>Text</th> </tr> </thead> <tbody> <tr> <td>28.06.2006 07:37:33</td> <td></td> <td>Client</td> <td>Start Panel</td> </tr> <tr> <td>28.06.2006 07:37:56</td> <td>Developer</td> <td>Client</td> <td>Stop Panel</td> </tr> <tr> <td>28.06.2006 07:58:14</td> <td></td> <td>Client</td> <td>Start Panel</td> </tr> <tr> <td>28.06.2006 07:58:50</td> <td>Developer</td> <td>Client</td> <td>Stop Panel</td> </tr> </tbody> </table>	Date	User	Prog	Text	28.06.2006 07:35:15		Server	Server Start	28.06.2006 07:37:33		Client	Start Panel	28.06.2006 07:37:56	Developer	Client	Stop Panel	28.06.2006 07:58:14		Client	Start Panel	28.06.2006 07:58:50	Developer	Client	Stop Panel	Date	User	Prog	Text	28.06.2006 07:37:33		Client	Start Panel	28.06.2006 07:37:56	Developer	Client	Stop Panel	28.06.2006 07:58:14		Client	Start Panel	28.06.2006 07:58:50	Developer	Client	Stop Panel
Date	User	Prog	Text																																										
28.06.2006 07:35:15		Server	Server Start																																										
28.06.2006 07:37:33		Client	Start Panel																																										
28.06.2006 07:37:56	Developer	Client	Stop Panel																																										
28.06.2006 07:58:14		Client	Start Panel																																										
28.06.2006 07:58:50	Developer	Client	Stop Panel																																										
Date	User	Prog	Text																																										
28.06.2006 07:37:33		Client	Start Panel																																										
28.06.2006 07:37:56	Developer	Client	Stop Panel																																										
28.06.2006 07:58:14		Client	Start Panel																																										
28.06.2006 07:58:50	Developer	Client	Stop Panel																																										
Filter-Position	Column to which the filter function is to be applied.																																												
Data from Row	Indicates the row from which the display of data is to be started. This is necessary if the file includes one or several																																												

	headers.
Width of List	Indicates the width of the overview list in pixels. If 0 is entered here, the standard value from the Skin definition will be used.
Grouped	Defines if the content of a text file has to be grouped. This is useful for statistical evaluations, in which the sum of identical entries, or summed-up durations are added up. Examples are given further below in this Section.

### Columns

Allows defining the format of the columns to be displayed. The list of displayed data can be manipulated by means of the context menu.

Menu Item	Description
Add CTRL+INSERT	This menu item is used to add a new column.
Delete CTRL+DEL	Deletes the highlighted entry.
Up CTRL+U	Moves the highlighted entry Up by one position.
Down CTRL+D	Moves the highlighted entry DOWN by one position.

The following settings are available for each column:

Input	Description
Title	Title of the column.
Position	Position of the column within the file, starting with 1.
Position (Duration “To“)	Position of the second column within the file. This column is needed to determine a duration. In this case, the first “Position” defines the start time; the second “Position” defines the end time of the duration.
Min. Width [Pixel]	Defines the minimum width of the column in pixels. If the column text requires more space, the width is adjusted automatically.
Print Width [mm]	Defines the width of the column (in mm) for the printout.
Data Type	Defines the type of data in a column. The following data types are possible: <ul style="list-style-type: none"> <li>• Digits: Data is expressed in integers or floating-comma digits. The “Decimal Places” field can be used to adapt the output format.</li> </ul>

	<ul style="list-style-type: none"> <li>• Date: Data is expressed in the form of date. The format is defined under “Date Format”.</li> <li>• Text: Data is expressed in the form of a text field.</li> <li>• Duration: Data is expressed in the form of a duration.</li> </ul>
Alignment	Defines the alignment of the field (left, right, or centred).
Decimal Places	Defines the number of decimal places for numerical entries.
Date Format	Defines the format of date fields, e.g. dd.mm.yyyy hh:nn:ss:zzz
Date Format of Entry	Defines the date format of an entry within the file, e.g. yy-mm-dd hh/nn/ss
Sorted	Defines in which order data is to be sorted after loading. This entry is only used for <u>one</u> column each!
Group: Duration	Sums up the duration of rows with identical fields (Grouped Display).
Group: Number	Returns the number of rows with identical fields (at Grouped Display). In this case the “Position” field is not relevant.
Group: Bar Graph	Provides the column with a bar graph indicating the percentage as compared to the total value and/or total duration.
Group: Bar Colour	Allows setting the colour of the bar graph.
Group: Display in Percent	Does not display the duration (or number) as a unit, but as a percentage of the total number or total duration.

### Print and Print Preview

The Print area may be used to enter the general print settings. These settings can be activated under Print Preview.

Input	Description
Title	<p>Text of the Print title; this text will only appear on the first page. The title can include the following wildcards:</p> <p><b>&lt;PAGE&gt;</b>: Replaced with the number of the current page.</p> <p><b>&lt;#PAGES&gt;</b>: Replaced with the number of pages.</p> <p><b>&lt;SELECT_TITLE&gt;</b>: Replaced with the displayed text of the selected file.</p> <p><b>&lt;SELECT_FILE&gt;</b>: Replaced with the filename of the selected file.</p>

	<p><b>&lt;DATE:format&gt;</b>: Replaced with the current date.</p> <p><b>format</b> may be used to define the format of the date output. Possible formats are: :</p> <p><b>dd</b> Day as a two-digit value  <b>ddd</b> Day as an abbreviation, Mon, Tue, ...  <b>mm</b> Month as a two-digit value  <b>mmm</b> Month as an abbreviation, Jan, Feb, ...  <b>yy</b> Year as a two-digit value  <b>yyyy</b> Year as four-digit value  <b>hh</b> Hour as a two-digit value  <b>nn</b> Minute as a two-digit value  <b>ss</b> Second as a two-digit value</p> <p>Example: <i>&lt;Date:dd.mm.yyyy hh:nn &gt;</i></p>
Header	Defines the header text. This text is displayed in the header of each page. It may include the same wildcards as the title.
Footer	Defines the footer text. This text appears left-justified in the footer of each page. The text may include the same wildcards as the title.
Portrait Format	Defines if the Portrait format or Landscape format is selected for the printout.
Left Margin [mm]	Defines the left margin in millimetres.
Right Margin [mm]	Defines the right margin in millimetres.
Top Margin [mm]	Defines the top margin in millimetres.
Bottom Margin [mm]	Defines the bottom margin in millimetres.
Title: X-Pos [mm]	Defines the title position in the X-direction.
Title: Y-Pos [mm]	Defines the title position in the Y-direction.
Table: X-Pos [mm]	Defines the table direction in the X-direction.
Table: Y-Pos [mm]	Defines the table direction in the Y-direction.
Horizontal Lines	Defines if horizontal lines are to be inserted between the rows (a line is drawn under the column titles in any case).
Vertical Lines	Defines if vertical lines are to be drawn between the columns.
Font: Title	Allows selecting the title font.
Font: Table	Allows selecting the table font.
Font: Header	Allows selecting the header font.
Font: Footer	Allows selecting the footer font.
Logo: X-Pos [mm]	Defines the position of the logo in the X-direction.
Logo: Y-Pos [mm]	Defines the position of the logo in the Y-direction.

Logo: Width [mm]	Defines the width of the logo. The height is calculated automatically based on the original page ratio. This ensures that the image is not distorted.
Logo	Allows embedding an image in the printout.

**Examples**

A) *You wish to create a statistics on the total duration of individual faults from the Fault Protocol:*

Column Definitions:

Input	Column 1	Column 2
Title	Fault	Duration
Position	4	1
Position (Duration "To")	0	2
Data Type	Text	Duration
Date Format		dd.mm.yyyy hh:nn:ss
Date Format of the Entry		dd.mm.yyyy hh:nn:ss
Sorted	None	Descending
Group: Duration	False	True
Group: Number	False	False
Group: Bar graph	False	False
Group: Show in Percent	False	False

B) *Now you wish to create a statistics on the number of individual faults from the Fault Protocol. In addition to the numerical value, you also wish to display a bar graph with % indication.*

Column Definitions

Input	Column 1	Column 2	Column 3
Title	Fault	Number	Percentage
Position	4	0	0
Data Type	Text	Number	Number
Sorted	None	Descending	None
Group: Duration	False	False	False
Group: Number	False	True	True
Group: Show Bar Graph	False	False	True
Group: Bar Graph Colour	□ 16777215	□ 16777215	■ 255
Group: Show in Percent	False	False	True



## Step Function Window

The Step Function window is used to activate, deactivate and monitor sequential functions (step functions) or alternative processes during the startup. The Stepper functions can be displayed in their own window, so that these processes can be viewed and monitored during normal operation.

### Text

The texts are given in the General part. The following text entries are required for the Stepper Window:

ID	Description
Title: Chain	Defines the title text of the column that includes the chain descriptions.
Title: Step	Defines the title text of the column that includes the text of the current step.

### Settings

Input	Description
Display Own Window	Defines if the window is to be displayed as a separate window. In this case it will continuously appear in the foreground. With a double-click on the title bar it may be reduced to the height of the title bar.
Width: Chain Column [Pixel]	Defines the width of the Chain column in pixels. The remaining width is used for the Step column.

### Stepper Data

Used to define data for the step chains to be displayed. For enhanced clarity, the data may be grouped in folders. The list of displayed data may be manipulated by means of the context menu.

Menu Item	Description
New CTRL+INSERT	This menu item is used to add a new entry to the list.
Delete CTRL+DEL	Deletes the highlighted entry.
Copy CTRL+C	Copies the highlighted entry to the clipboard.
Insert	Inserts an entry from the clipboard.

CTRL+V	
Up CTRL+U	Moves the highlighted entry UP by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.
Down CTRL+D	Moves the highlighted entry DOWN by one position (within its level). If you wish to move the entry to another level, you may drag it with your mouse.

The following information can be entered for each data input:

Input	Description
Name	Name of entry
Var: Chain Active	Variable for activating or deactivating the entry.
Var: Current Step	Variable showing the current status of the entry.



### Special Window (Delphi)

This dialogue is used to define a special window. Special windows have to be created with Delphi (see: Own Delphi Programme).

