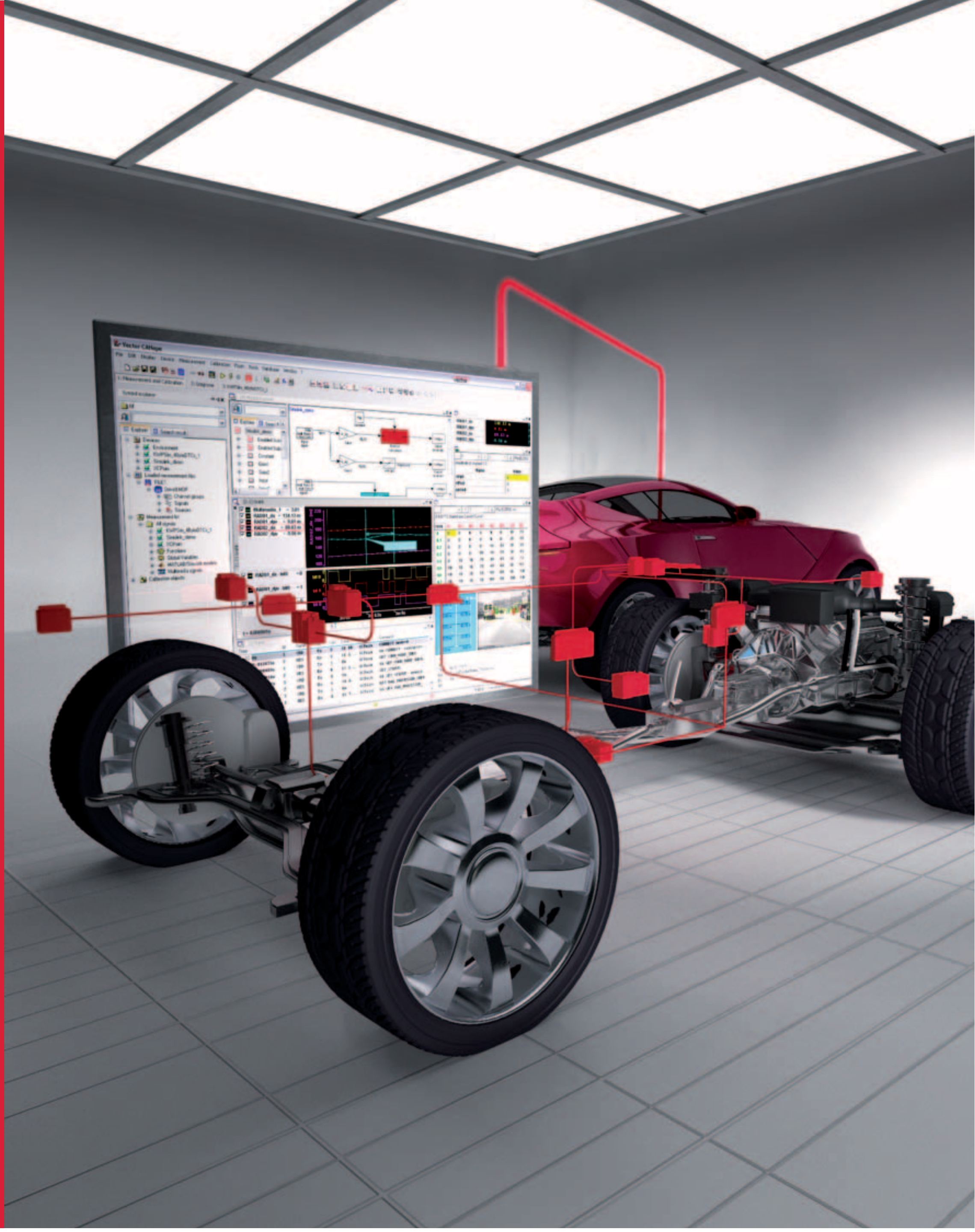

Manual

ASAP2 Editor

ASAP2 Editor

Manual

Version 13.0

Vector Informatik GmbH, Ingersheimer Straße 24, D-70499 Stuttgart
Phone: +49.711.80670.0, Fax: +49.711.80670.111, Email info@de.vector.com
Internet http://www.vector.de

Certified Quality/Process Management

The Quality/Process Management of the whole Vector Group is certified according to DIN EN ISO 9001:2000-12.

Worldwide Subsidiaries

Germany and all countries not named below **Vector Informatik GmbH** Ingersheimer Straße 24 70499 Stuttgart Germany Phone: +49.711.80670.0 Fax: +49.711.80670.111 info@de.vector.com http://www.vector.de	**USA, Canada, Mexico** **Vector CANtech, Inc.** Suite 550 39500 Orchard Hill Place Novi, Mi 48375 USA Phone: +1.248.449.9290 Fax: +1.248.449.9704 info@us.vector.com http://www.vector-cantech.com
France, Belgium, Luxemburg **Vector France SAS** 168, Boulevard Camélinat 92240 Malakoff France Phone: +33.1.4231.4000 Fax: +33.1.4231.4009 info@fr.vector.com http://www.vector-france.com	**Japan** **Vector Japan Co., Ltd.** Seafort Square Center Bld. 18F 2-3-12, Higashi-shinagawa, Shinagawa-ku Tokyo 140-0002 Japan Phone: +81.3.5769.7800 Fax: +81.3.5769.6975 info@jp.vector.com http://www.vector-japan.co.jp
United Kingdom & Ireland **Vector GB Limited** Rhodium, Central Boulevard Blythe Valley Park Solihull, Birmingham West Midlands, B90 8AS United Kingdom Phone: +44.121.50681.50 Fax: +44.121.50681.66 info@uk.vector.com http://www.vector-gb.co.uk	**Korea** **Vector Korea IT Inc.** Daeryung Post Tower III, 508 182-4 Guro-dong, Guro-gu Seoul, 152-790 Republic of Korea Phone: +82.2.2028.0600 Fax: +82.2.2028.0604 info@vector-korea.com http://www.vector-korea.com
Sweden, Denmark, Norway, Finland, Iceland **VecScan AB** Theres Svenssons Gata 9 417 55 Göteborg Sweden Phone: +46.31.764.7600 Fax: +46.31.764.7619 info@se.vector.com http://www.vecscan.com	**India** **Vector Informatik India Private Limited** Regus Connaught Place Level 2, Connaught Place Bund Garden Road Pune 411 001 India Phone: +91.20.4014.7673 Fax: +91.20.4014.7576 info@in.vector.com http://www.vector-france.com

Typographic Conventions

Caution
This symbol calls your attention to warnings.

Info
Here you can obtain supplemental information.

Practical Procedure
Step-by-step instructions provide assistance at these points.

Example
Here is an example that has been prepared for you.

Edit
Instructions on editing files are found at these points.

Do not edit manually
This symbol warns you not to edit the specified file.

FAQ
In this area you can get answers to frequently asked questions.

[OK]	Notation for buttons in dialogs
<TAB>	Notation for keys on the computer keyboard
<Ctrl>+<Z>	Notation for keys on the computer keyboard which should be pressed simultaneously
Add...	Notation for menu, command and dialog names
File | Save file	**Save file** is identified as a command in the **File** menu
`on message 0x100`	Notation for the function/script syntax

Contents

1 Introduction

This manual explains how to use the ASAP2 Editor. The editor manages the different controller databases. This editor allows you to view, input and change all information in a dialog-driven format.

Info
You can execute the ASAP2 Editor also without a valid license. In this case it runs in the "Viewer" mode.

In this mode no changes of the database are possible (creating, editing and deleting of objects, import and export of databases...); the loaded database cannot be saved or exported. The respective menu commands are disabled (grayed out).

To purchase a license please contact the Vector support: canape-support@de.vector.com.

Here we will also show how to create a new controller database that is appropriate for your ECU. We also describe how existing controller description files can be imported and exported.

Info
The terms database and controller database will be used here as synonyms.

1.1 System Requirements

To use ASAP2 Editor the following system requirements apply:

PC with Windows XP (SP3), Vista or Windows 7 (32 and 64 bit editions)

Info
Windows 95/98/ME/2000 and NT are not supported by ASAP2 Editor

1.2 Copy Protection

The data medium is not copy-protected. The licensee is authorized to produce a (single) backup copy. The conditions of the Vector license agreement must be observed.

1.3 New Functions of ASAP2 Editor Version 13.0

- Performance optimization in reading databases
- Extended plausibility check (check memory segments for duplicate names)
- Support keyword `sysc()` for using system constants in algebraic conversion methods, provide picklist selection to insert system constants into formulas
- Extended support for editing more than one variable:
  - The assigned record layout can be changed if all variables have the same record layout at start of dialog and if all have same object type.
  - The dimension of value blocks can be edited.
  - The properties of X axis and Y axis can be modified if all edited curves / maps refer to the same record layout.

2 Installation

The Setup Program starts automatically when the CD is inserted. In the dialog **ASAP2 Editor Setup** click on the button **[Install Application].**If you have deactivated the Autorun Function of your computer, start the program SETUP.EXE on the CD and follow the instructions of the installation program.

The language of the dialogs for the installation process depends on the operating system settings. The language of the ASAP2 Editor user interface can be selected during installation. The language setting can be changed later in the options dialog.

Info
Some Windows standard buttons and dialogs are displayed independently from the ASAP2 Editor language setting in the language set on the computer (e.g. the dialogs for selecting color and font and buttons such as **[OK]** and **[Cancel]**)

Should you have questions or problems:

Telephone:	+49 (0)711/80670-200
Fax:	+49 (0)711/80670-555
e-mail:	canape-support@de.vector.com
Internet:	http://www.vector.de

3 The Basics of How to Create a Controller Database

In general, it is the task of the ECU developer to create an appropriate database for the controller. To create a database, the Linker MAP file of the ECU program is needed. The ECU developer creates all the needed database objects, for example, measurement objects, characteristic curves and maps, with the ASAP2 database editor and the Linker MAP file (see Figure 1).

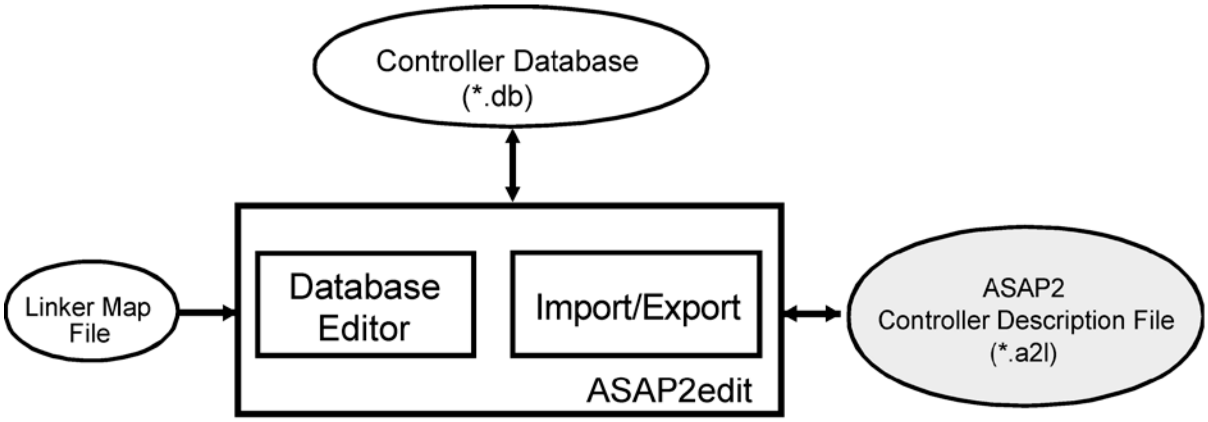

Figure 1: Creating an ECU Database

In Section 4.1, "Creating a Controller Database" this process will be explained and illustrated with examples.

The contents of a controller database depends on the current ECU program release. If the ECU program changes, all the address and data type information on already defined objects in the controller database are updated automatically with the help of the Linker MAP file the next time the program starts.

After the ECU development is finished, the ECU developer can generate an ASAP2 controller description file with the database export function. This ASAP2 file is in general part of the ECU delivery to the calibration engineers. The calibration engineers' task is to optimize the control and calibration parameters so that the technical system meets all requirements.

3.1 Importing a Controller Database

The CANape ASAP2 Editor supports the import of the following database file formats:

- `*.A2L` ASAP2 format
- `*.DB` CANape standard format
- `*.DBU` UPDATE format / Simple ASCII file format for updating information in the database
- `*.ini` ESB Valve description files

- `*.ccd` CANdela database format

To import an ASAP2 file, select the menu item **File | Import** and select the ASAP2 file which is to be imported. If an ECU database has already been loaded, you are asked if the contents of the database should be cleared before import. If you answer with [Yes], the entire contents of the database will be replaced. Otherwise, only the changes and the new entries of the ASAP2 ECU description file are updated in the ECU database. You are also asked at this time whether or not objects already existing in the old database with the same name should be overwritten.

3.2 Exporting the Controller Database

CANape supports export of the controller database in accordance with the following database file formats:

- `*.A2L` ASAP2 Format
- `*.LST` CANape Listing Format
- `*.DB` Several CANape Standard formats

Info
The data base should be exported in the format of the ASAP2 Editor version with which it will be edited later.

- `*.DBB` CANape / Encoded, binary database
- `*.ROB` Gredi (Kleinknecht) format
- `*.CDP` Car-Diag format (CDP = Car-Diag-Preparation)
- *.* CANape Report format

For example, to export an available controller database in CANape format to an ASAP2 file, select the menu item **File | Export**. You will see a message stating that the a2l database already exists. You must then agree to replace it. A name for the file must be indicated for the ASAP2 file to be created, and a directory in which it is to be placed. The desired export format and the interface-specific data which the ASAP2 file is to contain are selected in the next dialog, **ASAP2 Export** (see Figure 2).

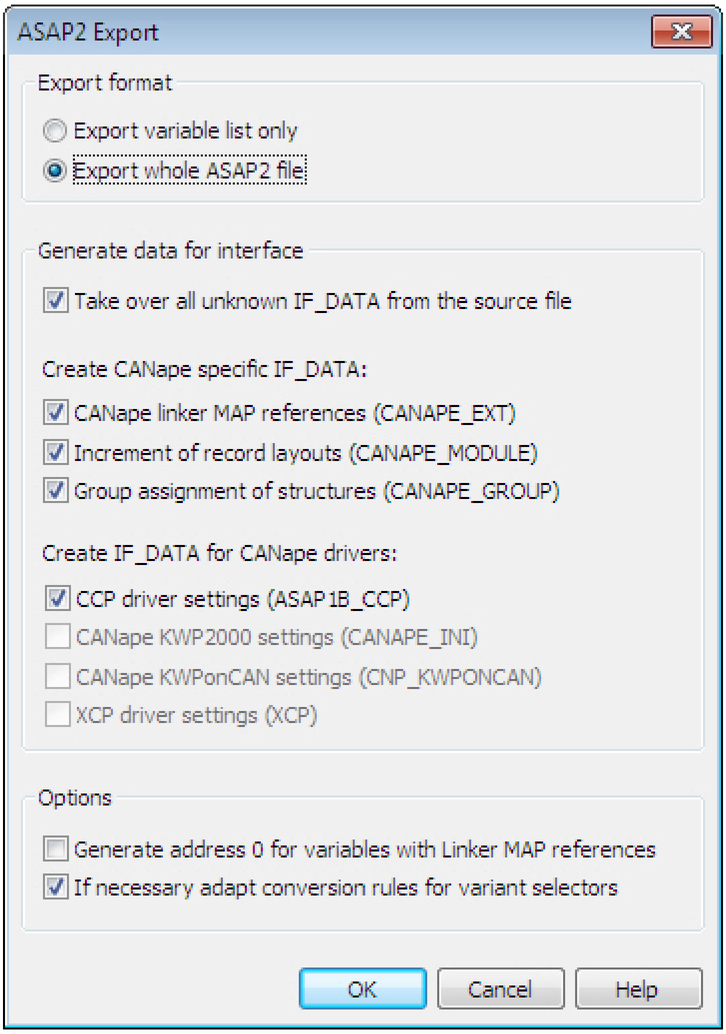

Figure 2: ASAP2 Export Dialog

In this example, the ASAP2 Editor is to generate the entire ASAP2 controller description file with the CCP interface data. The ASAP2 file is created by clicking **[OK]**.