#### Settings

Input	Description
Type Overview	Defines what has to be displayed in the Overview area. the following options are available: <ul style="list-style-type: none"> <li>• <b>No Margin:</b> The overview is masked out, leaving more space for the image.</li> <li>• <b>Display values:</b> Allows displaying Measuring and Display values from the objects of the station.</li> <li>• <b>Display elements:</b> Allows placing freely definable display elements in the Overview area.</li> <li>• <b>ID (Delphi):</b> If this item is selected, the Overview area will be defined by a customised Delphi application.</li> </ul>
Display Values	Defines the display elements to be shown in the Overview area.
Display Elements	Allows positioning display elements within the fixed margin of the User Interface (see <b>Display Elements</b> ).

#### Parameter

Used to send parameters to the Delphi window in the format:

<ID>=<Value>



## CHAPTER 4: WINDOW DATA

in order to adapt the settings of the special windows via the parameterisation.

## Display Elements

*Display elements may be designed freely and placed at the outer areas or Detail areas of Station windows.*

**D**isplay elements allow you to freely design the User Interface. They can be created in the header or footer area, or within the fixed margin areas by means of the Project information data. The display of elements in the Overview area is only possible for the Station windows.

In addition to these options, display elements may also be added to each graphical element or image. This allows operating the objects under Manual Functions. If Display elements are added to the image of a station, these elements will appear in the appropriate Station window.










### The Image Area Editor

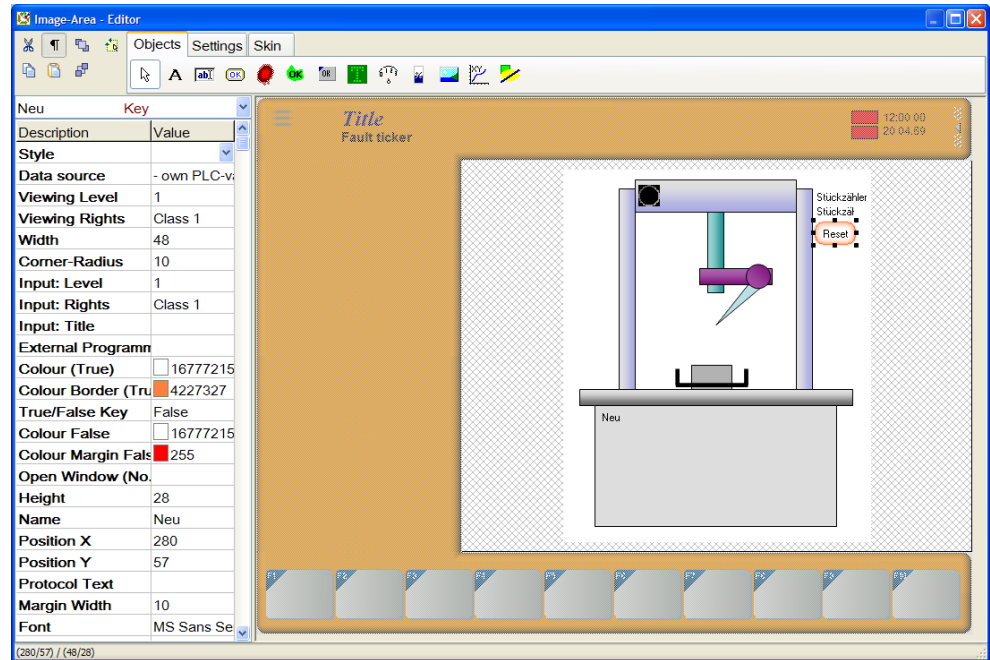
The Image Area Editor allows positioning and parameterising the individual Display elements in various areas. The currently inactive areas are shaded in grey.

To **insert** a Display element in the image area, first select the desired object and then click on the position where it is to be inserted. The object data will be shown in the Object Editor for changing or editing.

To **delete** a Display element, highlight it and then press the **–** key.

Element	Description
	Deletes the highlighted elements.
	Shows or hides the names of key areas.
	Copies the highlighted object to the clipboard.
	Copies an object from the clipboard to the image area.
	Displays the highlighted elements in the foreground.
	Displays the highlighted elements in the background.
	Highlights all elements.
Objects	Includes all available objects.
Settings	Allows you to define if the objects are to be aligned to a grid,

	and defines the grid size.
	By means of “Highlight Colour” you may change the colour of the currently highlighted (active) object.
Skin	Allows you to select the Skin to be used.



## The Object Editor

The Object Editor displays the data of the highlighted object and allows changing the settings. The behaviour of the objects can be controlled by these settings.

Neu		Key
Description	Value	
<b>Style</b>		
Data source	- own PLC-v	
Viewing Level	1	
Viewing Rights	Class 1	
Width	48	
Corner-Radius	10	
Input: Level	1	

A shortlist including the names of all existing objects is shown at the top.

This list can also be used to highlight an object.

For integer variables, it is possible to increase or reduce a value by means of the key combination CTRL+A and CTRL+S.

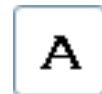


## The Image Objects in Detail

Below is a description of existing image objects and their features and functions. However, let us start with some comments concerning the general features of the objects:

- The name of an object is given for easier identification only. It does not have to be unique and has no further effects.
- The width, height, and X/Y-positions are indicated in pixels.
- Within images, the X/Y-position refers to the upper left corner of the graph. When an image is displayed, it will appear in the centre of the detail area! In all other areas the positions are referenced to the upper left corner of the respective area.

## Text Objects



Text Objects allow displaying language-independent texts in the Display area.

Properties	Description
Alignment Horiz.	Defines the horizontal alignment of the text within its area (left-justified, right-justified, or centred).
Alignment Vert.	Defines the vertical alignment of the text within its area (top margin, bottom margin, or centred).
Width	Defines the admissible width of a text.
Background Colour	Defines the background colour of the text field. The background colour will only be used, if the property "Transparent" is set to False.
Height	Defines the height of the text field.
Name	Defines the name of the object.
Position X	Defines the X-position of the text field.
Position Y	Defines the Y-position of the text field.
Border	Defines whether or not a border has to be displayed.
Border Width	Defines the width of the border in pixels.
Border Colour	Defines the colour of the border.
Font type	Selects the font type of the displayed text.

Text	Defines the displayed text (in the language currently selected). The following wildcards may be used: <ul style="list-style-type: none"> <li>• <b>{OBJECT}</b> for the object name and <b>{STATION}</b> for the station name.</li> </ul>
Transparent	Defines whether or not the background shall be visible.
Var: Background Colour	This PLC variable allows changing the background colour of the text field at runtime. The background colour will only be displayed, if the property “Transparent” is set to False.
Var: Text Colour	This PLC variable allows changing the text colour at runtime.
Angle	Angle (in degrees) of the text output.

### Input/Display Object



Input/Display Objects are used to display text and values from the PLC. The displayed values may be changed and transferred to the PLC.

Property	Description
Alignment Horiz.	Defines the horizontal alignment of the text within its area (left-justified, right-justified, or centred).
Alignment Vert.	Defines the vertical alignment of the text within its area (top margin, bottom margin, or centred).
Data Source	For Image elements only: Defines if the displayed PLC variable is obtained from the variable defined under “Var: Input Value”, or whether this is a variable from the Manual Functions of the object.
Article Relations	For Image elements only: This entry allows linking the input field to Article variables. This makes it possible to create masks for entering Article Data.
Width	Defines the permissible width of a text.
Input: Level	For Image elements only: Defines the level of an entry. To allow an entry, this level has to be equal or greater than the window level (see Station <i>Windows</i> ).
Input: Rights	Defines the minimum right a User must be assigned to change the contents of the variable.
Input: Title	Title of the input window opened for changing a value.
Background Colour	Background colour of the input field. The background colour will only be only used, if the property “Transparent” is sent to False.
Height	Defines the height of the input field.

Name	Defines the name of the object.
Read Only	Defines if the content of a variable has to be displayed only, or may also be changed.
Position X	X-position of the input field.
Position Y	Y-position of the input field.
Protocol Text	If a value is entered here, the protocol text and the entered value will be included in the protocol. The following wild-cards may be used: <ul style="list-style-type: none"> <li>• <b>{OBJECT}</b> for the object name and <b>{STATION}</b> for the station name.</li> </ul>
Border	Defines whether or not a border has to be displayed.
Border Width	Defines the width of the border in pixels.
Font type	Defines the font type for the display of the text.
Transparent	Defines whether or not the background colour shall be visible.
Var: Input Value	Defines the PLC variable the value of which has to be displayed and/or changed.
Var: Background Colour	This PLC variable allows changing the background colour of the text field at runtime. The background colour will only be displayed, if the property “Transparent” is set to False.
Var: Text Colour	This PLC variable allows changing the text colour at runtime.
Value range	Defines the value range for an entry.
Angle	Angle (in degrees) of the text output.

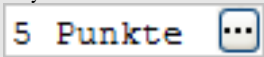
### Key Area Object



The Key Area object function allows any desired number of polygons (max. 20 points) to be defined in an image. These areas are treated as keys, i.e. action is initiated by a mouse-click.

Unlike other objects, key objects cannot be moved by means of Drag and Drop – they have to be redrawn.

Property	Description
Width	Defines the width of the object.
Data Source	For Image elements only: Defines if the key press has to operate the PLC variable from the property “Var: Key Pressed” and “Var: Key Released”, or if the pressing of a manual function key has to be simulated.
Input: Level	For Image elements only: Level of an entry. To allow an en-

	try, this level has to be equal or greater than the window level (see Station Windows).
Input: Rights	Defines the minimum right a User must be assigned to change the contents of the variable.
Input: Title	Title of the input window opened for changing a value.
Open Window (No.)	Defines the number of the window to be opened after pressing the key.
Height	Defines the height of the object.
Name	Defines the name of the object.
Protocol Text	If a value is entered here, the protocol text and the entered value will be included in the protocol. The wildcard <b>{OBJECT}</b> may be used for the object name.
Position X	X-position of the object.
Position Y	Y-position of the object.
Region	Indicates how many supporting points the region has at the moment. The region may be newly defined by pressing the key  To define the region, please proceed as follows: <ul style="list-style-type: none"> <li>• Press the "...“ key.</li> <li>• In the image, now click the desired points consecutively with the left mouse-button.</li> <li>• After selecting all the points, terminate your entry by clicking the right mouse-button.</li> </ul>
Special ID	Special ID. Allows executing special IDs (Virtual Keyboard and Print).
Var: Input Value	Defines the PLC variable the value of which has to be changed.
Var: Key Pressed	Defines the PLC variable to be set to the default value after pressing this key.
Var: Key Released	Defines the PLC variable to be set to the default value after releasing this key.
Value Range	Allows defining the value range for an entry.

### Key Object



A Key Object is used to display a key. The events of the key may be transmitted to the PLC.



Property	Description
Style	This property selects predefined styles.
Data Source	For Image elements only: Defines if the key press has to activate the PLC variable from the property “Var: Key Pressed” and “Var: Key Released”, or if the pressing of a manual function key has to be simulated.
Display: Level	To display the element, the level has to be equal to or greater than the window level (see Station Windows).
Display: Rights	Defines the minimum right the User must have to display the element.
Width	Defines the width of the key.
Corner Radius	Defines the corner radius of the key (in pixels)
Input: Level	For Image elements only: Level of an entry. To allow an entry, this level has to be equal or greater than the window level (see Station Windows).
Input: Rights	Defines the minimum right a User must be assigned to change the contents of the variable.
Input: Title	Title of the input window opened for changing a value.
External Programme	Allows accessing an external programme. Enter the programme including the path. If the programme should await access parameters, enter these in quotes after the path, e.g.: C:\Windows\Notepad.exe “c:\temp\Hallo.txt“
Colour (True)	Colour of the key area. If this is a “True/False Taste“ key, it defines which colour the key area has to assume if the PLC variable “Var: Colour” corresponds to the default value.
Colour Border (True)	Colour of the key border. If this is a “True/False Taste“ key, it defines which colour the key border has to assume if the PLC variable “Var: Colour” corresponds to the default value.
True/False key	If this field is set to True, the colour of the keys will not be determined by the colour value of the PLC variable “Var: Colour”, but by the properties “Colour (True)“, “Colour Border (True)“, or respectively, “Colour False” and “Colour Border False”.
Colour Border False	Defines which colour the key area has to assume, if the PLC variable “Var: Colour” <u>does not</u> correspond to the default value.
Colour False	Defines which colour the border of the key has to assume, if the PLC variable “Var: Colour” <u>does not</u> correspond to the default value.



Open Window (No.)	Defines the number of the window to be opened after pressing the key.
Height	Defines the height of the key.
Name	Defines the name of the key.
Position X	Defines the X-position of the key.
Position Y	Defines the Y-position of the key.
Protocol Text	If a value is entered here, the protocol text and the entered value will be included in the protocol. The wildcard <b>{OBJECT}</b> may be entered for the object name.
Border width	Defines the width of the coloured border of the key.
Font	Defines the font type for the key designation.
Special ID	Special ID of the key. Allows executing the standard special IDs (Virtual Keyboard and Print).
Text	Defines the designation of the key.
Text Shade	Defines if the text has to be shaded.
Var: Input Value	Defines the PLC variable the value of which has to be changed.
Var: Colour	This PLC variable allows changing the colour of the key at runtime.
Var: Border Colour	This PLC variable allows changing the border colour of the key at runtime.
Var: Key Pressed	Defines the PLC variable to be set to the default value after pressing this key.
Var: Key Released	Defines the PLC variable to be set to the default value after releasing this key.
Var: Text	This PLC variable allows changing the key designation at runtime.
Value Range	Allows defining the value range for an entry.

### Skin Key Object



A Skin Key object allows displaying a key in the form of a function key. The height and width of this key cannot be changed, whereas the function of the key can be sent directly to the PLC.



Property	Description
Data Source	For Image elements only: Defines if the key press has to activate the PLC variable from the property “Var: Key Pressed” and “Var: Key Released”, or if the keypress of a manual function key has to be simulated.
Input: Level	For Image elements only: Level of an entry. To allow an entry, this level has to be equal or greater than the window level (see Station Windows).
Input: Rights	Defines the minimum right a User must be assigned to change the contents of the variable.
Image	Icon (symbol) to be shown on the key.
Input: Level	For Image elements only: Level of an entry. To allow an entry, this level has to be equal or greater than the window level (see Station Windows).
Input: Rights	Defines the minimum right a User must be assigned to change the content of the variable.
Input: Title	Title of the input window opened for changing a value.
External Programme	Allows accessing an external programme. Use this button to enter the programme including the path name. If the programme should prompt you to enter access parameters, enter these parameters in quotes, e.g.: C:\Windows\notepad.exe „c:\temp\Hallo.txt“
Open Window (No.)	Defines the number of the window to be opened on pressing the key.
Name	Defines the name of the object.
Position X	Defines the X-Position of the key.
Position Y	Defines the Y-Position of the key.
Protocol Text	If a value is entered here, the protocol text and the entered value will be included in the protocol. The wildcard <b>{OBJECT}</b> may be used for the object name.
Special ID	Special ID. Allows executing special IDs (Virtual Keyboard and Print).
Title	Defines the text in the title area of the key. This is the area that would otherwise show the function key number (F1...F12)
Var: Input Value	Defines the PLC variable the value of which has to be changed.
Var: Key Pressed	Defines the PLC variable to be set to the default value after pressing this key.

Var: Key Released	Defines the PLC variable to be set to the default value after releasing this key.
Var: Row 1	PLC variable defining the first text line (row) of the key.
Var: Row 2	PLC variable defining the second text line (row) of the key.
Value Range	Allows defining the value range for an entry.
Row 1	Designation of the first text line of the key.
Row 2	Designation of the second text line of the key.

### LED Object



An LED object is used to display an LED.



Property	Description
Width	Width of the LED.
Colour LED Off	If the colour value defined here is returned from the PLC to the LED, the LED will be switched Off.
Colour LED-True	This colour is shown if the PLC variable corresponds to the default value. See Property "Type True/False".
Colour LED-False	This colour is shown if the PLC variable does NOT correspond to the default value. See Property "Type True/False".
Shape	Defines the shape of the LED. Possible options are: Round, Rectangular, Rhomboid, or Semi-Circular.
Height	Defines the height of the LED.
Name	Defines the name of the object.
Illuminated Area	Defines whether or not an illuminated area has to appear around the LED. This setting may also be used to increase the intensity of the colour display.
Size of Illuminated Area	Defines the size of the illuminated area. Permissible values are between 0-10.
Intensity of Illuminated Area	Defines the intensity of the displayed illuminated area. Permissible values are between 1 – 5.
Position X	Defines the X-position of the LED.
Position Y	Defines the Y-position of the LED.
Type - True/False	If this variable is set to True, the Type True/False will be used, i.e. the LED will be set to the Property "Colour LED-True" (variable corresponds to default). In all other cases, the LED will be set to "Colour LED False".

	An AND-link may be entered in the PLC variable, e.g. “AND 2” – this effects that the LED assumes “Colour LED-True”, if: ( $\langle \text{Value} \rangle \text{ AND } 2 \rangle \langle \rangle 0!$
Var: Colour	This PLC allows changing the LED colour at runtime.

### Image Object



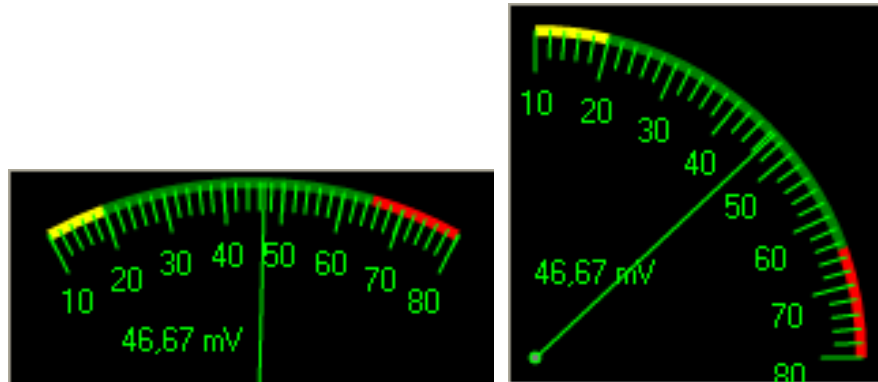
An image object is used to display any desired image via the PLC. This image is previously defined by means of the filename returned by the PLC.

Property	Description
Auto-Size	If True is entered here, the size of the object will adapt to the size of the image. Otherwise the image will be adjusted to the defined height and width.
Width	Width of the image.
Height	Height of the image.
Name	Name of the object.
Stretch	Defines if the image has to be stretched to the defined size, or if it has to be reduced.
Transparent	Defines a transparent display of the image. The transparent colour is that of the lower left pixel of the image.
Var: Filename	PLC variable used to return the filename of the image at runtime.
Var: X-Pos	PLC variable used to change the X-position of the image at runtime.
Var: Y-Pos	PLC variable used to change the Y-position of the image at runtime.
Var: Width	PLC variable used to change the width of the image at runtime. This is only advisable if Stretch True and Autosize False are set.
Var: Height	PLC variable used to change the height of the image during the runtime. This is only advisable if Stretch True and Autosize False are set.
Directory	Directory in which the image is stored. Using “.\BildListe“ the file is searched by the Client in the ImageList directory.
Default Image File	Defines the filename of the image, if no filename is defined by the PLC.
X-Position	Defines the X-position of the image.
Y-Position	Defines the Y-position of the image.

## Analogue Display Object



Using the Analogue Display Object, a numerical value from the PLC can be displayed on an analogue device. The scale of the analogue display can be subdivided into three areas, which can be shown in different colours:



Property	Description
View: Display	Allows selecting predefined display settings.
View: Colour Scheme	Allows selecting predefined colour schemes.
Anti-Aliasing	This setting is used to define how the object is to be drawn. If Anti-Aliasing is selected, diagonal lines will not be “broken”, but gradually shaded off.
Width	Defines the width of the object.
Region: Max Colour	Defines the colour of the max. region (range).
Region: Max Visible	Defines if the max. region has to be visible.
Region: Max Value	Defines the value that separates the max. region from the medium region.
Region: Min Colour	Defines the colour of the min. region.
Region: Min Visible	Defines if the min. region has to be visible.
Region: Min Value	Defines the value separating the max. region from the min. region.
Region: Centre Colour	Defines the display colour of the Centre region (range).
Region: Centre Visible	Defines if the Centre region has to be displayed.
Unit: Colour	Defines the display colour of the unit.