When exporting into the CANape Report format, certain things should be considered: First, it is possible with this format to create an ASCII file in a self-defined format.

To do this, corresponding format templates must be present or created in the respective working directory for all variable types to be exported. This process deals with the following files:

"Value.TEMPL"	Scalar parameters and measurement values
"String.TEMPL"	ASCII strings
"Axis.TEMPL"	Shared axes
"Curve.TEMPL"	Characteristic curves whose sampling points are not in the controller's memory (curves with no axes or with fixed axis)
"CurveX.TEMPL"	Characteristic curves whose sampling points are in the controller's memory (curves with standard or shared axes)
"Map.TEMPL"	Maps whose X sampling points and Y sampling points are not in the controller's memory (no X axis or X fixed axis, no Y axis or Y fixed axis)

"MapX.TEMPL"	Characteristic curves whose X sampling points are in the controller's memory and whose Y sampling points are not in the controller's memory (X standard axis or X shared axis, no Y axis or Y fixed axis)
"MapY.TEMPL"	Characteristic curves whose X sampling points are not in the controller's memory and whose Y sampling points are in the controller's memory (no X axis or X fixed axis, Y standard axis or Y shared axis)
"MapXY.TEMPL"	Characteristic curves whose X sampling points and Y sampling points are in the controller's memory (X standard axis or X shared axis, Y standard axis or Y shared axis)

These files contain macros (placeholders) for the individual description elements of the respective variables. The following macros are replaced in the template files:

$TEMPLATE_NAME$	File name of the template file used
$NAME$	Name of the database variable
$COMMENT$	Comment about the variable from the database
$UNIT$	Physical unit for the variable
$PHYS_MIN$	Minimum physical value
$PHYS_MAX$	Maximum physical value
$X_DIMENSION$	Number of X sampling points for axes, characteristic curves and maps
$Y_DIMENSION$	Number of Y sampling points for maps
$DATATYPE_C$	Data type in C syntax
$DATATYPE$	Data type in general formulation
$VARIANT$	Name of the variant for variant-coded characteristic parameters
$VALUE_LIST$	Value or value list, separated with commas
$MEMORY_DUMP$	Memory map (list of bytes, separated with commas)
$INDEX$	Consecutive index
X_AXIS_NAME	Name of the X axis
$X_AXIS_COMMENT$	Comment about the X axis
X_AXIS_UNIT	Physical unit for the X axis
$X_AXIS_PHYS_MIN$	Minimum physical value of the X axis
$X_AXIS_PHYS_MAX$	Maximum physical value of the X axis
$X_AXIS_DATATYPE_C$	Data type of the X axis in C syntax
$X_AXIS_DATATYPE$	X axis data type in general formulation
$X_AXIS_SOURCE_NAME$	Name of the input parameter for the X axis

$X_AXIS_SOURCE_COMMENT$	Comment about the input parameter for the X axis
$X_AXIS_VALUE_LIST$	Value list for the X axis
Y_AXIS_NAME	Name of the Y axis
$Y_AXIS_COMMENT$	Comment about the Y axis
Y_AXIS_UNIT	Physical unit for the Y axis
$Y_AXIS_PHYS_MIN$	Minimum physical value of the Y axis
$Y_AXIS_PHYS_MAX$	Maximum physical value of the Y axis
$Y_AXIS_DATATYPE_C$	Data type of the Y axis in C syntax
$Y_AXIS_DATATYPE$	Y axis data type in general formulation
$Y_AXIS_SOURCE_NAME$	Name of the input parameter for the Y axis
$Y_AXIS_SOURCE_COMMENT$	Comment about the input parameter for the Y axis
$Y_AXIS_VALUE_LIST$	Value list for the Y axis
$LINKER_NAME$	Name of the associated object from the Linker MAP file

Infos
Only those macros must be specified in the template files which are to be exported. If a macro cannot be replaced (e.g. in the Y_AXIS_NAME when working with a characteristic curve), then it remains unchanged.

In the directory `\ReportFormat` of the installation are some template files that can be copied into a new project and edited as needed.

During export the appropriate template file is determined for every database object to be exported and appended at the end in the target file after the macros are replaced. To do this, the name of the template file is determined as follows: When a record layout has been assigned to the database object in the database, the name of this record layout is used as the template name with the file extension ".TEMPL". If there is no record layout for this database object, the name is determined according to the variable type:

Example
Only those names and comments about parameters and measurement values from the database should be exported. There is a file in the working directory for this purpose with the name "Value.TEMPL" with the following contents:

Name: `$NAME$,Comment: $COMMENT$`

Info
Only those data within the $ sign are evaluated. All other strings are accepted, as well as formattings such as e.g. line breaks or tabs.

Then either parameters or measurement values can be selected and highlighted via the Search function (see Section 4.3.1), or the entire database is exported (in this case a message appears saying that the template files are missing for the other database objects, such as characteristic curves etc.). After "CANape Report" has been selected as the export format and a file name has been entered with whatever extension you wish, the corresponding file is created. Here are some example contents:

```
Name: BitSlice, Comment: Test signal: 4 Bits on a byte boundary
Name: BitSlice0, Comment: Test signal: 5 Bits
Name: BitSlice1, Comment: Test signal: 5 Bits
Name: BitSlice2, Comment: Test signal: 5 Bits
Name: Counter_B4, Comment: Single bit demo signal (bit from a byte shifting)
Name: Counter_B5, Comment: Single bit demo signal (bit from a byte shifting)
Name: Counter_B6, Comment: Single bit demo signal (bit from a byte shifting)
Name: Counter_B7, Comment: Single bit demo signal (bit from a byte shifting)
```

It is also possible to select individual database objects and export them as a new database (see Figure 3). Groups (folders) and individual database objects can be selected. The marked database objects can be exported with either the **Export** function from the popup menu or via **File | Export**.

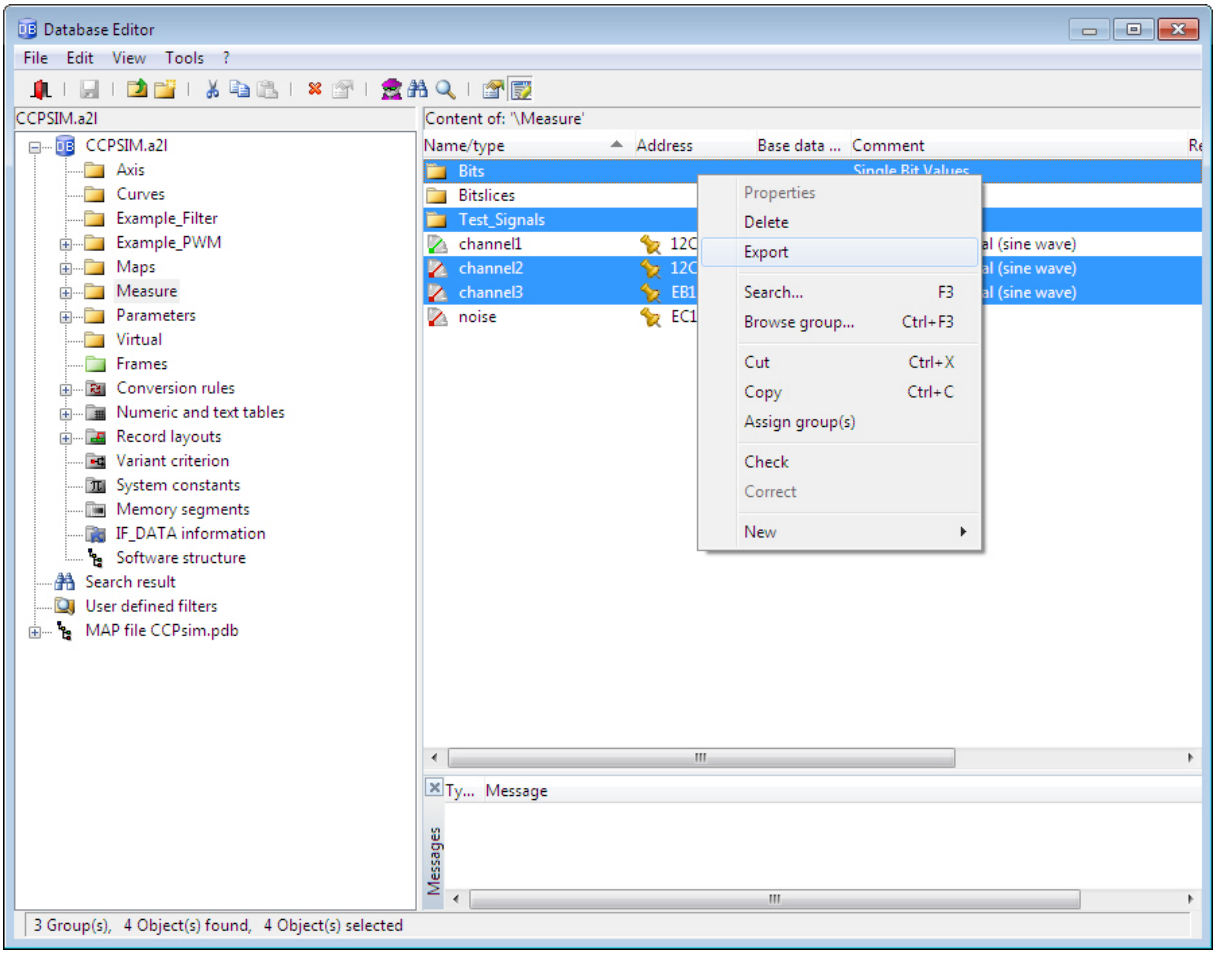

Figure 3: Exporting Groups and Database Objects

There are several options for export (see Figure 4):

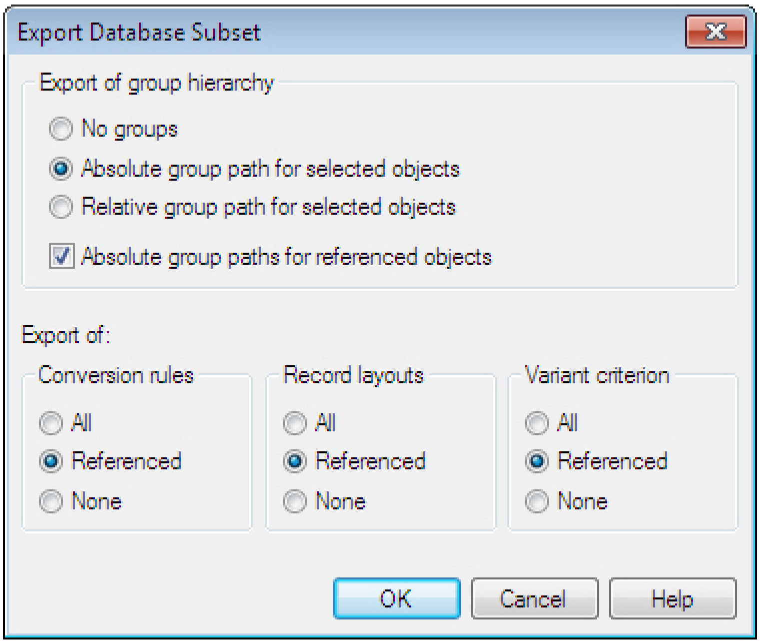

Figure 4: Export Options

Info
The default settings can be kept for most applications.

All options in the dialogs **ASAP2 Export** and **Export database** subset are thoroughly described in the Online Help.

4 The Database Editor

In Figure 5 you can see the ASAP2 Editor with a sample ECU database

The ASAP2 Editor is divided into 4 sub windows; the left window shows a tree representation of all the groups in the database, and the right window all the database elements (objects) of the selected group. Symbols differentiate between the database objects such as groups, characteristic curves and maps, axes etc. (see table below). If there is a pin  in front of the address of a database object, this means that the object was inserted into the database via the Linker MAP file.

If the mouse arrow is held over an entry in the table, additional information appears about the corresponding element. The following table shows all symbols used for the objects in the ASAP2 Editor:

Symbol	Description
	Measurement value (cannot be calibrated)
	Measurement value in Read/Write mode (can be calibrated)
	Parameter
	Parameter in ReadOnly mode (cannot be calibrated)
	Virtual parameter (only exists in the database and thus has no address range in the ECU)
	Dependent parameter (can only be calibrated as an input variable via one or more other parameters)
	Virtual dependent parameter (can only be calibrated as an input variable via one or more other parameters, only exists in the database and has no address range in the ECU)
	Shared Axis
	Shared axis in ReadOnly mode (cannot be calibrated)
	Characteristic curve
	Characteristic curve in ReadOnly mode (cannot be calibrated)
	Map
	Map in ReadOnly mode (cannot be calibrated)
	Measurement value block (cannot be calibrated)
	Measurement value block in Read/Write mode (can be calibrated)

Symbol	Description
	Calibration value block
	Calibration value block in Read/Only mode (cannot be calibrated)
	Cuboid. Cuboids are not supported at the moment and are marked as invalid ( ).
	Virtual signall; Virtual signals can be calculated from real measurement values via an algebraic formula.
ABC	ASCII string
ABC	ASCII string in ReadOnly mode (cannot be calibrated)
1:1	1:1 conversion
	Linear formula
	Rational formula
f(x)	Algebraic formula
	Conversion table
	Symbolic conversion table
	Numeric table
	Text table
	Record layout
	Variant criterion
π	System constant
	Memory segment
	IF_DATA information

More details about the Database Editor can be found in the Online Help.

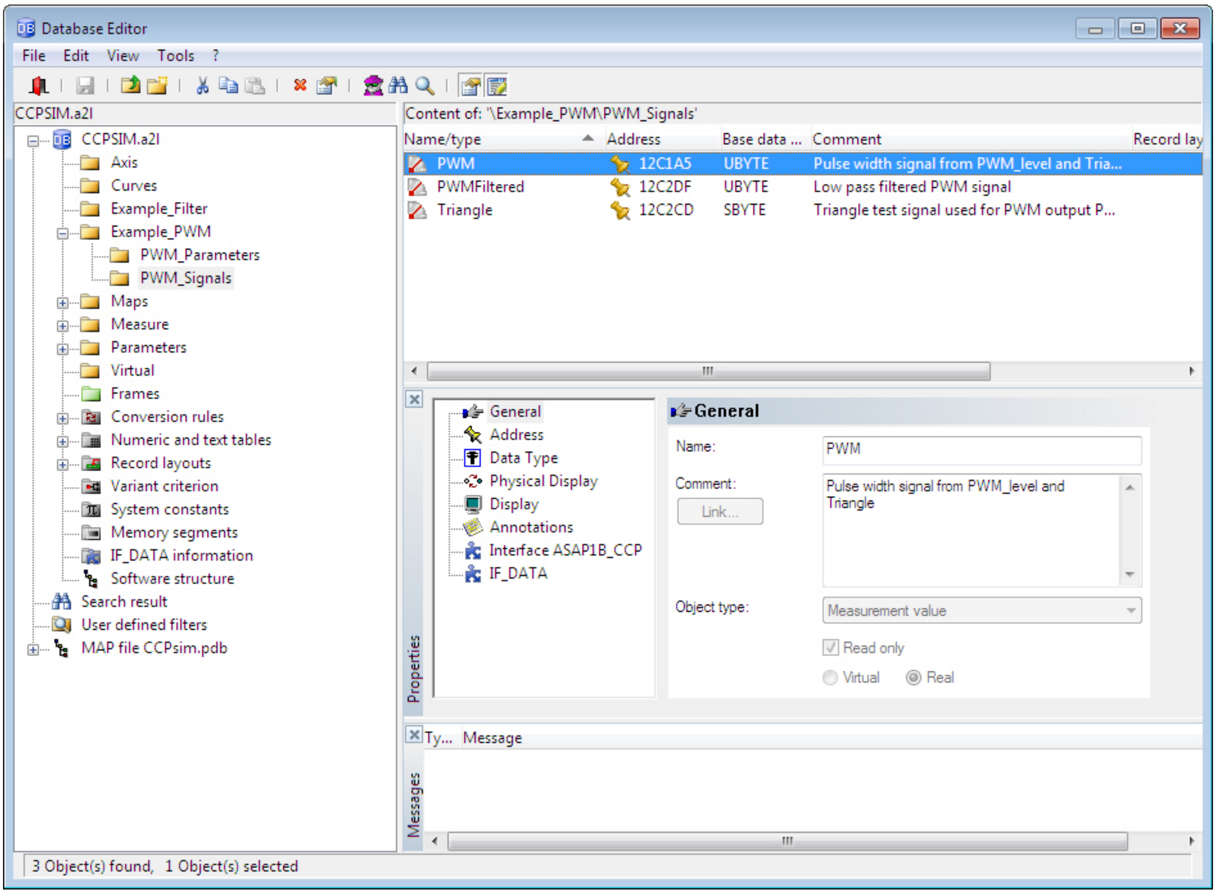

Figure 5: Database Editor with Controller Database

4.1 Creating a Controller Database

The following will show how to create a new controller database appropriate for a specific controller.

Info
The options of the single dialog pages to create database objects are explained in detail in the online help.

The database may contain the following types of objects

- Groups (on page 21)
- Measurement values (on page 25)
- Parameters (on page 32)
- Dependent parameters (on page 38)
- Shared axes (on page 40)
- Characteristic maps (on page 42)
- Characteristic curves (on page 46)
- Value blocks (on page 48)

- Virtual signals (on page 49)
- ASCII strings (on page 50)
- Conversion rules (on page 51)
- Record layouts (on page 59)
- Variant criteria (on page 61)
- System constants (on page 66)
- Memory segments (on page 67)
- Frames (on page 69)
- Partial databases (on page 71)
- IF_DATA information (on page 71)
- Software structures (on page 73)

These objects can be hierarchically arranged in groups for structuring.

Info
Groups can only be defined for quantities to be measured and calibrated.

4.1.1 Assignment of a Linker MAP File

A Linker MAP file can be used to create an ECU database during program development. New entries in the ECU database can be added easily with the help of the information contained in the Linker MAP file. If the addresses of existing entries shift due to program changes, the ASAP2 Editor updates all addresses and data type information in the ECU database automatically using the Linker MAP file.

4.1.2 Groups in the Database

The logical connection between individual database objects can be shown by assigning database objects to individual groups. In doing so, the user can freely define the groups and subgroups and in assigning the individual database objects to the groups.

Click the symbol on the toolbar or choose the menu item **Edit | New | Group** to open the first page of the **Group** dialog (see Figure 6). Enter the name of the new group to be created and a short comment describing the group here.

System groups are identified by the option **Protected**. This setting only has an effect if the ECU database was loaded in encrypted form. In this case, the entire database is write-protected, i.e. the objects to be measured and calibrated cannot be changed in the Database Editor. Protected groups are not displayed.

Info
Protected groups are only possible for *.db files, not for *.a2l files.

A new group can also be created via the popup menu. Click the corresponding group in the right or left sub-window with the right mouse button, select the menu item **New** and then the option **Group**.

The group can be defined as a group of functions with the **Function group** check-box. Database objects in function groups can also be classified according to their use.

Info
If a group is converted into a function group, the measurement variables receive the status of "local" and the calibration variables the status of "referenced".

The new group is created as a subgroup in the currently active group. If a new group is to be created in the database root directory, click the database name in the left window and open with the right mouse button via the popup menu.

Info
No space characters or special characters are permitted when assigning the group name and object names. Each name may only be assigned once.

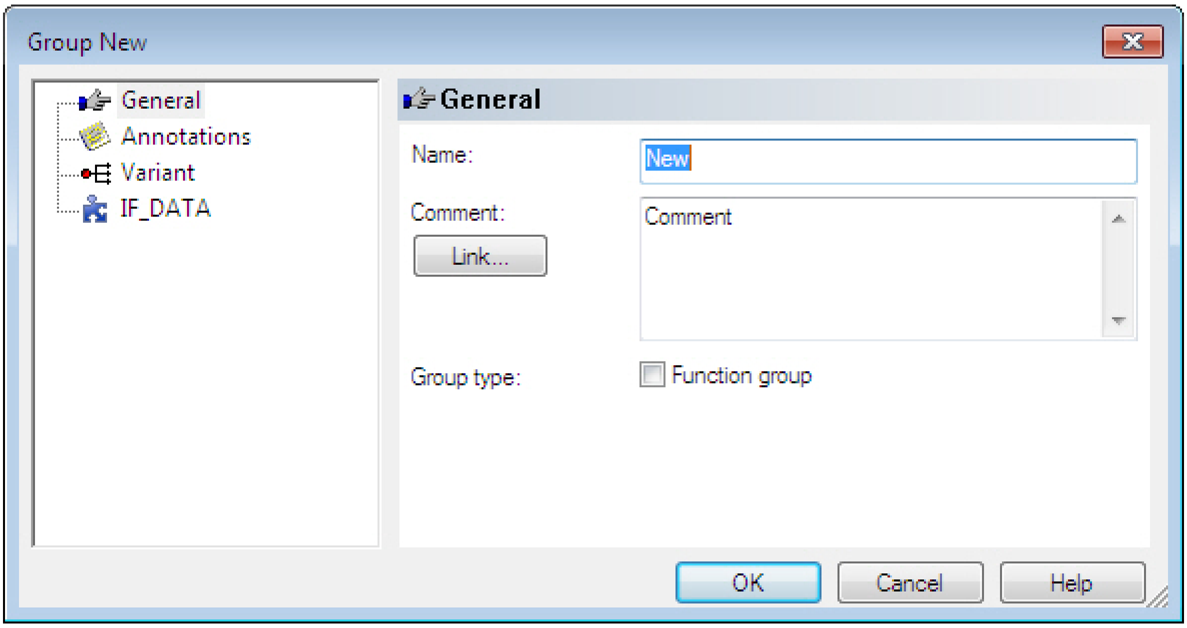

Figure 6: Dialog page "Group - General"

Additional information (ASAP ANNOTATIONS) about the group can be input on the next dialog page (see Figure 7).

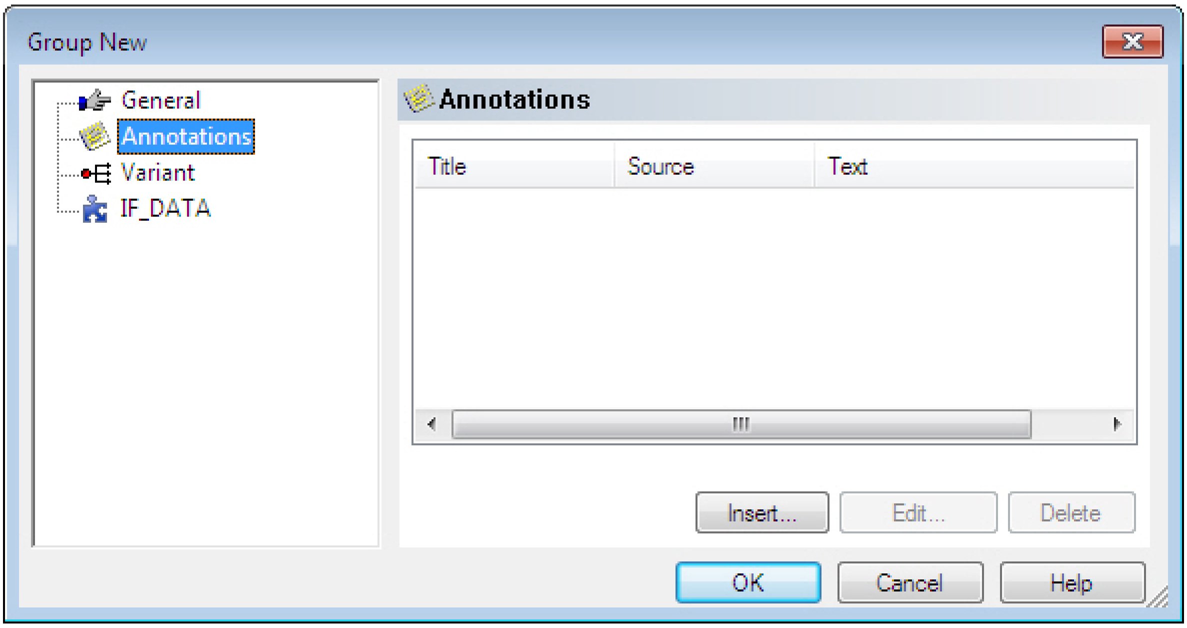

Figure 7: Dialog page "Group - Annotations"

The comments can only be seen in this dialog. To edit an existing comment, double click on the particular entry or click the **[Edit]** button. To create a new entry, click the **[Insert]** button. These actions all open a dialog to edit the comment. This dialog is described in detail in the Online Help.

A variant criterion can be assigned to the group on the next dialog page (see Figure 8), if a variant criterion has been defined in the database (see section 4.1.14, "Variant Criteria").

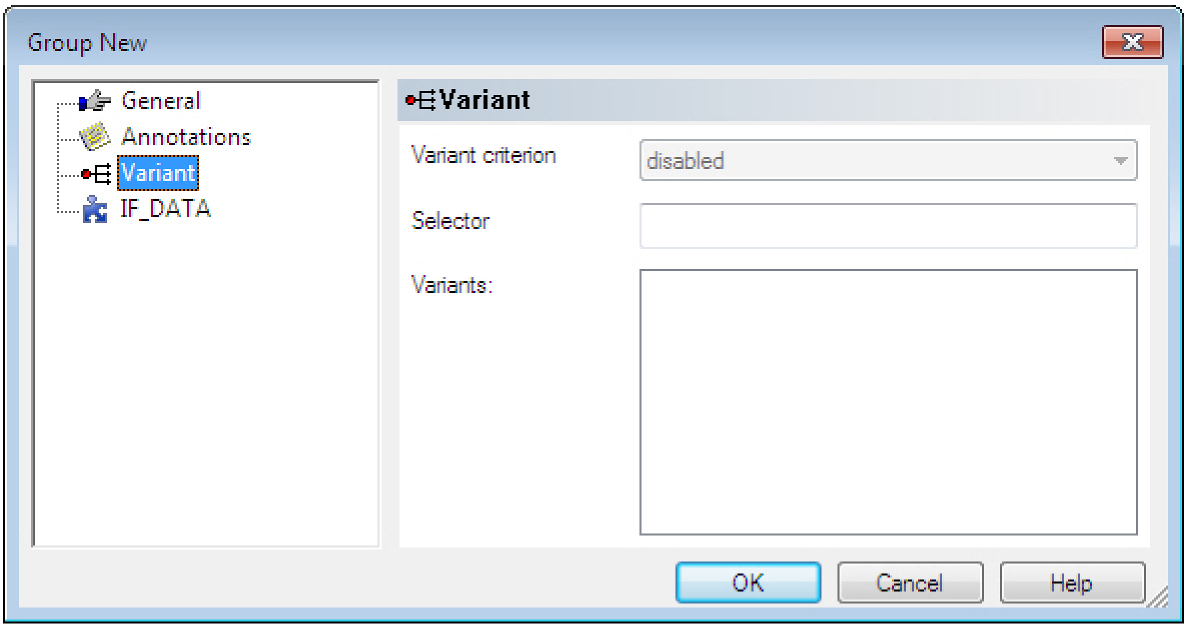

Figure 8: Dialog page "Group- Variant"

On the last dialog page (see Figure 9) all IF_DATA information of the group are displayed. The button **[Insert]** allows the user to add further IF_DATA sections. A double click on the entry or the button **[Edit]** opens a dialog page to edit the respective entry (see Figure 10).

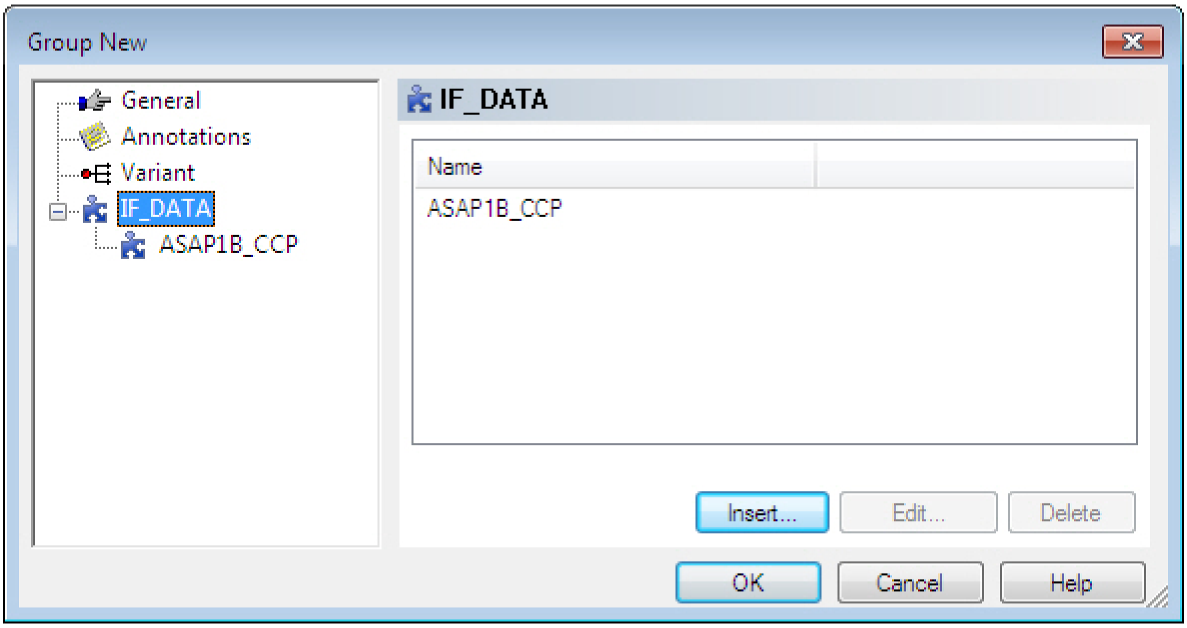

Figure 9: Dialog page “Group – IF_DATA"

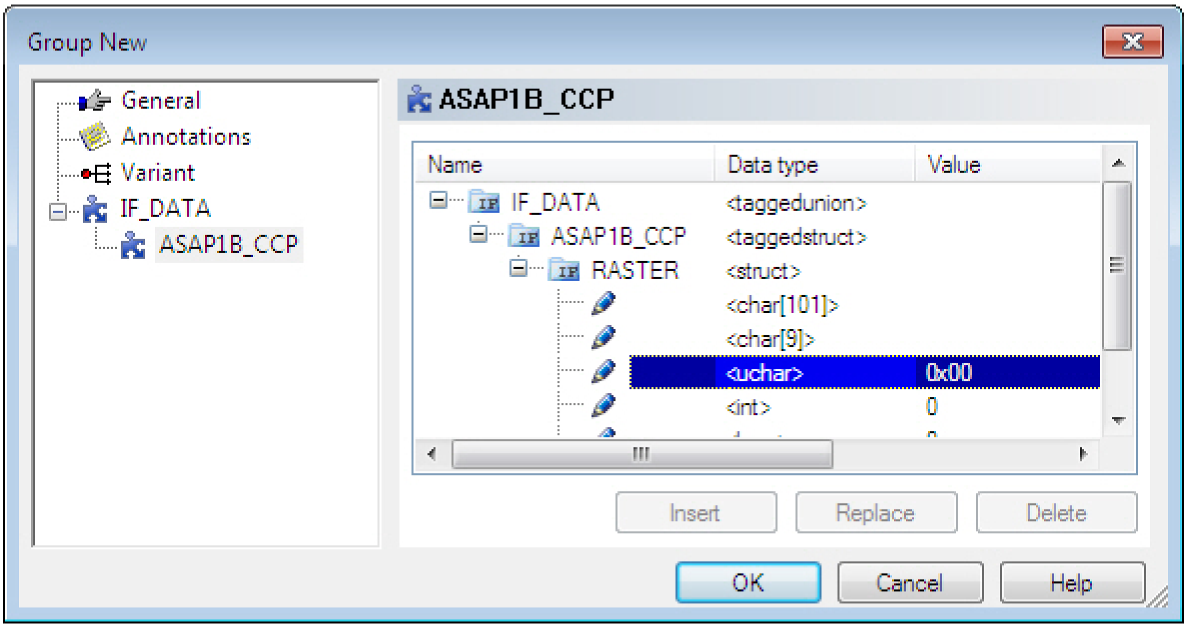

Figure 10: Edit IF_DATA entry

There are no limitations on the number and type of measurement or calibration variables that may be added to a group. Consequently, groups may be created for:

- Measurement objects
- Parameter or calibration objects
- Virtual signals

All measurement and calibration objects which logically belong to the same subsystem of the control device can also be inserted into a group.