**CHAPTER 5: DISPLAY ELEMENTS**

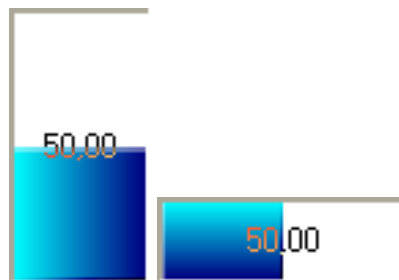
Unit: Font Size	Defines the font size in which the unit is displayed.
Unit: Visible	Defines if the unit has to be displayed.
Unit: Text	Defines the text of the unit.
Unit: X-Position	Defines the X-position of the unit.
Unit: Y-Position	Defines the Y-position of the unit.
Background Colour	Defines the background colour.
Height	Defines the height of the object.
Name	Defines the name of the object.
Position X	Defines the X-position of the object.
Position Y	Defines the Y-position of the object.
Font	Defines the font to be used for the scale, value and unit.
Scale: Main Tick Count	Indicates the number of main ticks on the scale.
Scale: Sub-Tick Count	Indicates the number of sub-ticks between main ticks on the scale.
Scale: End-Angle	Indicates the angle at which the scale has to end.
Scale: Main Ticks Visible	Indicates if main ticks are to be displayed.
Scale: Main Ticks Length	Defines the length of main ticks in pixels.
Scale: Post Decimal Places	Defines the number of post decimal places to be displayed on the scale text.
Scale: Start-Angle	Defines the angle at which the scale has to start.
Scale: Tick Colour	Defines the colour of the ticks on the scale.
Scale: Sub-Ticks Visible	Defines if the sub-ticks are to be displayed.
Scale: Sub-Tick Length	Defines the length of the sub-ticks in pixels.
Scale: Text Colour	Defines the colour of the scale text.
Scale: Text Visible	Defines if the scale text has to be displayed.
Border 3D	Indicates if the border has to be drawn with a 3D effect.
Border Visible	Defines if a border has to be displayed.
Var: Background Colour	PLC variable used to change the background colour at runtime.
Var: Max Region Value	PLC variable used to change the border between the max. and centre region at runtime.
Var: Max Value	PLC variable used to change the max. limit value at runtime.
Var: Min Region Value	PLC variable used to change the border between the lower and centre region at runtime.
Var: Min Value	PLC variable used to change the lower limit value at runtime.
Var: Value	PLC variable that includes the value to be displayed.

Value: Decimal Places	Defines the number of post-decimal places to be displayed for the value.
Value: Max-Value	Defines the max. limit value to be displayed.
Value: Min-Value	Defines the min. limit value to be displayed.
Value: Visible	Defines if the value has to be displayed as text (for a unit).
Pointer: Width	Defines the pointer width in pixels.
Pointer: Colour	Defines the pointer colour.
Pointer: Radius	Defines the pointer radius.
Pointer: X-Position	Defines the X-position for the start of the pointer.
Pointer: Y-Position	Defines the Y-position for the start of the pointer.
Pointer: Centre Colour	Defines the colour of the pointer centre.
Pointer: Centre Radius	Defines the radius of the pointer centre.
Pointer: Centre Visible	Defines whether or not the centre of the pointer has to be displayed.

### Progress Bar - Object



The Progress Bar is used to display the PLC value in the form of a bar graph.



Property	Description
Width	Defines the width of the object.
Gradient Start Colour	Defines the colour for the start of the colour range.
Gradient End Colour	Defines the colour for the end of the colour range.
Background Colour	Defines the background colour of the free area.
Horizontal	Defines horizontal or vertical display of the progress bar.
Height	Defines the height of the object.

Max-Value	Defines the upper limit value.
Min-Value	Defines the lower limit value.
Name	Defines the name of the object.
Position X	Defines the X-position of the graph.
Position Y	Defines the Y-position of the graph.
Border	Defines if a border has to be drawn (recessed, embossed, or no border).
Font	Defines the font in which the text is to be displayed.
Text	Defines the format of the displayed text. A value can be inserted in the text as follows: <b>%w.&lt;Post-Decimal Places&gt;f</b> returns a value <b>%p.&lt;Post-Decimal Places&gt;f</b> returns a value in percent. Example: %w.2f
Var: Colour to	PLC variable used to change the end of the colour gradient at runtime.
Var: Colour from	PLC variable used to change the start of the colour gradient at runtime.
Var: Max-Value	PLC variable used to change the upper limit at runtime.
Var: Min-Value	PLC variable used to change the lower limit at runtime.
Var: Text	PLC variable used to change the text at runtime.
Var: Value	PLC variable containing the text to be displayed.

### LCD Display Object



The LCD Display Object allows showing text on an LCD display.



Property	Description
Width	Defines the width of the object.
Height	Defines the height of the object.
Name	Defines the name of the object.
Position X	Defines the X-position of the image.
Position Y	Defines the Y-position of the image.
Border: Width	Defines the width of the border.

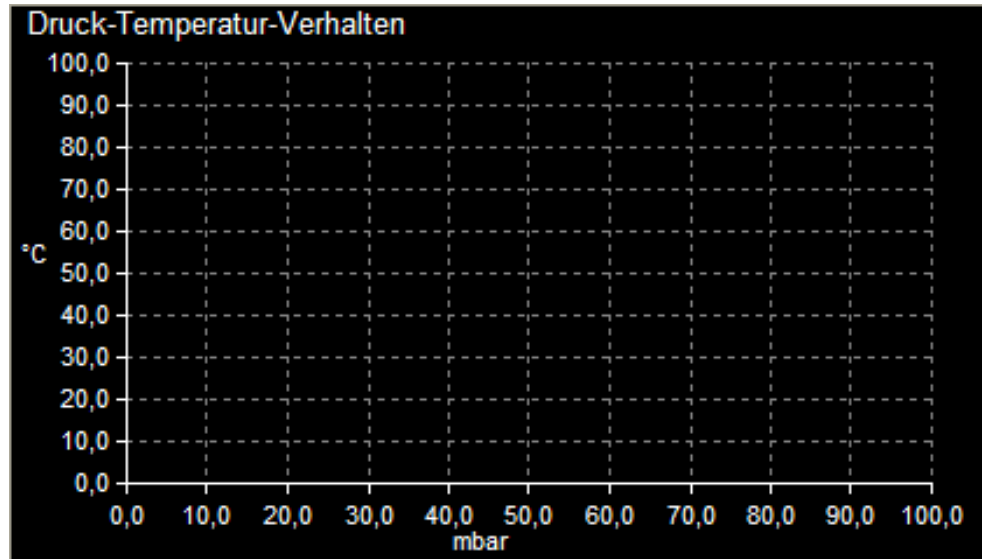


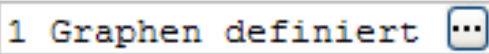
Border: Colour	Defines the colour of the border.
Border: Type	Defines whether and how the border has to be drawn (recessed, embossed, or no border).
Grid: Distance	Defines the distance of points between each other.
Grid: Colour Distance	Defines the background colour between points.
Grid: Colour OFF	Defines the colour of inactive points.
Grid: Colour ON	Defines the colour of active points.
Grid: Pixel Width	Defines the width of a point.
Grid: Pixel Height	Defines the height of a point.
Grid: X-Offset	Defines the X-value for the zero offset at which the text has to start.
Grid: Y-Offset	Defines the Y-value of the zero offset at which the text has to start.
Font Type	Defines the font type for the display of the text.
Text	Defines the text to be displayed. To start a new line, the wildcard <b>&lt;NL&gt;</b> may be used in the text.
Var: Colour OFF	PLC variable used to change the colour of inactive points at runtime.
Var: Colour ON	PLC variable used to change the colour of inactive points at runtime.
Var: Colour Border	PLC variable used to change the colour of the border at runtime.
Var: Text	PLC variable that includes the text to be displayed. To start a new line, the wildcard <b>&lt;NL&gt;</b> may be used in the text.
Line space	Defines the line space with reference to the default line space of the font. By entering negative values the space between lines may be reduced; by entering positive values the space may be increased. If 0 is selected, the default space of the font type will be used.

### XY-Plotter Object






The XY Plotter object allows displaying graphs. Several graphs may be displayed at the same time.



Property	Description
Width	Defines the width of the object.
Graphs	Indicates how many graphs are displayed at the moment. The Graph Editor can be opened by pressing the key: 
Background Colour	Defines the background colour of the XY plotter.
Height	Defines the height of the object.
Name	Defines the name of the object.
Position X	Defines the X-position of the image.
Position Y	Defines the Y-position of the image.
Show Border	Defines if a border has to be shown.
Border 3D	Defines if the border has to be drawn in 3D.

### Operating the Graph Editor

By using the  key in the Graph Editor, a new graph can be added; by using the  key the current graph is deleted, and by using the  key the Editor can be left. The graph to be edited may be selected from the shortlist.

Property	Description
Line Width	Defines the width of the line to be used for drawing the graph.
Memory-File	Defines the name of the file in which the points are saved.