Additional subgroups may be inserted in a group.

Individual objects and groups together with the objects contained in them can be relocated or copied within the database structure. This is done either by choosing the menu items **Edit | Copy** or **Edit | Cut** (these commands can also be given by selecting the object and right clicking to the popup menu) or by dragging the corresponding object or group with the left mouse button to its new location. If the <Ctrl> key is pressed while dragging with the mouse, a copy of the group or object is created. The name of the new object/group is composed of the original name with the suffix `_Copyn` (where n is a counter)

Example
`MinChannel_Copy3.`

The properties of an existing group can be changed by again selecting the group and opening the “General” dialog page via the popup menu or via **Edit | Properties** in Figure 6.

4.1.3 Measurement Values

A measurement value can be e.g. a variable of a controller program. The description includes, for example, a name, address, data type and a conversion formula, if required.

In this section it will be shown how to insert a measurement value into the database in the ASAP2 Editor. To add a new measurement value to a group, select the desired group. Select **Edit | New | Measurement value** from the popup menu. This opens the first page of the dialog shown in Figure 11.

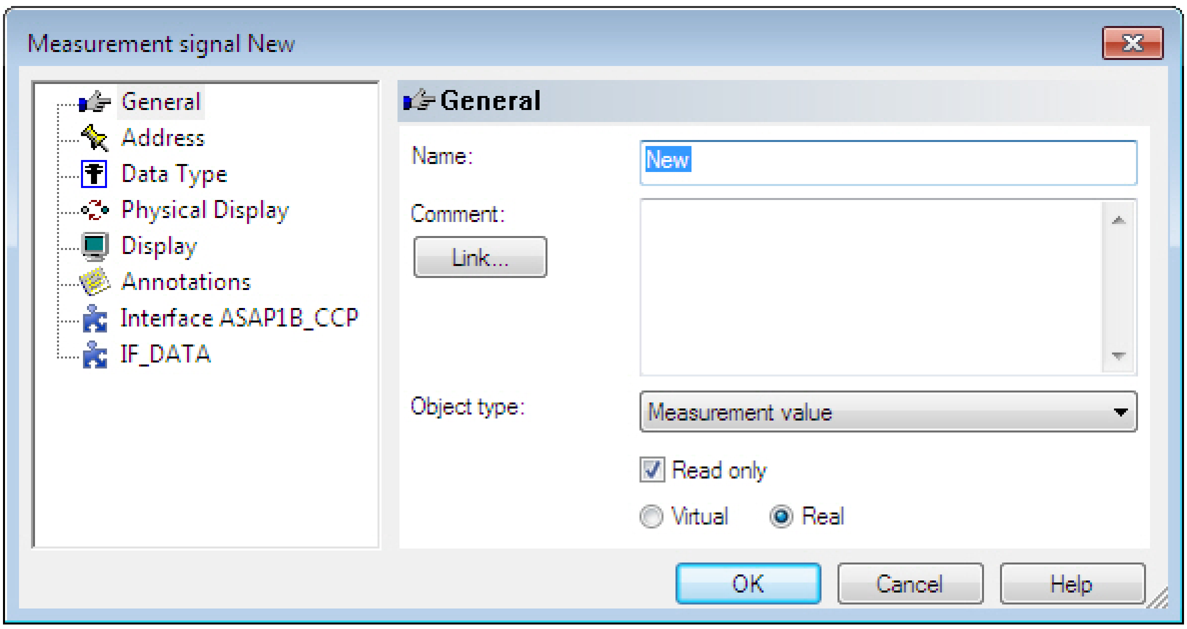

Figure 11: Dialog page "Measurement value - General"

A symbolic name for the measurement value must be entered in the **Name** input box. In the **Comment** input box, you can enter a brief textual description of this measurement value.

The **[Link]** button next to the comment field opens a dialog in which a link to a file on the local hard disk or another database object in the current database can be made. The link appears in the "Comment" column and, when clicked, opens the corresponding file or the properties dialog of the corresponding database object. This dialog page is also provided when multiple objects are selected.
Because we are creating a new measurement value, the **Object type:** field receives the designation "Measurement value". This entry cannot be changed.

In normal mode a measurement value cannot be calibrated (adjusted) in the ECU. When the control field **Read only** is deactivated, however, the measurement value can be enabled for external calibration in the ECU (see Section 4.1.10, "Virtual Measurement Signal").

On the next dialog page (see, Figure 12) is where the address is specified for a real measurement value.

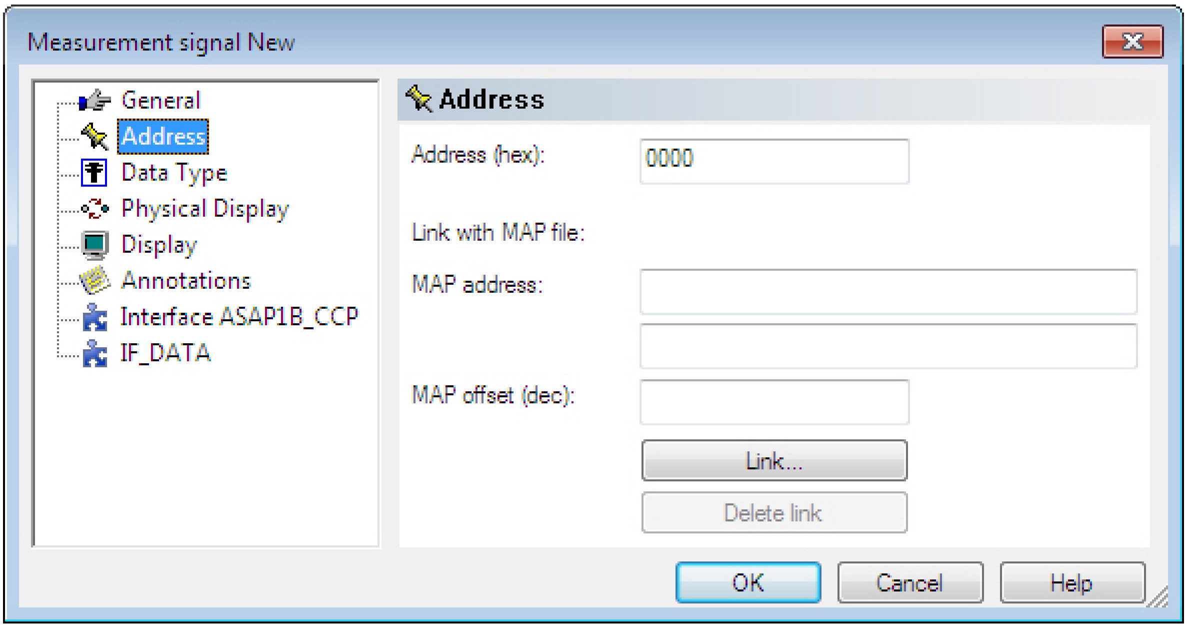

Figure 12: Dialog page "Measurement value - Address"

There are two ways to assign a memory address in the ECU: either enter the address in the **Address** input box manually or press the **[Link]** button and select a symbolic name from the list in the Linker MAP file. If you select from the list, the address of the measurement object in the MAP file is assigned to the symbol. When the address is input from the Linker MAP file, a value can be entered in the **Offset (dec)** box.

Assignment from the Linker MAP file has a significant advantage: The address and data type information of the ECU database can be automatically updated with a new Linker MAP file via the menu item **File | Update addresses**.

The data type used is selected on the next dialog page (see Figure 13).

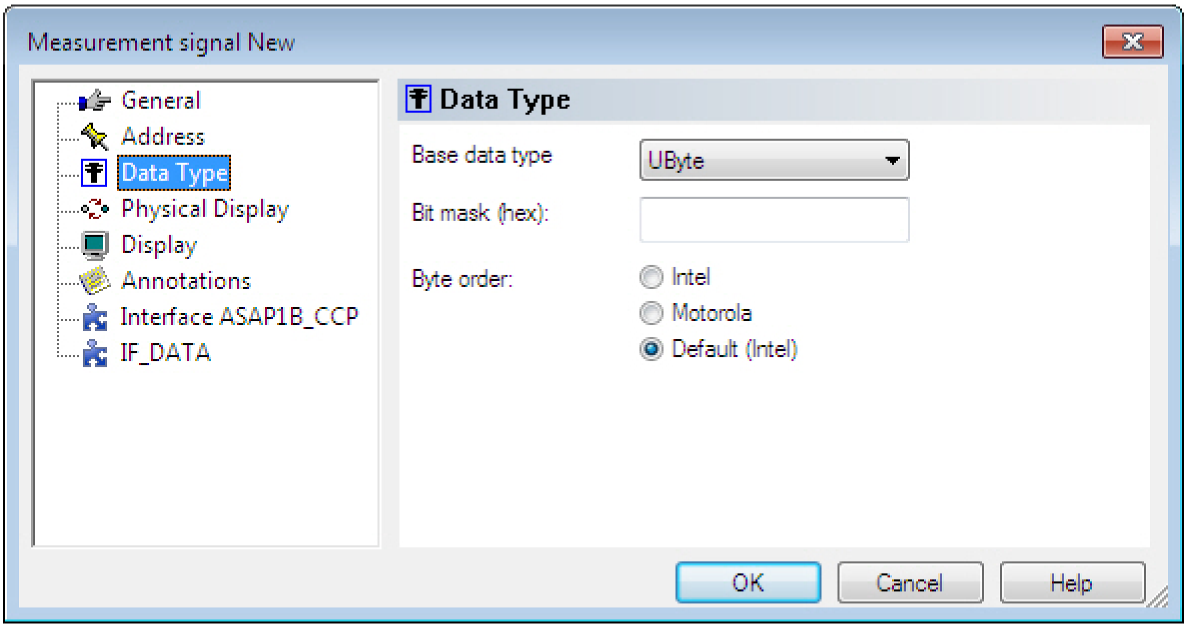

Figure 13: Dialog page "Measurement value - Data type"

In general, the following data types are available in the Editor:

- Float
- Double
- Signed
- Unsigned

For integer types you can input the number of bits (1 to 32) and a bit offset (e.g. to identify the bit position within a byte for binary measurement objects such as a limit switch).

The byte order can be set individually for every newly created measurement value. When the **Default** option is activated, the byte order that was set in the database **Tools | Settings database/MAP file** dialog is used.

The next dialog page (see Figure 14) is where the physical display of the measurement value is specified.

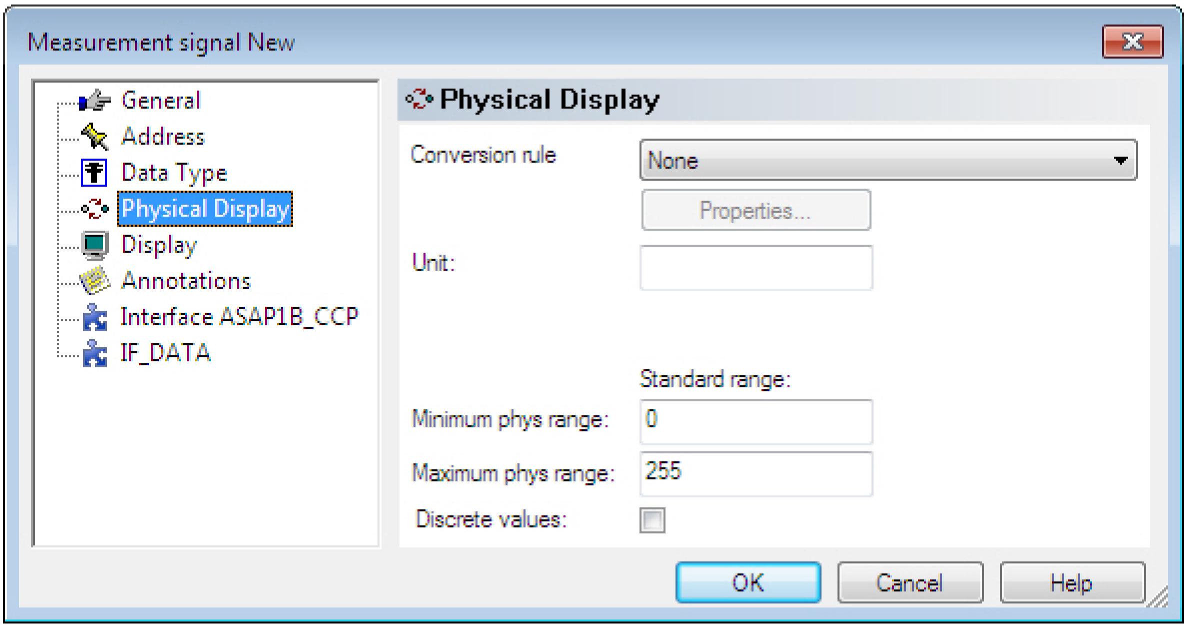

Figure 14: Dialog page " Measurement value – Physical Display"

You can assign a conversion rule to each measurement value to convert the ECU internal value into a physical value. To do this, a globally defined conversion can be selected from the database (these are marked with an * in the **Conversion rule** selection list) or an individual conversion rule must be defined for that measurement object. To define an individual conversion rule, a conversion type (all entries without an *) must be chosen from the **Conversion rule** selection list. Pressing the **[Properties]** button opens a dialog in which the respective conversion rule can be defined. The conversion rules are explained in detail in the online help of this dialog page.

In the **Unit** input box, you can enter the physical units of the object (e.g. Volts, Hz, etc.) when a local conversion formula is defined. Inputting a physical unit does not affect the conversion method.

Set your limits for the physical value range in the input fields **Minimum phys. value** and **Maximum phys. value** in the **Standard range** area of the dialog.

Info
Limiting the value range here has no influence on the measurement. A measurement even continues when the measurement value exceeds or falls below these limits.

When the option **Discrete values** is activated no interpolation between measured values is allowed (e.g. when the measurement course is displayed in the Graphics window).

How the measurement value will be displayed is configured on the next dialog page (see Figure 15).

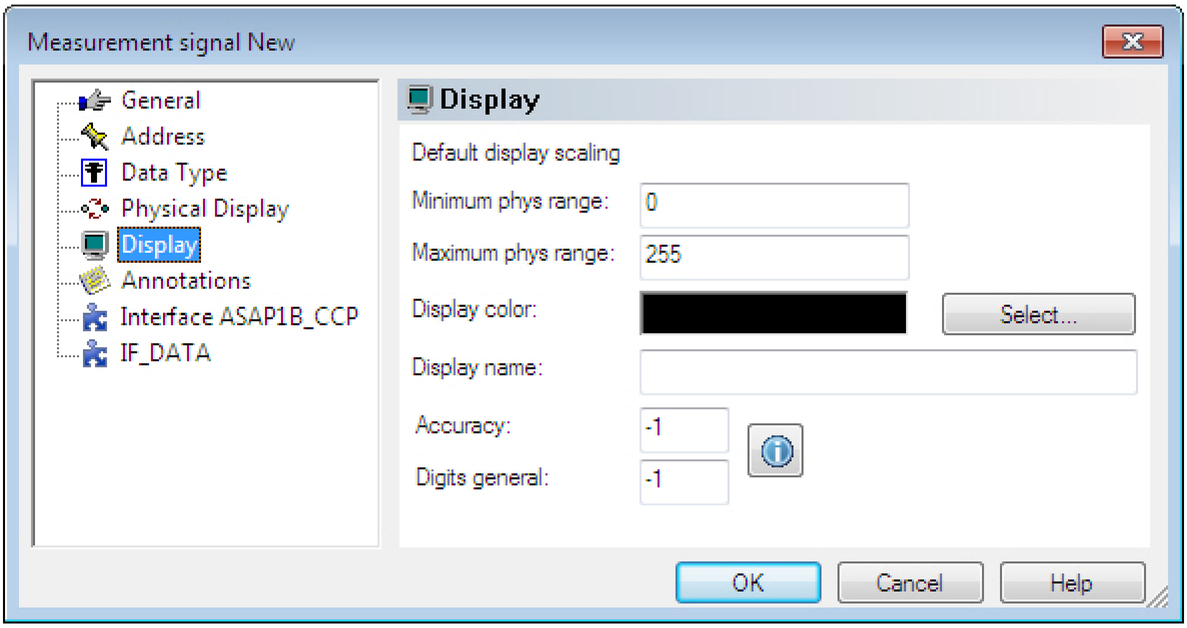

Figure 15: Dialog page " Measurement page - Display"

The upper and lower limits displayed when the measurement value is added to a display window can be entered in the **Minimum** and **Maximum** input fields in the **Default Display Scaling** section of this dialog. If a name is entered in the **Display name** input field, this name will be displayed instead of the variable name in all display windows when the option **Use display names** is activated.

The button  opens a help page that explains in detail the settings "Accuracy" and "Digits general".

On the next dialog page (see Figure 16) you can enter additional information (ASAP ANNOTATIONS) about the measurement value.

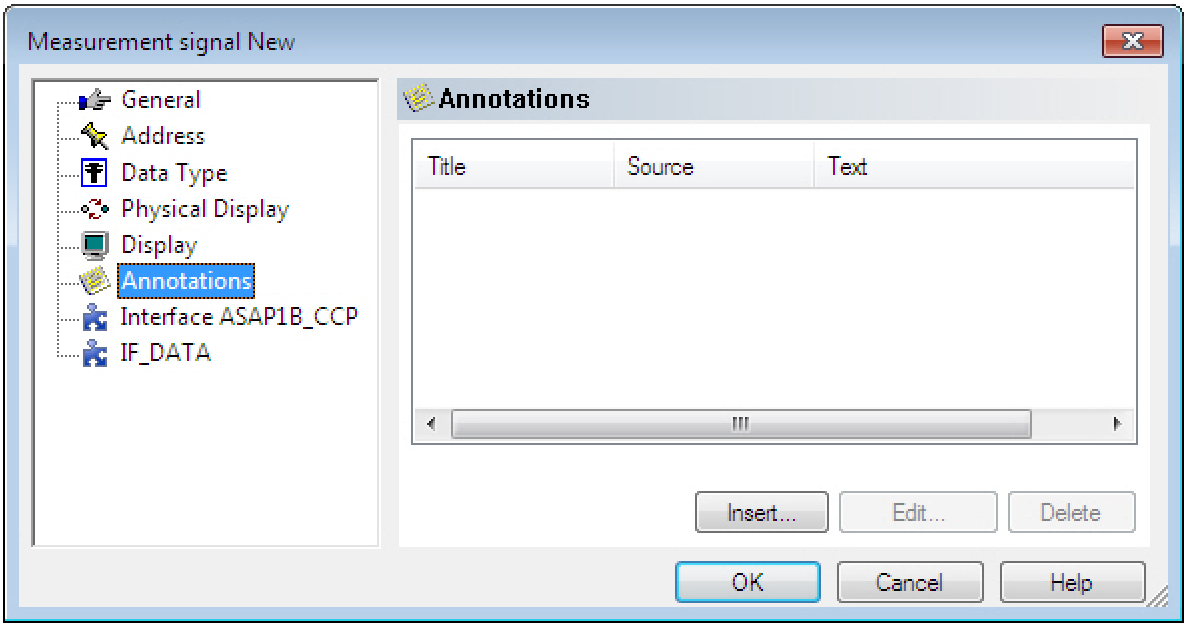

Figure 16: Dialog page " Measurement value - Annotations"

Comments in this window can only be viewed. To edit an existing comment, double click on the particular entry or click the **[Edit]** button. To create a new entry, click the **[Insert]** button. These actions all open a dialog to edit the comment. This dialog is described in detail in the Online Help.

The next dialog page (see Figure 17) displays information about the address of the CCP database object from the ASAP file.

Info
This page in only displayed when an ASAP file is already loaded.

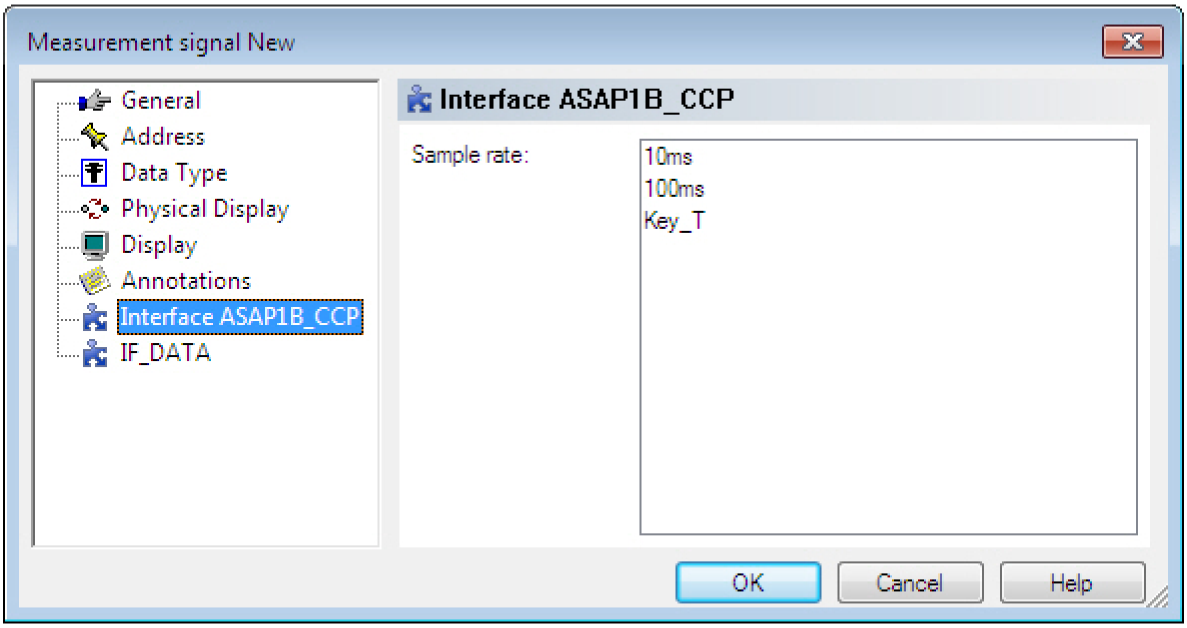

Figure 17: Dialog page "Measurement value - CCP Driver Interface"

This data is only for information. The values cannot be changed.

The dialog page lists the sampling rates for measurement values which were input in the Driver Settings dialog. If you make a selection from this list, and add the measurement value to the Measurement Signal List in the Edit Measurement List dialog, the value appears in the Measuring Mode column. A value can be selected, but set values cannot be changed.

In case of an ASAP file describing XCP database objects, another selection appears in the tree view at the left. This selection opens a dialog seen in Figure 18**Fehler! Verweisquelle konnte nicht gefunden werden.**.

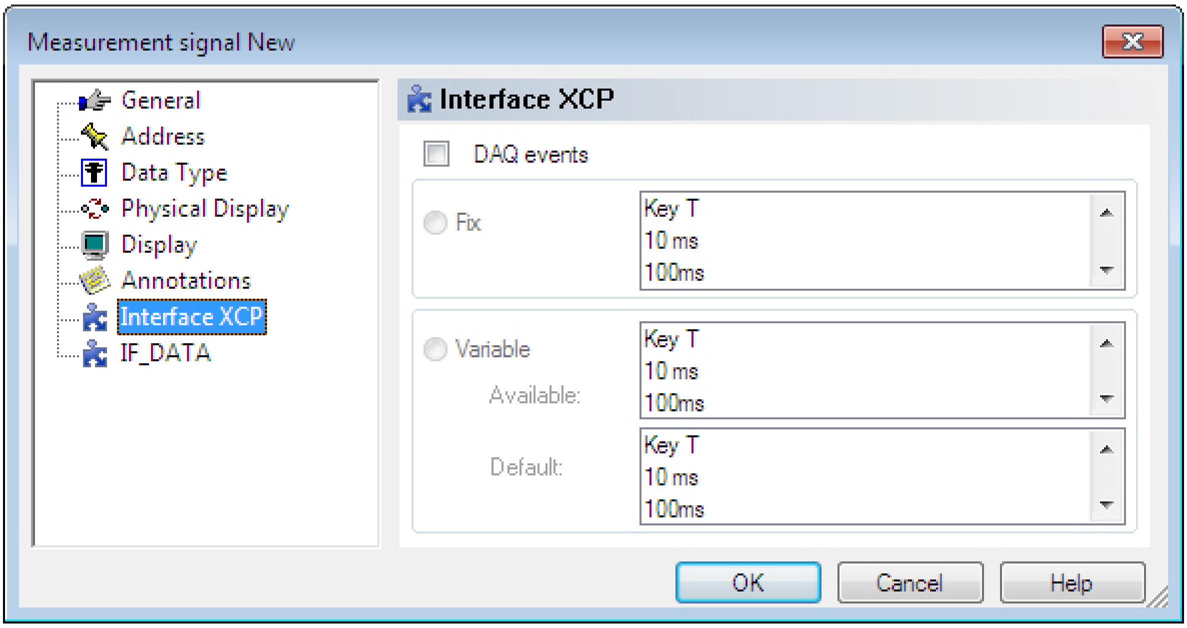

Figure 18 Dialog page "Measurement value - XCP driver interface"

Here, partial lists can be created from the list of all events in the XCP driver dialog. A detailed explanation of the settings can be found in the online help.

On the next dialog page (see Figure 19) all IF_DATA sections of the respective memory segment are displayed. Via the button **[Insert]** additional IF_DATA section can be added. A double click on the entry opens a dialog page to edit it in case it is of an editable data type. The data types of IF_DATA sections are explained in detail in the online help.

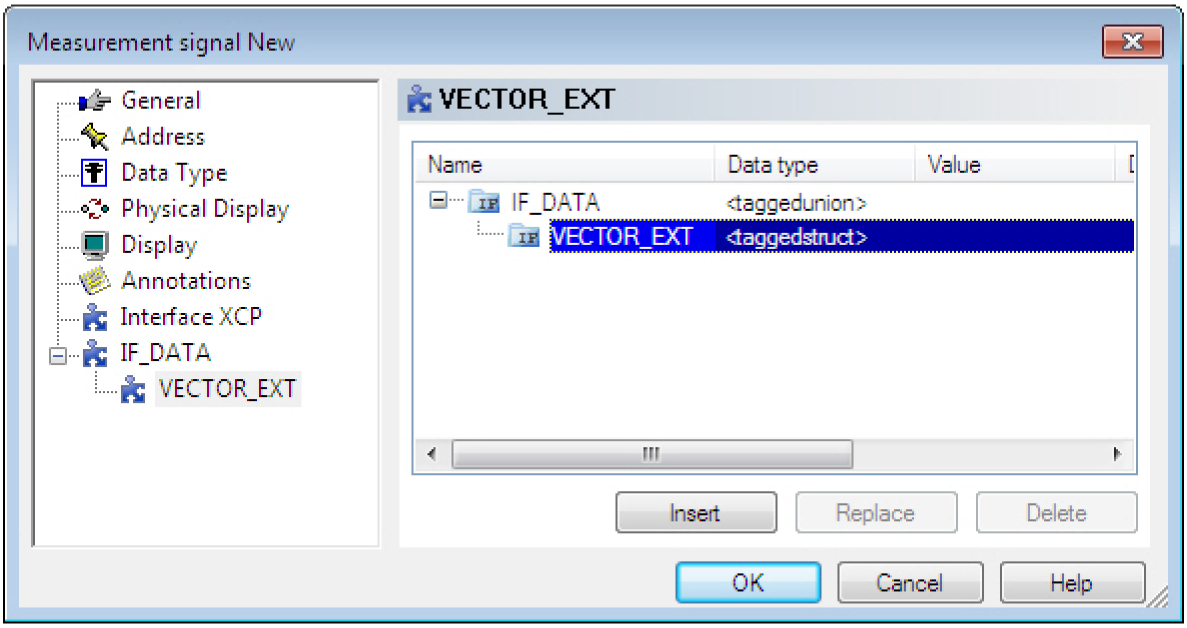

Figure 19: Dialog page "Measurement value – IF_DATA"

The properties of an existing measurement value can be changed by selecting it and reopening the dialog page "General" (Figure 11) via the popup menu, via **Edit | Properties** or by double-clicking it.

A measurement object can be converted to another data type (parameter, map, characteristic curve or value block) via the popup menu. The new object keeps its name and receives the appropriate new symbol. To calibrate the object, **Read only** must be deactivated.

Please refer to the online help for more detailed information on each of these dialogs.

4.1.4 Parameters

A parameter is an adjustable value called a calibration object, i.e. it can be put into a Calibration window and the values assigned to it can be changed. A description of it includes e.g. a name, an address, a data type, a conversion rule and specifications about its range of values.