Save Points	Defines if the points are to be saved.
Border: Left	Defines the width of the left border (margin).
Border: Top	Defines the width of the top margin.
Border: Right	Defines the width of the right border.
Border: Bottom	Defines the width of the bottom margin.
Grid X: Display	Defines if the X-grid has to be shown.
Grid X: Colour	Defines the colour of the X-grid.
Grid Y: Display	Defines if the Y-grid has to be shown.
Grid Y: Colour	Defines the colour of the Y-grid.
Ring buffer	Defines if data has to be transferred to a ring buffer by the PLC, or simply read out from the start of the array.
Font	Defines the font type used for the graph text.
Script	<p>Defines the script the data of which are to be displayed. Separated by a semicolon, several objects can be drawn in the graph. The following objects are possible:</p> <ul style="list-style-type: none"> <li>• Rectangle R=&lt;TickWidth&gt;,&lt;Colour&gt;,&lt;X&gt;,&lt;Y&gt;,&lt;Width&gt;,&lt;Height&gt;</li> <li>• Filled Rectangle FR=&lt;TickWidth&gt;,&lt;Colour&gt;,&lt;Filling Colour&gt;,&lt;X&gt;,&lt;Y&gt;,&lt;Width&gt;,&lt;Height&gt;</li> <li>• Poygon R=&lt;TickWidth&gt;,&lt;Colour&gt;,&lt;X1&gt;,&lt;Y1&gt;,&lt;X2&gt;,&lt;Y2&gt;,...&lt;Xn&gt;,&lt;Yn&gt;</li> <li>• Filled Poygon FR=&lt;TickWidth&gt;,&lt;Colour&gt;,&lt;FillingColour&gt;,&lt;X1&gt;,&lt;Y1&gt;,&lt;X2&gt;,&lt;Y2&gt;,...&lt;Xn&gt;,&lt;Yn&gt;</li> <li>• Circle K=&lt;TickWidth&gt;,&lt;Colour&gt;,&lt;X&gt;,&lt;Y&gt;,&lt;Radius&gt;</li> <li>• Filled Circle KR=&lt;TickWidth&gt;,&lt;Colour&gt;,&lt;FillingColour&gt;,&lt;X&gt;,&lt;Y&gt;,&lt;Radius&gt;</li> </ul>
Visible	Defines if the graph has to be displayed.
Title	Defines the title text of the graph.
Title: Font	Defines the font to be used for the title text.
Title: X-Pos	X-position at which the title has to appear.
Title: Y-Pos	Y-position at which the title has to appear.
Var: Clear Graph	PLC variable which allows clearing the graph.
Var: Read Pointer	PLC variable which includes the read pointer. This variable informs the PLC up to which value data is read out already (ring buffer).
Var: Write-Pointer/Number	PLC variable which includes the write pointer. This variable is used by the PLC to indicate the point up to which data is

	written (ring buffer). If no ring buffer is used, the number of triples will be indicated here.
Var: Visible	PLC variable used to display or respectively, mask out the graph at runtime.
Var: Script	PLC variable used to draw objects into the graph at runtime. The same commands as used for the Script parameter may be used here.
Var: Save	PLC variable by means of which the data of the displayed graph can be saved in a file. After successful saving the filename will be cleared from the User Interface. The formats ASCII-Text, XML and Bitmap will be supported. The file type is determined by a filename extension (.txt = ASCII-Text, .xml = XML and .bmp = Bitmap).
Var: Points	PLC variable which includes the points. This has to be an array providing three entries for each point (X, Y, and colour). Example: Point: array[1..12]; Includes the data for 4 points. Note: The size of the array has to be indicated, separated by a semicolon, e.g. : .arlrValues:90
Var: X-Axis Min.	PLC variable used to change the X-axis minimum at runtime.
Var: X-Axis Max.	PLC variable used to change the X-axis maximum at runtime.
Var: Y-Axis Min.	PLC variable used to change the Y-axis minimum at runtime.
Var: Y-Axis Max.	PLC variable used to change the Y-axis maximum at runtime.
X-Axis: Unit	Defines the unit of the X-axis.
X-Axis: Unit X-Pos	Defines the X-position of the X-axis unit.
X-Axis: Unit Y-Pos	Defines the Y-position of the X-axis unit.
X-Axis: Colour	Defines the colour of the X-axis.
X-Axis: Max-Value	Defines the maximum of the X-axis.
X-Axis: Min-Value	Defines the minimum of the X-axis.
X-Axis: Position	Defines the position of the X-axis (Top, Bottom, or the value set with "X-Axis: Y-Position").
X-Axis: Scale Interval	Defines the interval of the X-axis scaling.
X-Axis: Scale Type	Defines the scale type (linear or logarithmic) of the X-axis.
X-Axis: Scale Format	Defines the format used for scaling the X-axis. This format can be defined as follows, e.g.: 0.00 = digit with two post-decimal places.
X-Axis: Y-Position	Defines the Y-position of the X-axis.
Y-Axis: Unit	Defines the unit of the X-axis.

Y-Axis: Unit X-Pos	Defines the X-position of the Y-axis unit.
Y-Axis: Unit Y-Pos	Defines the Y-position of the Y-axis unit.
Y-Axis: Colour	Defines the colour of the Y-axis.
Y-Axis: Max-Value	Defines the Y-axis maximum.
Y-Axis: Min-Value	Defines the Y-axis minimum.
Y-Axis: Position	Defines the position of the Y-axis (Left, Right, or defined by the value set under “Y-Axis: X-Position”).
Y-Axis: Scale Interval	Defines the scaling interval for the Y-axis.
Y-Axis: Scale Type	Defines the scale type (linear or logarithmic) for the Y-axis.
Y-Axis: Scale Format	Defines the format used for scaling the Y-axis. This format can be defined as follows, e.g.: 0.00 = digit with two post-decimal places.
Y-Axis: X-Position	Defines the X-position of the Y-axis.

### Graph Object



An LED object is used to display an LED.



Property	Description
Width	Defines the width of the object.
Colour	Defines the colour of the object (if not defined via the PLC variable).
Height	Defines the height of the object.
Name	Defines the name of the object.
Position X	Defines the X-position of the object.
Position Y	Defines the Y-position of the object.
Region	Indicates how many points are currently within a region. The region can be newly defined by pressing the Region key. <div style="border: 1px solid black; padding: 2px; display: inline-block;">                     5 Punkte <span style="border: 1px solid black; padding: 0 2px;">...</span> </div> To define the region, please proceed as follows:

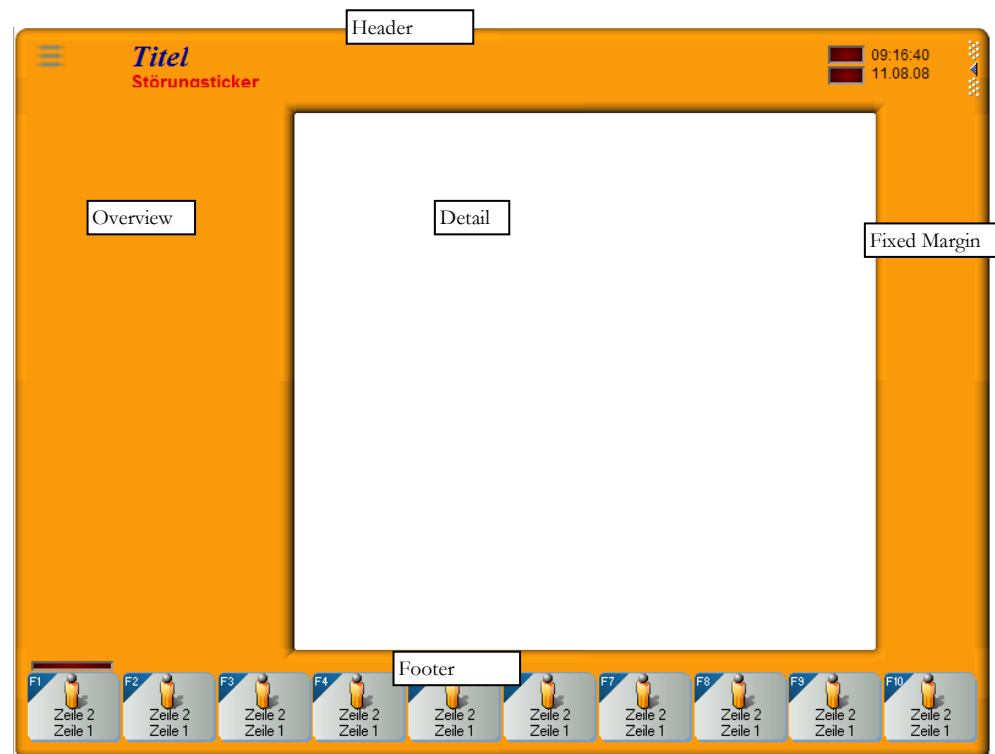
## CHAPTER 5: DISPLAY ELEMENTS

	<ul style="list-style-type: none"><li>• Press the “...” key.</li><li>• Within the graph, consecutively click the desired points with your left mouse button.</li><li>• After selecting all the points, terminate your entry with the right mouse button.</li></ul>
Transparency	Defines the degree of transparency of the object. 0 = Completely transparent. 255 = Solid.
Type	Defines the Object type. Possible types are: <ul style="list-style-type: none"><li>• Rectangular</li><li>• Ellipsoid</li><li>• Polygonal</li></ul>
Var: Colour	PLC variable used to change the colour of the object at runtime.
Var: Visible	This PLC variable is used to display or mask out an object at runtime.

## Skin Data

*The Skin Data function is used to define the appearance of the BROSIS-Commander.*

The Skin Data can be manipulated via the menu item TOOLS | SKINDEFINITION and maintained in its own files, independently of the project file. Which Skin is used by the User Interface is defined the INI-File of BROSIS Commander. The User Interface of BROSIS-Commander is subdivided into the following sections: Header, Footer, Overview, Detail and Fixed Margin.



The contents of the header and footer sections are interchangeable, i.e. the function key bar of the footer may also appear in the header. In addition, image/graph elements may be displayed in the header and footer section, as well as in the fixed margin.

## Menus

Menu Item	Description
File   Open	Opens a Skin file
File   Save	Saves the Skin file
File   Save as...	Saves the currently opened Skin file under a different filename.
File   Close	Closes the Skin Editor.
View   Show Overview	Displays or hides the Overview section.
View   Show Overview Tree Demo	Displays or hides the menu tree in the Overview section.
View   Fault Ticker	Shows the Fault Ticker in the colour of fault ticker.
View   Warning Ticker	Displays the Fault Ticker in the colour of the warning ticker.