In this section it will be shown how to insert a parameter into the database in the ASAP2 Editor. To add a new parameter to a group, select the desired group. Select **Edit | New | Parameter** or **New | Parameter** from the popup menu. This opens the first page of the dialog shown in Figure 20:

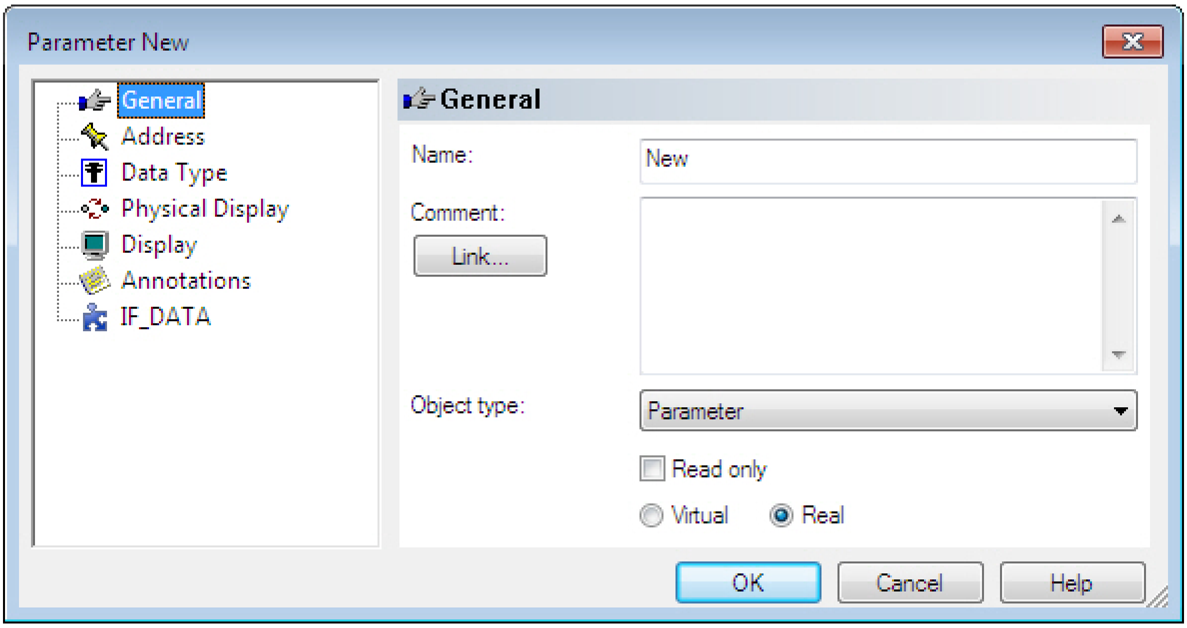

Figure 20: Dialog page "Parameter - General"

A symbolic name must be entered for the parameter in the **Name** input field. In the **Comment** input box you can enter a brief textual description of the parameter.

The field **Object type** receives the designation "Parameter". This entry cannot be changed.

You can prevent the calibration parameter from being adjusted by checking the control field **Read only**. The values assigned to the parameter can then no longer be changed, but they can still be read. A disabled parameter can be identified in the Editor window by a little red cross in the symbol.

The radio buttons **Virtual** and **Real** determine how the parameter will be stored in the ECU. If you select **Real** the value occupies a real address in the ECU, which must be specified on the next page (Address). If the setting **Virtual** is selected, the parameter only exists during run time. A conversion calculation formula must accordingly be input on the **Conversion** page. The parameter thus corresponds to the type **virtual parameter**.

A virtual parameter has an initial value, but no memory area for storage. As a result, it cannot be loaded into the controller, but it can be changed e.g. in CANape. Virtual parameters can be utilized as input variables for dependent parameters.

When the parameter has the attribute **Virtual**, this is shown via a corresponding symbol in the database.

If the parameter is real, the address is input on the next dialog page (see Figure 21).

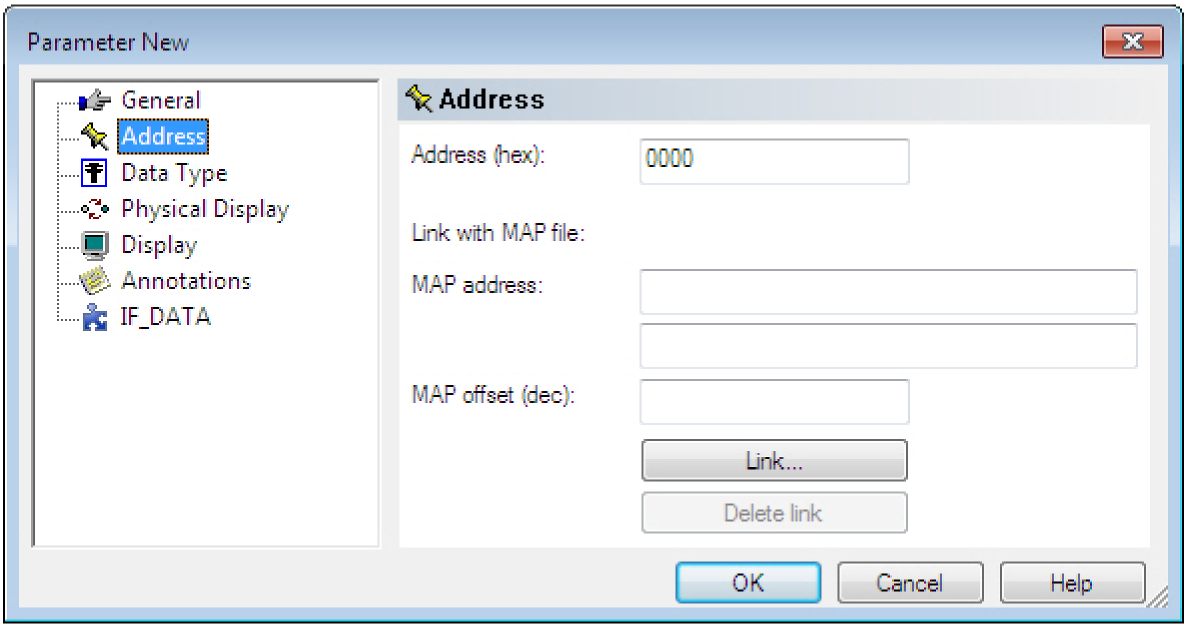

Figure 21: Dialog page "Parameter - Address"

There are two possible ways to assign an ECU memory location: The address can be entered manually in the **Address** input field, or the address can also be selected via a symbolic name from the Linker MAP file of the ECU program. The button **[Link]** opens a corresponding selection list. When the address is input from the Linker MAP file the **Address** input field contains the memory address, and **Offset (dec)** becomes 0.

Assigning an address from the Linker MAP file has a significant advantage: The address and data type information from the ECU database can be automatically updated with a new Linker MAP file via the Database Editor command **File | Update addresses**.

The next dialog page (see Figure 22) is where the data type used is selected.

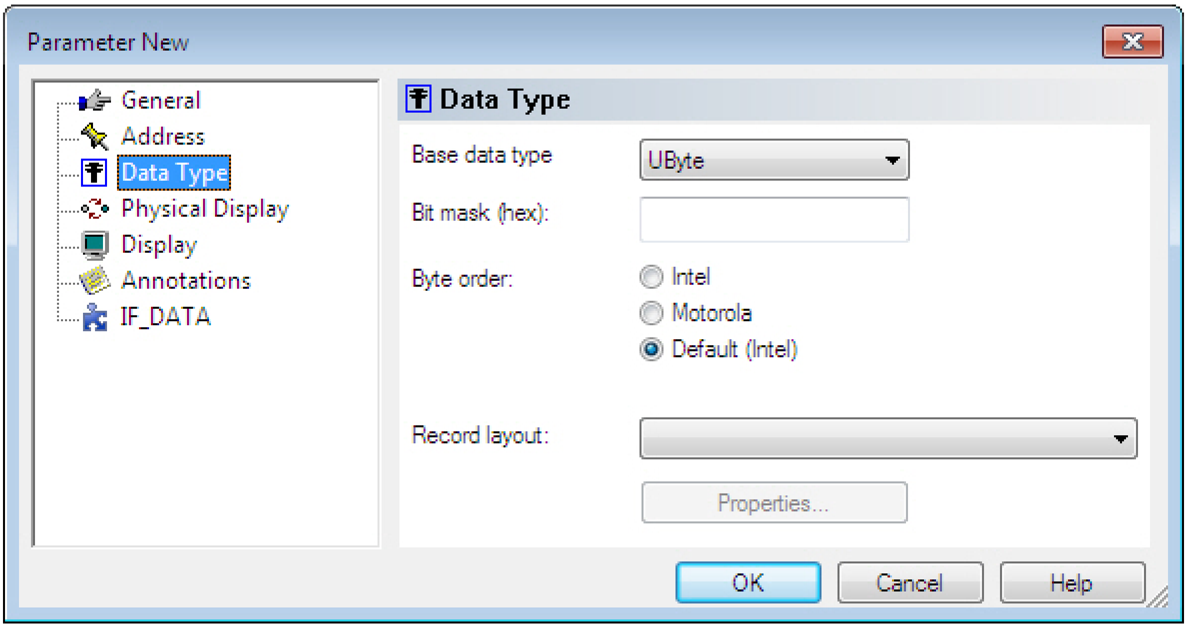

Figure 22: Dialog page "Parameter - Data Type"

A controller program structure saved in the database (called a record layout) or the default standard controller program structure can be selected from the **Record layout** drop-down menu.

Info

If a user-defined record layout is selected for the parameter the **Data type** drop-down list is deactivated, because the data type of the values at the sampling points is defined in the record layout.

These globally defined record layouts can be defined under **Edit | New | Record layout** and are then available for the entire database.

On the next dialog page (see Figure 23) you can set up the physical display of the parameter.

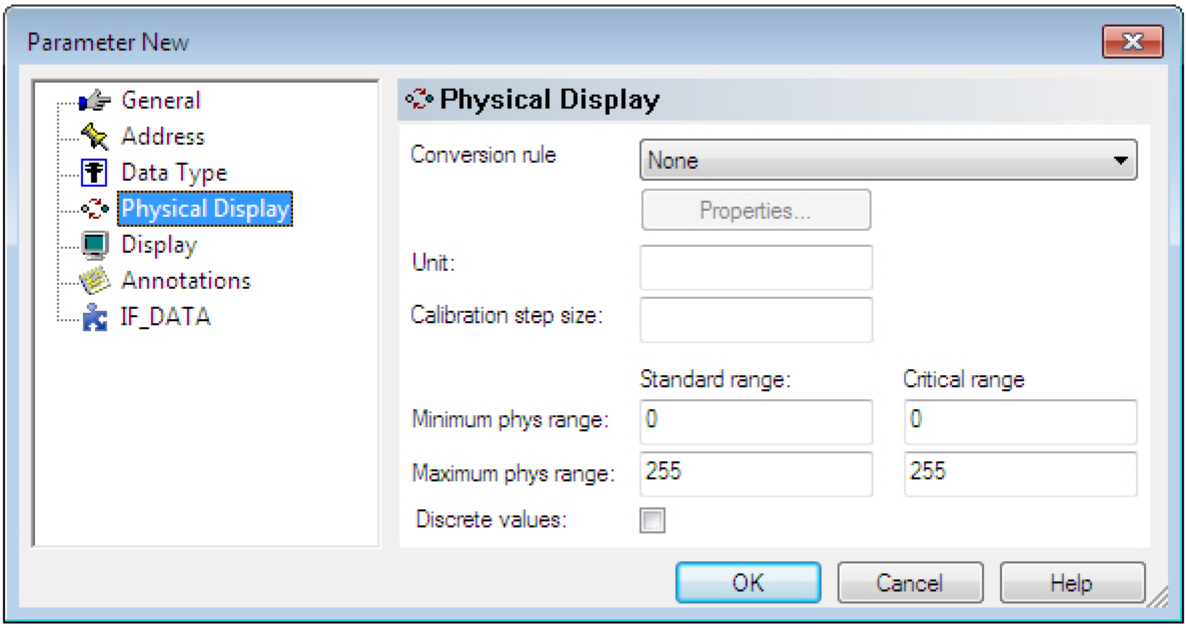

Figure 23: Dialog page "Parameter – Physical Display"

You can assign a conversion rule to each parameter to convert the ECU internal value into a physical value. To do this, either a globally defined conversion rule from the database (these are marked with an * in the selection list) or an individual conversion rule must be defined for that measurement object. To define an individual conversion rule, a conversion type (all entries without an *) must be chosen from the selection list and then the conversion rule must be defined by clicking on the **[Properties]** button. The conversion formula for an already existing parameter can be edited by highlighting the value, right clicking to **Properties**, selecting the conversion type and clicking the **[Conversion]** button.

In the **Unit** input box you can enter the physical units of the object (e.g. volts, Hz, etc.) when a local conversion formula is defined. Input of a physical unit has no influence on the conversion method.

In the **Calibration step size** field you can define a default value for the calibration increment in the CANape Calibration window. Of course you can change this value direct in the calibration window, but this change is not stored in the database. When no value is set here as default 1 is used. The calibration step size is only available for calibration windows, not for group calibration windows.

The range of physical values for the calibration object can be limited via the input fields **Minimum phys range** and **Maximum phys range** in the **Standard range** section of the dialog. The threshold or limiting values for calibrating a parameter are set in the **Critical range** section. The values in the standard range must lie within the critical range of values or match them. If values depart from the standard range of values during calibration of the parameter there will be a query about exceeding the standard range. A parameter cannot be adjusted in such a way that the critical value range is exceeded.

Caution
The parameters for the critical range must be selected in such a way that the ECU cannot be damaged. The values for the critical range must never lie outside the range of values given by the microcontroller manufacturer (see Figure 24). Changes in the **Critical range** fields should only be made by the ECU developers and in compliance with the microcontroller documentation.

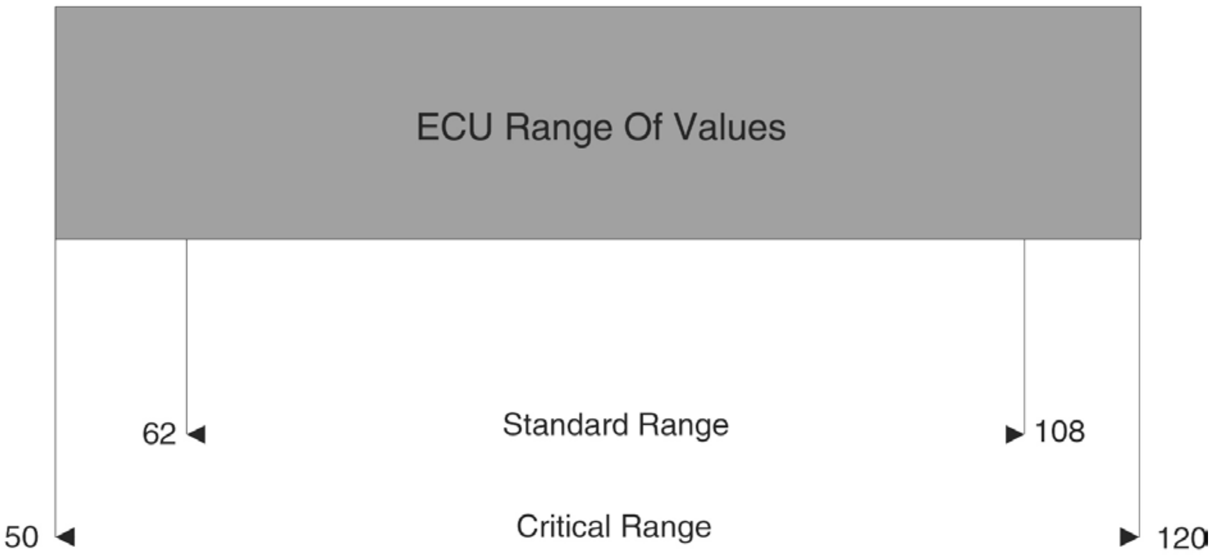

Figure 24: Difference Between Standard and Critical Ranges

The way the parameter will be displayed is configured on the next dialog page (see Figure 25).