## Settings

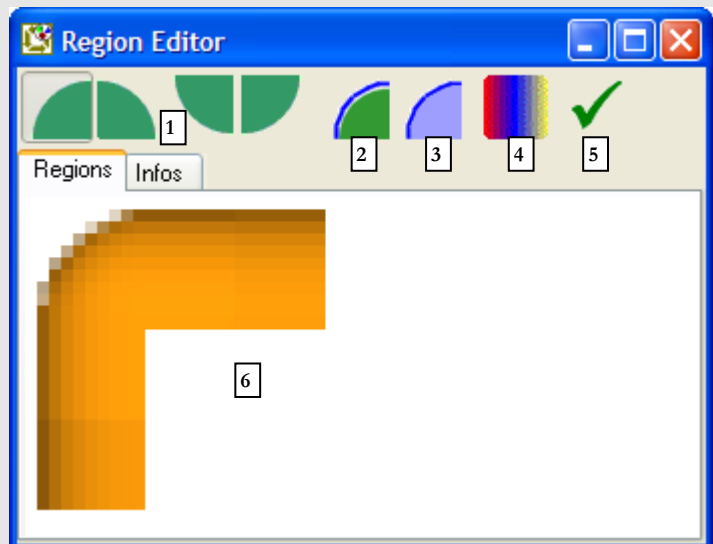
Input	Description
Resolution X	Resolution to be applied to the Skin file. The resolution generally depends on the resolution of the monitor of the target computer.
Resolution Y	Resolution to be applied to the Skin file. The resolution generally depends on the resolution of the monitor of the target computer.
Width Overview	Indicates the width of the overview section.
Width Overview Off	Indicates the width of the margin on the Overview page, if no overview is shown for a window.
Fixed Margin: Visible	Displays or hides the fixed margin.
Fixed Margin: Left	Defines if the fixed margin is to be displayed on the Left or Right.
Fixed Margin: Outside	Defines if the fixed margin trims the header and footer.
Fixed Margin: Width	Defines the width of the fixed margin.
Overview Left	Defines if the overview section is to be displayed on the Left or Right.
Position of Function Keys	Defines if the function keys are to be displayed in the Header or Footer area.
Font: General	Defines the general font type to be used. This font type will be used for all text elements except image elements (where it is freely selectable).
Font: Faults	Defines the font type used to display the fault ticker.



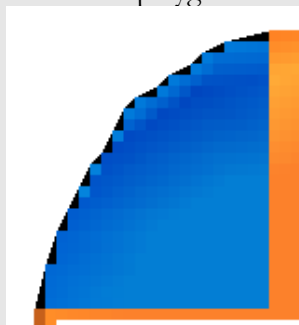
**CHAPTER 7: CUSTOMISED  
DELPHI - PROGRAMME**

Font: Title	Defines the font type used for the title.
Font: Title	Allows entering any desired colour for the title text.
Detail: Background Colour	Defines the colour to be used for the Detail area.
Overview: Background Colour	Defines the colour to be used for the remaining section.
Colour: Fault Text	Defines the colour used to display the fault text in the fault ticker.
Colour: Warning Text	Defines the colour used to display the warnings in the fault ticker.
Font size: Overview	Defines the font size used for the Overview section.
Font size: Footer	Defines the font size used in the Footer section.
Font size: Other	Defines the font size of other text.
Font size: Date/Time	Defines the font size used to display the clock.

Dialogue-Region Used to define the shape of the corners of the dialogue windows.



1. These keys allow you to define which corner you wish to edit.
2. Press a key to start drawing a polygon (in a clockwise direction).
3. Shows the polygon used for shaping the corner.





4. Adjusts the background colour in Section 6.
5. Closes the dialogue and saves the changes.

### **Title Section**

<b>Input</b>	<b>Description</b>
Height	Height of the title section in pixels.
Date/Time: X-Pos	X-position of the clock
Date/Time: Y-Pos	Y-position of the clock
LED 1: X-Pos	X-position of the first LED
LED 1: Y-Pos	Y-position of the first LED
LED 1: Width	Width of the first LED
LED 1: Height	Height of the first LED
LED 2: X-Pos	X-position of the second LED
LED 2: Y-Pos	Y-position of the second LED
LED 2: Width	Width of the second LED
LED 2: Height	Height of the second LED
Logo: X-Pos	X-position of the logo
Logo: Y-Pos	Y-position of the logo
Title: X-Pos	X-position of the title
Title: Y-Pos	Y-position of the title
Faults: Width	Width of the fault ticker
Faults: X-Pos	X-position of the fault ticker
Faults: Y-Pos	Y-position of the fault ticker
Menu tree: X-Pos	X-position of the menu tree for direct access to the dialogues
Menu tree: Y-Pos	Defines the Y-Position of the menu tree for direct access to the dialogues.

### **Function Keys**

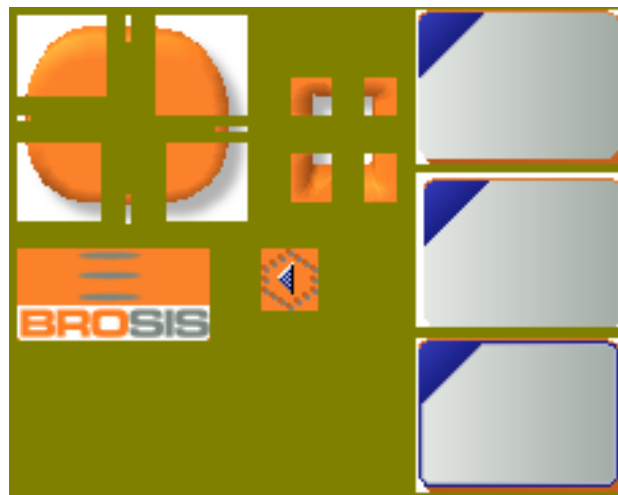
<b>Input</b>	<b>Description</b>
Height	Defines the height of the section (Header or Footer) in which the function keys are displayed.
Number Function Keys	Defines the number of function keys – usually 10 or 12. More than 12 function keys cannot be operated via the keyboard.
From X-Pos	X-position starting from which the function keys have to be drawn.
To X-Pos	X-position up to which the function keys have to be drawn.
Y-Pos	Y-Position at which the function keys have to be drawn.

Text Line 1: X-Pos	X-position of the function key text (line 1). The position always starts from the upper left corner of the key and defines the centre of the text.
Text Line 1: Y-Pos	Y-Position of the function key text (line 1). The position always starts from the upper left corner of the key and defines the centre of the text.
Text Line 2: X-Pos	X-Position of the function key text (line 2). The position always starts from the upper left corner of the key and defines the centre of the text.
Text Line 2: Y-Pos	Y-Position of the function key text (line 2). The position always starts from the upper left corner of the key and defines the centre of the text.
Image: X-Pos	Defines the X-position of the image centre, starting from the upper left corner of the key.
Image: Y-Pos	Defines the Y-position of the image centre, starting from the upper left corner of the key.
Pressed: Offset X	Offset in X-direction of the pressed key, used for text and image. This is required to obtain a 3D-effect.
Pressed: Offset Y	Offset in Y-direction of the pressed key, used for text and image. This is required to obtain a 3D-effect.
LED: X-Pos	Defines the X-position of the LED relative to the corner of the key.
LED: Y-Pos	Defines the Y-position of the LED.
LED: Width	Defines the width of the LED.
LED: Height	Defines the height of the LED.
LED: Hide	Defines whether or not the LED has to be hidden if the colour is black or the variable is not linked.
Title: X-Pos	Defines the X-position of the key title. 
Title: Y-Pos	Defines the Y-position of the key title. 
Font: Text	Defines the font type for the function key text.
Font: Title	Defines the font of the key title.

### Images

Allows defining all the images required for drawing the Skin. The corners and borders have to be adapted (see image). To change an image, first select the image from the list, then change the entry. Using the context menu, the images can be copied (CTRL+K), inserted (CTRL+E) and deleted (CTRL+DELETE).

**CHAPTER 7: CUSTOMISED  
DELPHI-PROGRAMME**

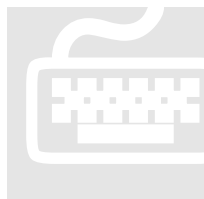


Input	Description
Transparent	Defines if an image has to appear transparent. In this case the colour of the lower left pixel determines the transparent colour.
Stretched	Defines if the image has to appear stretched or in its original size.
Image	Allows defining the image.

## Customised Delphi- Programme

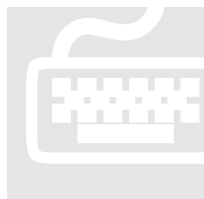
*Special functions may be implemented by means of programming customised windows in Delphi (Version 7)*

This programming option allows displaying customised special dialogues programmed in Delphi. The “Development” directory of the CD is required for this purpose. It includes the directories `.\dcu` and `.\lib`, which again include the libraries of the development environment. The Development directory includes the Delphi project file as well as the `Verteiler.pas` file.



### Delphi Settings

No special settings are required in Delphi. Just copy the contents of the directory `\Development` on your hard disk and open the Project **BROSISCommander.dpr** in Delphi (Version 7 minimum). The project should now be able to be compiled.



### Creating a Dialogue

The window type “Special Dialogue (Delphi)” is used to integrate windows created with Delphi in your User Interface. The Delphi window has to be derived from the Class **TFoParentSonderFenster**. To embed your own programmed dialogue in the User Interface, please proceed as follows:

1. Create a special Delphi window in the Project Administration and assign a special dialogue ID.
  2. Create a new window from the Class `TFoParentWindow` by opening the tab `BROSISCommander` in Delphi under `FILE|NEW|PROCEED` and selecting `FoParentSpecialWindow`.
  3. Assign the **Create** functions (minimum) to this new window.
-

### Example: Definition

```
{
*****
* Name: Create
*
* The window is created
*****}
class procedure TFoFenster.Kreieren
(   AParent: TComponent;
    AFensterDaten: TFensterParent;
    ASpsVariablen: TVariablenListe;
    ASkin: TSkinDaten;
    AProjekt: TProjekt;
    ATypDaten: TTypListe;
    APanelNr, ASprachCode: Integer;
    ARechte: TRechteKlassen;
    AToolFenster: Boolean = False
);
var
    Fenster: TFoFenster;
begin
    inherited;

    { Create window and save parameters }
    Fenster := TFoFenster( KreierenAusfuehren
        (   TFoParentFenster( TFoFenster.Create( AParent ) ),
            AParent,
            AFensterDaten,
            ASpsVariablen,
            ASkin,
            AProjekt,
            ATypDaten,
            APanelNr,
            ASprachCode,
            ARechte,
            AToolFenster
        ) );

    { Register PLC Variables }
    Fenster.VariablenAnmelden;
    Fenster.LadeSprache( ASprachCode );
    Fenster.RechteGeaendert( ARechte );
end;
```

4. The following units are to be integrated in the Uses Part:  
WindowData, PLCData, GeneralData, ObjectData, GenTypeConstants, Skin-Data, ProjectData, Functions and UtilityFunctions.  
(FensterDaten, SpsDaten, AllgemeineDaten, ObjektDaten, AllgTypenKonstan-ten, SkinDaten, ProjektDaten, Funktionen and DienstFunktionen)
5. Use the function **RegisterVariable / VariableAnmelden** to register variables at the PLC. If the contents of the variable should change, the function Variable-Changed will be enabled, thus allowing you to respond to a change.
6. In the file **Verteiler.pas** the unit of the new dialogue has to be entered.
7. Using the function **VerteilerFensterKreieren** the desired dialogue is created based on the ID of the special window entered in the Project Administration.