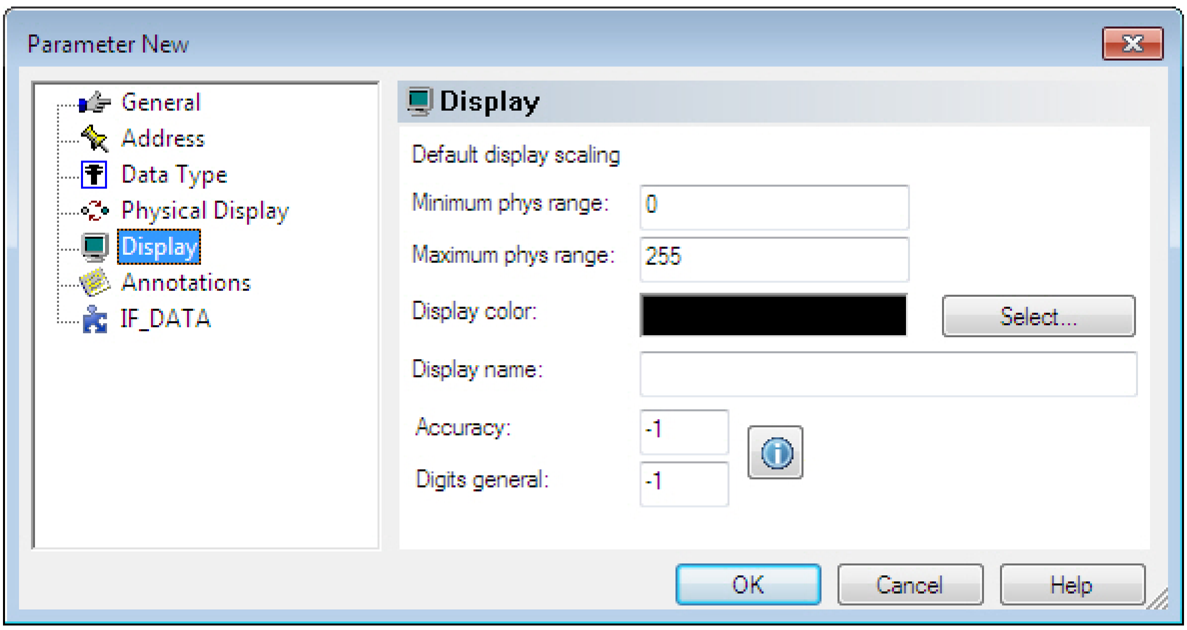

Figure 25: Dialog page "Parameter - Display"

The upper and lower limits displayed when the measurement value is added to a display window can be entered in the **Minimum** and **Maximum** input fields in the **Default Display Scaling** section of this dialog. If a name is entered in the **Display name** input field, this name will be displayed instead of the variable name in all display windows when the option **Use display names** is activated.

On the next dialog page (see Figure 26) you can enter additional information (ASAP ANNOTATIONS) about the parameter.

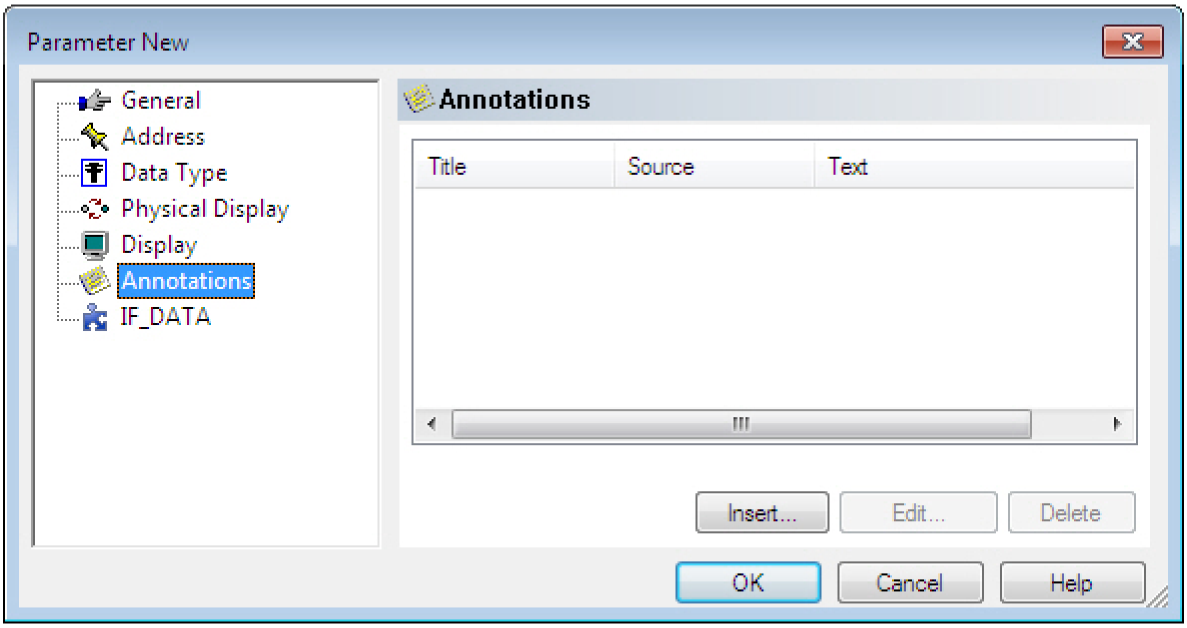

Figure 26: Dialog page "Parameter - Annotations"

Comments in this window can only be viewed To edit an existing comment, double click on the particular entry or click the **[Edit]** button. To create a new entry, click the **[Insert]** button. These actions all open a dialog to edit the comment. This dialog is described in detail in the online help.

On the last dialog page (see Figure 27) all IF_DATA sections of the respective memory segment are displayed. Via the button [Insert] additional IF_DATA section can be added. A double click on the entry opens a dialog page to edit it in case it is of an editable data type. The data types of IF_DATA sections are explained in detail in the online help.

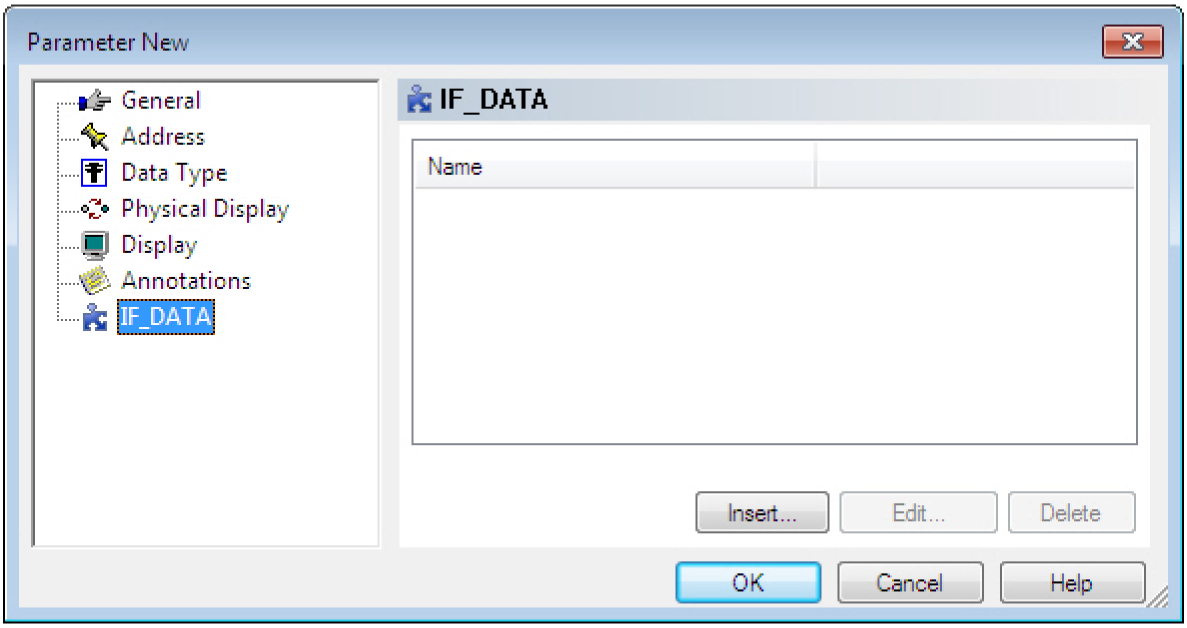

Figure 27: Dialog page "Parameter - CCP driver interface"

This data is only for information. The values cannot be changed.

The properties of an already existing parameter can be changed by highlighting it and reopening the dialog page "General" (Figure 20) by right clicking and selecting **Properties**, the main menu item **Edit | Properties** or by double clicking it.

A parameter can be converted to another data type (measurement value, parameter, map, curve or value block) via the popup menu. The new object keeps its name and receives the appropriate new symbol.

Please refer to the online help function for more detailed information about this dialog.

4.1.5 Dependent Parameter

A parameter is an adjustable value called a calibration object, i.e. it can be put into a Calibration Window and the values assigned to it can be changed. A description of it includes e.g. a name, an address, a data type, a conversion rule and specifications about its range of values.

A dependent parameter needs input values with which its values are calculated. These parameters cannot be changed directly in e.g. CANape, but only via their dependencies.

In this section it will be shown how to insert a parameter into the database in the ASAP2 Editor. To add a new parameter to a group, select the desired group. Select **Edit | New | Dependent Parameter** from the popup menu. This opens the first page of the dialog shown in Figure 28:

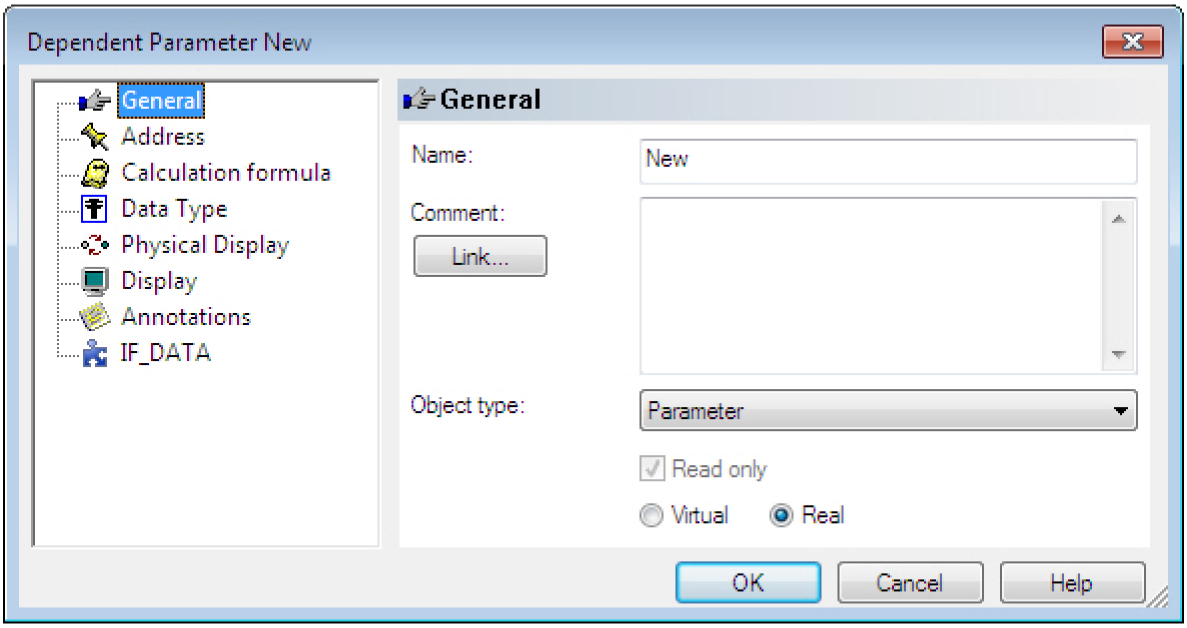

Figure 28: Dialog page "Dependent parameter - General"

The dialog pages are consistent with those of a parameter (see section 4.1.4, Seite 32) so only the dialog page "Calculation formula" (see Figure 29) will be explained here.

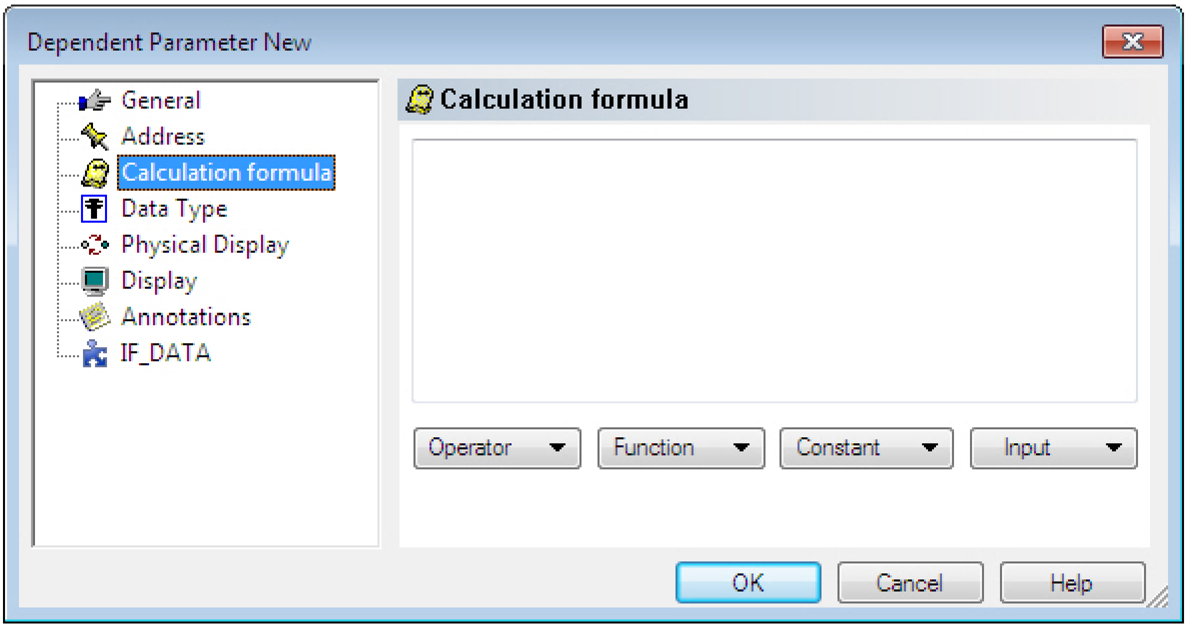

Figure 29: Dialog page Dependent parameter - Calculation formula "

A calculation formula for dependent objects can be created on this dialog page.

Clicking the buttons **[Operator]**, **[Function]**, **[Constant]** or **[Input]** opens popup menus for selecting elements to insert into the formula text in the main window. The text you will enter is an algebraic formula for one or more signals to be measured. In addition to the conventional mathematical functions, Boolean operations can also be used on the signals.

The online help explains all permissible operators, functions, constants and inputs in detail.

A dependent parameter can be converted to another data type (measurement value, parameter, map, curve or value block) via the popup menu. The new object keeps its name and receives the respective new symbol.

Please refer to the online Help function for more detailed information about this dialog.

4.1.6 Shared Axes

A shared axis is one which can be used by multiple characteristic curves / maps.

A shared axis consists, as does a standard axis, of a list of axis point values which is stored in the controller program's memory. Unlike the standard axis, which is always a component of a map and does not have its own address, shared axes are considered to be separate objects (calibration parameters). The most important difference from a standard axis (assigned to exactly one map/curve) is that a shared axis can be assigned to as many maps/curves as you like. If the data describing a shared axis is changed in the Database Editor, this influences how the axis point values of all affected maps/curves are recorded and displayed.

This section will show how a new shared axis is added to the database in the ASAP2 Editor. To add the new shared axis to an existing group, the group must be selected. Then the entry **New | Shared axis** is selected in the pop-up menu or the menu item **Edit | New | Shared axis** is selected. This opens the dialog shown in Figure 30.