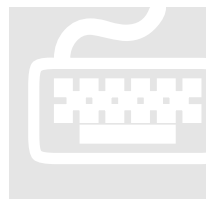
### Example

```
procedure VerteilerFensterKreieren
```

```

(  AParent: TComponent;
  AFensterDaten: TDelphiFenster;
  ASpsVariablen: TVariablenListe;
  ASkin: TSkinDaten;
  AProjekt: TProjekt;
  ATypDaten: TTypListe;
  APanelNr, ASprachCode: Integer;
  ARechte: TRechteKlassen
);
begin
  case AFensterDaten.KennungDetail of
    1:
      TFoFenster.Kreieren
        (  AParent,
          AFensterDaten,
          ASpsVariablen,
          ASkin,
          AProjekt,
          ATypDaten,
          APanelNr,
          ASprachCode,
          ARechte
        );
  end;
end;

```



## Functions of TFoParentSpecialWindow

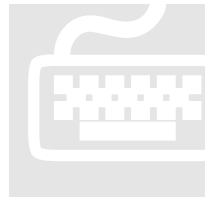
TFoParentSpecial Window offers overwrite functions, which allow you to respond to such events as a change of language, change of user, or change of a PLC variable. The following variables may be overwritten:

Function	Description
RegisterVariable	<p>This function is enabled as soon as the required variables can be registered. Allows access to RegisterVariables (...).</p> <p><b>Declaration</b></p> <pre>procedure VariableAnmelden; override;</pre>
Create	<p>This class function creates the window and has to exist. The definition has to be according to the example above.</p> <p><b>Declaration</b></p> <pre>class procedure Kreieren (  AParent: TComponent;   AFensterDaten: TFensterParent;   ASpsVariablen: TVariablenListe;   ASkin: TSkinDaten;   AProjekt: TProjekt;   ATypDaten: TTypListe;   APanelNr, ASprachCode: Integer;   ARechte: TRechteKlassen;   AToolFenster: Boolean = False ); override;</pre>

ShowData	<p>Enabled, if the window is displayed.</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Declaration</b></p> <pre>procedure DatenAnzeigen ( APanelUebersicht, APanelDetail: TPanel ); override;</pre> </div>
HideData	<p>Enabled, if the window is masked out (hidden).</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Declaration</b></p> <pre>procedure DatenAusblenden ( ADetail: Boolean ); override;</pre> </div>
VariableChanged	<p>Allows responding to changes in the variables.</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Declaration</b></p> <pre>procedure VariableGeaendert ( ASpsVariable: TVariable; AId: Integer; ADaten: TObject ); override;</pre> </div>
LoadLanguage	<p>This function is enabled if the language is changed. All the dialogue texts should be changed here. The texts of the dialogue may be accessed via <b>Daten.Texte.SprachTextByKennung</b> (<b>Data.Texts.LanguageTextByID</b>).</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Declaration</b></p> <pre>procedure LadeSprache ( ASprachCode: Integer ); override;</pre> </div>
RightsChanged	<p>This function is enabled if the rights are changed. In this way, the elements of the window can be accessed according to the rights assigned.</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Declaration</b></p> <pre>procedure RechteGeaendert ( ARechte: TRechteKlassen ); override;</pre> </div>
KeyPressed	<p>Enabled, if a key is pressed.</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Declaration</b></p> <pre>procedure TasteGedrueckt ( AKennung, ATasteNr: Integer; var AWeiter: Boolean ); override;</pre> </div> <p>Use AWeiter to determine if further steps (Manipulate variables, Open Dialogues...) have to be carried out (True) or not (False).</p>



KeyReleased	This function is enabled if a function key is released.		
	<table border="1"> <tr> <td><b>Declaration</b></td> </tr> <tr> <td> <pre> <b>procedure</b> TasteLosgelassen   (     AKennung,     ATasteNr: Integer   ); <b>override;</b> </pre> </td> </tr> </table>	<b>Declaration</b>	<pre> <b>procedure</b> TasteLosgelassen   (     AKennung,     ATasteNr: Integer   ); <b>override;</b> </pre>
<b>Declaration</b>			
<pre> <b>procedure</b> TasteLosgelassen   (     AKennung,     ATasteNr: Integer   ); <b>override;</b> </pre>			
TypeChanged	Enabled, if the content of an Article entry has changed.		
	<table border="1"> <tr> <td><b>Declaration</b></td> </tr> <tr> <td> <pre> <b>procedure</b> TypGeaendert   (     AId: Integer;     ATyp,     ATypAlt: TTyp;     AAktion: TTypAktion   ); <b>override;</b> </pre> </td> </tr> </table>	<b>Declaration</b>	<pre> <b>procedure</b> TypGeaendert   (     AId: Integer;     ATyp,     ATypAlt: TTyp;     AAktion: TTypAktion   ); <b>override;</b> </pre>
<b>Declaration</b>			
<pre> <b>procedure</b> TypGeaendert   (     AId: Integer;     ATyp,     ATypAlt: TTyp;     AAktion: TTypAktion   ); <b>override;</b> </pre>			



## Useful Utility Functions

TFoParentSpecialWindow offers utility functions, which allow accessing PLC variables or determining language-independent texts from the project, etc.

In addition to these functions, the PropertyData function allows access to the project data of the window.

### Function: RegisterVariable

Registers variables to the PLC, so that they can be read out and written by the programme.

Parameter	Type	Description
AIdentification	string	ID of the variable (as requested by the PLC).
APLCNumber	Integer	Number of the PLC. This number is 0, if the PLC number defined in the project has to be used.
ARefreshRateMs	Integer	Refresh rate of the variable. -1 = Write Only 0 = Server Default 1 = As fast as possible n = Every n milliseconds
AType	TVariableType	Type of the variables. Possible types are: vtBit, vtByte, vtWord, vtDWord, vtShortInt, vtInteger, vtLongInt, vtReal, vtLongReal and vtString.
AArraySize	Integer	Defines the variable number (for an array). Arrays of the type vtString are not supported.

AData	Pointer	Own data to be appended to the variable.
AIId	Integer	Own ID for easier identification of variables.

Tvariable is returned – this function also allows writing into the variable.

### **Function: Dat.Texts.LanguageTextByID**

Determines language-independent text. This text has to be defined for the Special Dialogue.

Parameter	Type	Description
AIIdentification	string	ID of the text entered within the Project Administration of the Special Dialogue.
ADefault	string	Text to be used if a text with this ID does not exist.
ALanguageCode	Integer	Language code defined in the Language.dat file.

The string with the text of the defined language will be returned.

### **Function: ShowWindow**

Shows the selected window. The windows are defined within the Tproject Class (Variable:Project).

The **Functions** unit has to be embedded!

Parameter	Type	Description
AWindow	TWindowParent	Window object describing the window. You will obtain this object via the entries in <b>Project.Window</b> . For example, you may determine the window object via Project.Window.FindByWindowNo.(Integer).

### **Function: GetTypeInformation**

Determines a specific Article Data type. Types may be determined by the TType-List class (Variable: TypeData).

Example: Type.Data.FindByID(AIIdentification:string).  
(TypDaten.FindByKennung( AKennung: string )).

The **Functions** unit has to be embedded!

Parameter	Type	Description
AType	TType	Type the Article input of which is to be searched.
AProject	TProject	Project Data (Variable: Project).
AObjectName	String	Name des object to which the Article Data are subordinated. If an empty string is returned here, the first Article entry corresponding to the se-

		lected variable name will be determined.
AVariableName	String	Name of the PLC variable of the Article entry.
ADefault	String	Value to be returned if the entry is not found.

The content of the Article entry is returned as a string.

### **Function: SetTypeInfo**

Defines the value of Article Data of a type. Types can be determined via the TTypeList class (Variable: TypData ), e.g. TypeData.FindByID (AIdentification: string)). To save the changed content, **SendChangeType( ATyp: TTyp )** has to be enabled afterwards.

The **Functions** unit has to be embedded!

Parameter	Type	Description
AType	TType	Type the Article input of which is to be searched.
AProject	TProject	Project data (Variable: Project).
AObjectName	String	Name of the object to which the Article Data are subordinated. If an empty string is returned here, the first Article entry corresponding to the selected variable name will be determined.
AVariableName	String	Name of the PLC variable of the Article entry.
AValue	String	Value to be entered into the Article data.

The function will return True after the value has been entered.

### **Function: MessageOk**

Opens a message to be acknowledged with OK by the User.

Parameter	Type	Description
AIdentification	String	ID of the message text (as entered within the Project Administration of the Special Dialogue).
ADefault	String	Text to be used if a text with ID does not exist.

### **Function: MessageYesNo**

Opens a message to be acknowledged by the User with Yes or No:

Parameter	Type	Description
AIdentification	String	ID of the message text (as entered within the Project Administration of the Special Dialogue).
ADefault	String	Text to be used if a text with ID does not exist.

The function will return True, if the user has closed the dialogue with Yes.

### Function: ValueInputFloat

Enables dialogue-controlled entry of a floating-comma digit.

Parameter	Type	Description
ATitle	String	Title of the input dialogue.
ADecimalPlaces	Integer	Number of decimal places for the display.
AMin	Double	Lower input limit.
AMax	Double	Upper input limit.
var AValue	Double	Value to be changed.

The function will return True after the User has closed the dialogue with OK.

### Function: ValueInputInt

This function allows dialogue-controlled input of an integer.

Parameter	Type	Description
ATitle	String	Title of the input dialogue.
AMin	Integer	Lower input limit.
AMax	Integer	Upper input limit.
var AValue	Integer	Value to be changed. .

The function will return True, if the User has closed the dialogue with OK.

### Function: ValueInputBool

Allows dialogue-controlled input of a Boolean value.

Parameter	Type	Description
ATitle	String	Title of the input dialogue.
var AValue	Boolean	Value to be changed.

The function will return True, if the User has closed the dialogue with OK.

### Function: ValueInputText

Allows dialogue-controlled input of a text.

Parameter	Type	Description
ATitle	String	Title of the input dialogue.
AMask	String	Input mask (as in TmaskEdit).
var AValue	String	Value to be changed.

The function will return True, if the User has closed the dialogue with OK.

### Function: ValueInputSelection

Allows dialogue-controlled input of a text according to a shortlist.

Parameter	Type	Description
Atitle	String	Title of the input dialogue.
AList	String	List of possible values. The list has to be in the following format: <Value>;<Input><CR><LF>...
var AValue	String	Value to be changed.

The function will return True if the User has closed the dialogue with OK.

### Function: ValueInputValueRange

This function allows dialogue-controlled input of any desired value by means of an object of the TvalueRange class.

Parameter	Type	Description
ATitle	String	Title of the input dialogue.
AValueRange	TValueRange	Value Range.
var AValue	String	Value to be changed.

The function will return True if the User has closed the dialogue with OK.

### Function: SetFuncKeyColourByID

Allows setting the LED colours via the function keys.

Parameter	Type	Description
AIdentification	Integer	ID of the function key.
AColour	TColour	Colour.
ADrawNew	Boolean	Defines if the keys have to be redrawn. If the colours of several keys are to be changed at the same time, only the value of the last key has to be set to True.

### Function: SetFuncKeyVisibleByID

Allows defining the visibility of colours of the function keys.

Parameter	Type	Description
AIdentification	Integer	ID of the function key.
AVisible	Boolean	Visible or non-visible.
ADrawNew	Boolean	Defines if the keys have to be redrawn. If the colours of several keys are to be changed at the same time, only the value of the last key has to be set to True.



## Reading and Writing of PLC Variables

For objects of the Tvariable class (these can be created by means of the RegisterVariable function), the content of a variable can be determined and written. Read and Write access to the variable content is possible via Properties; this function represents the various file types. The index in the array (0...n) always has to be indicated here (strings are excepted).

Parameter	Type
AsBit[ AIndex ]	Boolean
AsByte [ AIndex ]	Byte
AsWord [ AIndex ]	Word
AsDWord [ AIndex ]	DWord
AsShortInt [ AIndex ]	ShortInt
AsInteger [ AIndex ]	SmallInt
AsLongInt [ AIndex ]	LongInt
AsReal [ AIndex ]	Single
AsLongReal [ AIndex ]	Double
AsString	string

In addition, the settings of the variables may be accessed:

Parameter	Type
PLCNumber	Integer
Identification	String
RefreshRateMs	Integer
Type	TVariableType

## Miscellaneous

### *Description of Input Dialogues*



#### Input of Variables

A dialogue is presented for the definition of PLC variables. The variable data may be entered here. Except for the default line (fourth line), this dialogue will always appear in the same format.

Description	Value
Name	.bTasteHand
Type	Bit
Description	[Value 1]/[Toggle Value1]
Refresh-Rate [ms]	Server-Value
PLC no.	PLC-no. of the Object

Input	Description
Name	<p>Defines the name of the PLC variable. The following wildcards may be used:</p> <ul style="list-style-type: none"> <li>• <b>{Name}</b> Used to enter the name of the object; this allows defining the variable name with reference to the object.</li> <li>• <b>{Station}</b> Used to enter the name of the station; this allows defining the variable name with reference to a station.</li> <li>• <b>{Panel}</b> This wildcard is replaced with the number of the panel, thus making it possible to define the variable depending on the panel.</li> </ul>

	If an array should be necessary, the number of elements should be defined as follows: <b>&lt;VariableName&gt;:&lt;NumberElements&gt;</b>
Type	Defines the Data type of the variable.
Description	Internal description.
Default Value	The default value may be different for each variable, or it may not exist at all, if it is not required. The expected input for the default value is shown in the Description column. [Value 1]/[Toggle Value2] (e.g.: 0/1) for a key toggles between Value 1 and value 2. If Value 2 is not entered, Value 2 will be set.
Refresh-Rate [ms]	Defines how fast the content of the variables has to be refreshed. The following values are possible: -1 = Write Only (Reading is not necessary) 0 = Default value of the server 1 = As fast as possible n = Refresh rate in ms
PLC No.	Number of the PLC to which the variable has to be registered. If 0 is entered here, the PLC of the superimposed variable object will be used.

**Note:**

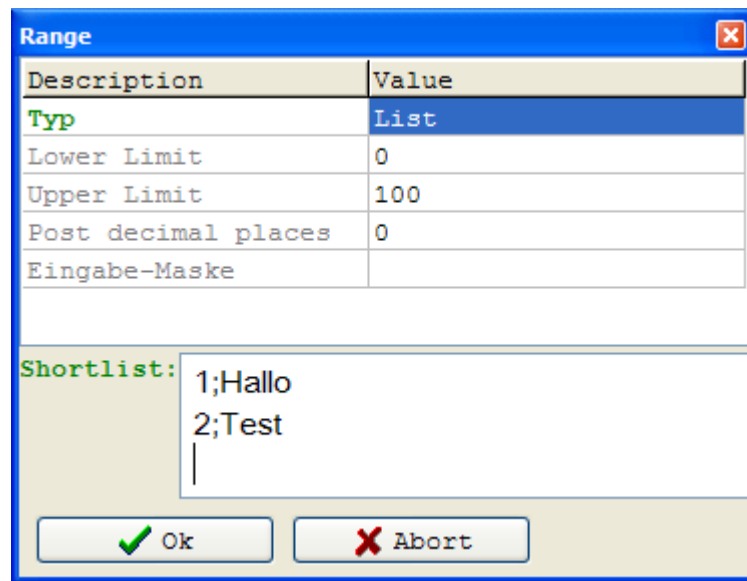
- The default value of the server for the refresh rate can be set in the INI-file of the server.
- Variables that are interdependent, e.g. Article Data transferred from the User Interface to the PLC, or variables used for the display of a message window, have to be set to a refresh rate of 1. This ensures that the required data are available when triggered (e.g. when opening the message window).



**Input of a Value Range**

By entering a value range you can define which values are admissible for a variable. We distinguish between a value range (for numerical values), a shortlist, and an input mask.





Input	Description
Type	Defines the type of input: Value Range, Shortlist, or Input Mask.
Lower Limit	Defines the lower limit for numerical input. This function is only required for the value range.
Upper Limit	Defines the upper limit for numerical input. This function is only required for the value range.
Decimal Places	Defines the number of decimal places for the display of a digit. This function is only needed for the value range.
Input Mask	Input Mask used for entering values (only needed for input masks).
ShortList	Lists all the available values. This list may include a value-text allocation. The following format has to be observed:  <b>&lt;Value&gt;;&lt;Text&gt;</b>  This function is only needed for shortlists.

**Example – Shortlist**

You wish to select between “Yes” and “No” when entering a Boolean value. The shortlist will now appear as follows: 0;No  
1;Yes

### Input Mask

Format of data in an input mask:

<Mask>;<0=Do not save literals| 1=Save literals>;<DrawLiterals>

Character	Description
!	If the sign ! is used within a mask, optional characters in the text will be displayed as blank characters. If the sign! is missing, the optional characters in the text will be displayed as final blank characters.
>	If the sign > is used in the mask, all subsequent signs up to the end of the mask, or up to the sign < will be displayed in capitals.
<	If the sign < is used in the mask, all subsequent signs up to the end of the mask, or up to the sign > will be displayed in small letters.
<>	If both signs are used together in the mask, there will not be any check for capitals/small letters. Capitals and small letters will be entered as defined by the User.
\	The sign after \ has to be interpreted as a literal. Use this sign, if you wish to include one of the mask characters in the data as a literal.
L	The L sign requests a letter in this position.
l	The l sign permits a letter in this position, but does not request it.
A	The letter A requires an alphanumeric sign in this position.
a	The letter a permits an alphanumeric sign in this position, but does not require it.
C	The letter C requests entry of any sign in this position.
c	The letter c permits entry of any sign in this position, but does not request it.
0	The 0 sign requests a numerical character in this position.
9	The sign 9 permits a numerical character in this position, but does not request it.
#	The sign # permits a numerical sign or a plus or minus sign in this position, but does not request this input.

:	The character : is used to separate time entries into hours, minutes, and seconds. If the country settings in the System Control of your computer use a different type of separator, this will be used instead of .:
/	The / sign is used to separate date entries into years, months, and days. If the country settings in the System Control of your computer use a different type of separator, this will be used instead of /.
;	The ; sign is used to divide the three fields of the mask.
_	The _ sign is used to insert blank spaces into the text automatically. If the user enters a character in the field, the cursor will skip the _ sign.

Examples of Masks		
>cc-99999;1;_	ZIP code international	CH-50000
000\ 000\ 00;1;_	Bank code number	500 100 00
!90:00:00;1;_	Time format, long	12:30:12
!99/99/00;1;_	Date	01.01.2003



### Opening an Image

To open an image, a dialogue is presented with a Preview of the image and an option to save a reference to the image. The image can also be deleted from the project.

There are two principal options of assigning images to a project, i.e. by saving an image normally or by saving the file reference only. Which mode has to be used can be defined for each image under **Only Save Reference to File**.

If you decide to save the image in the “normal” way, the image will be saved together with the project file. This may, however, be disadvantageous in that the file size will increase (a very large file means that more memory space is required to load it), but the benefit is that all data required for the project are combined in one single file.

To save a reference to the image file only, the file has to exist in a fixed directory (both of the Client and the Project Definition). The image directory can be set in the INI-File under

```
[Settings]
PathExternalImages = .\Images
```