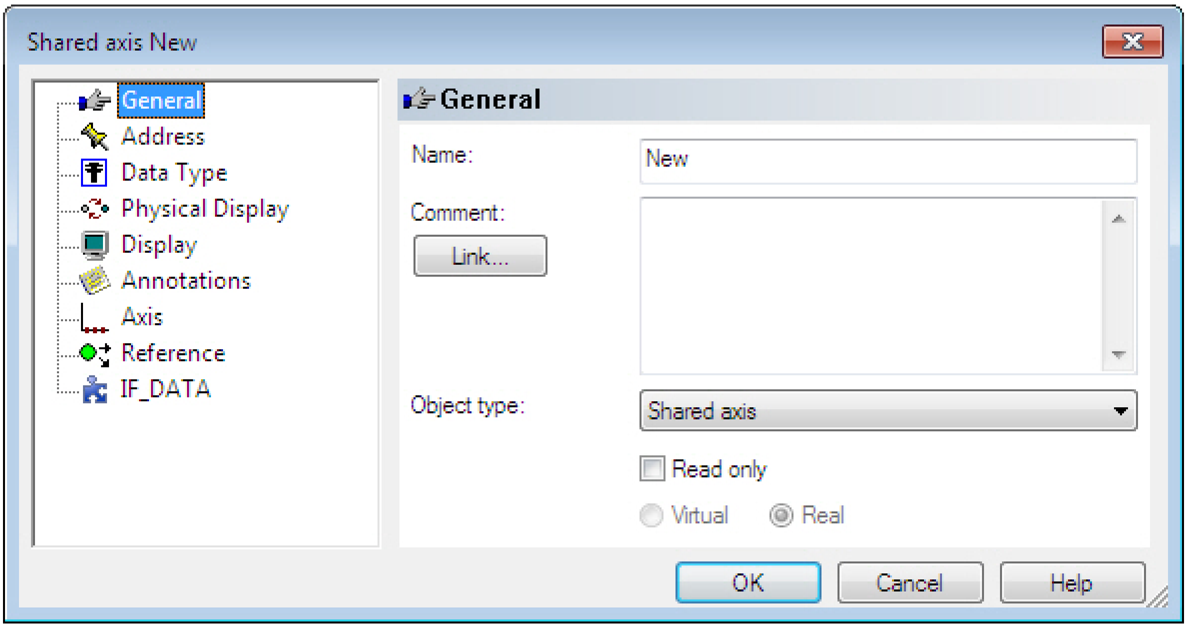

Figure 30: Dialog page "Shared axis - General"

The dialog pages are consistent with those of a parameter (see section 4.1.4, Seite 32) so only the dialog pages "Axis" and "Reference" will be explained here.

The dialog page "Axis" (see Figure 31) is where the X axis is configured.

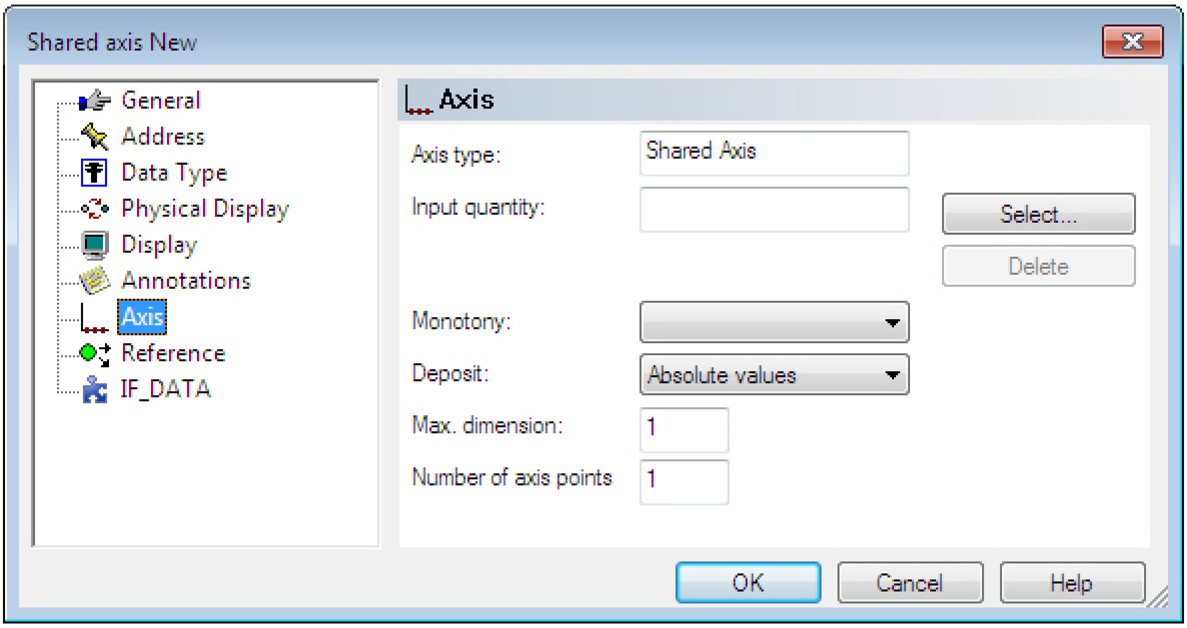

Figure 31: Dialog page "Shared axis - Axis"

The type of axis is given as "Shared axis" in the **Axis type** field. This field is not editable; the axis type can only be changed by a conversion, e.g. into a curve.

Clicking the **[Select]** button calls up the **Database Selection** dialog, from which you can assign a quantity to be measured to the shared axis. Assigning such an input parameter to a shared axis makes it possible later to display the working point in the curve or map during the measurement.

The shared axis properties **increasing/decreasing** or **strict increasing/strict decreasing** can be set via the drop-down menu **Monotony**. If one of these properties is selected, CANape checks to see if monotony is observed when values are changed in the Calibration window. The several monotony types are explained in detail in the online help.

In the Deposit field you can set how to interpret the axis value stored in the ECU:

Absolute value - every value corresponds to an axis value.

Differences - the first value corresponds to the start value and the other values correspond the respective differences between them.

Example

Absolute: 10, 12, 15, 17, 20, 22
Difference: 10, 2, 3, 2, 3, 2

The number of sampling points on the shared axis is entered in the fields **Max. dimension** and **Number of axis points**. Depending on the axis configuration only in of the fields an entry can be made; the other value is for information only.

If a user-defined record layout is selected for the shared axis in which the number of sampling points is defined, this field is then deactivated.

The dialog page "Reference" (see Figure 32) shows information on the use of the shared axis, i.e. for which objects the shared axis is used as X or Y axis.

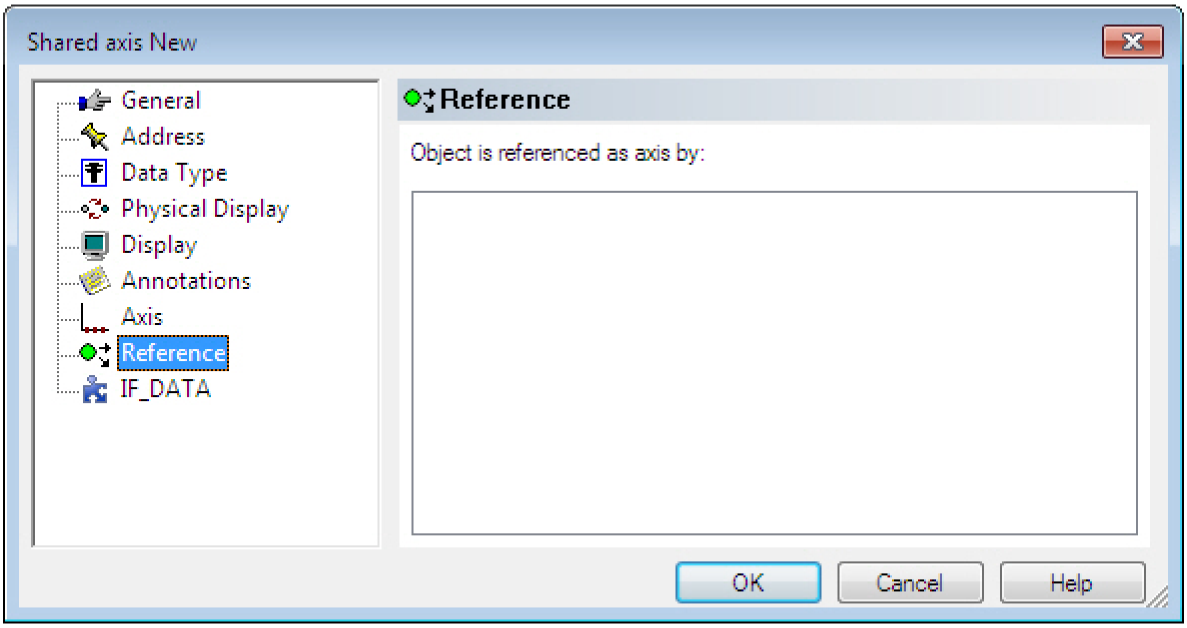

Figure 32: Dialog page "Shared axis - Reference"

When the shared axis is newly created, this page is empty. After the shared axis was assigned as an axis to a map, the respective object is displayed on this dialog page.

The page is for information only. The assignment to the respective object cannot be changed here. A change must be made in the "**X Axis**" or "**Y Axis**" dialog of the respective object.

The properties of an existing shared axis can be changed by highlighting it and right clicking and selecting **Properties**, which reopens the dialog page "General" (Figure 30) via the main menu item **Edit | Properties** or by double clicking it.

A shared axis can be converted to another data type (measurement value, parameter, map, curve or value block) via the popup menu. The new object keeps its name and receives the respective new symbol.

Please refer to the online help function for more detailed information about this dialog.

4.1.7 Maps

A characteristic map is always a calibration object of the controller, i.e. you can modify individual values during the measurement.

Since characteristic maps may be saved differently in the controller (e.g. by rows, by columns or user-defined), how the characteristic map values are stored in the controller memory must be defined in the controller database.

To add a new map to an existing group, the group must be selected. Then the entry **New | Map** is selected in the pop-up menu or the menu item **Edit | New | Map** is selected. This opens the dialog shown in Figure 33.

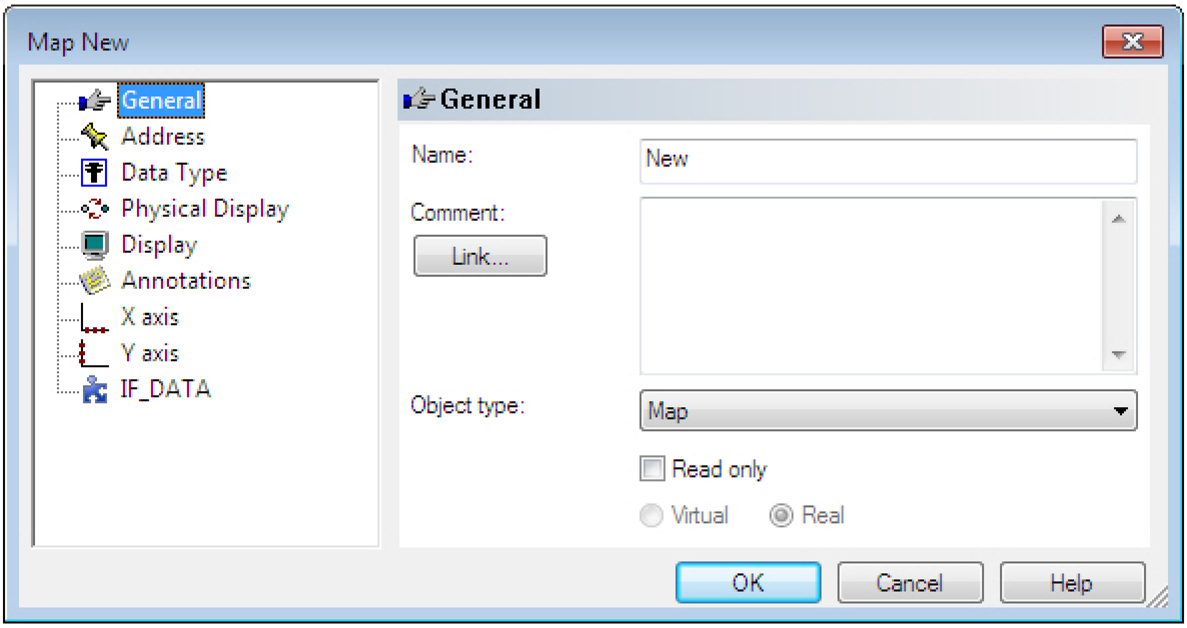

Figure 33: Dialog page "Map - General"

The dialog pages are consistent with those of a parameter (see section 4.1.4,on page 32) so only the dialog pages "X-axis" and "Y-axis" will be explained here.

The dialog page "X axis" (see Figure 34) is where the X axis is configured.

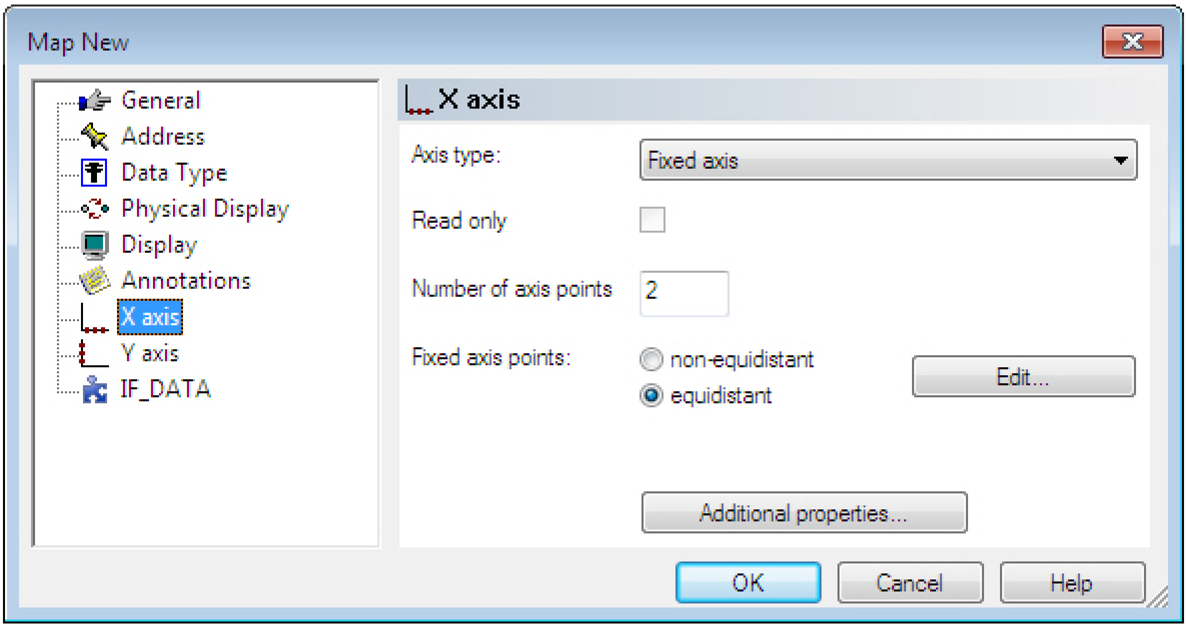

Figure 34: Dialog page "Map - X Axis"

Maps are distinguished by their axis types. The following axis types are possible:

Axis type	Description
None	The map has no axis values, i.e. it is simply a 1-dimensional or 2-dimensional table of function values. Indices (e.g. X1, X2, Xn) are shown instead of axis values in the calibration windows. If this axis type (**None**) is also set for the Y axis, the map corresponds to a value block.
Standard axis	The axis values together with function values of the map are considered as one object with a common starting address. The map structure in the controller's RAM memory is defined by a record layout. Based on this record layout the measurement and calibration system (e.g. CANape) determines offsets for individual function or axis values. Axis values can be calibrated like the function values. The axis type **Standard axis** can only be selected if a record layout is assigned to the map which has axis points.
Rescale axis	The axis values are taken from a selectable shared axis (see Section 4.1.6, "Shared Axes"). Shared axes can only be selected if they contain a record layout with scaling pairs.
Rescale curve	The axis values are taken from a selectable curve (see Section 4.1.8, "Curves").
Shared axis	The axis values of the map are considered to be an independent object with its own address. A shared axis has the property that more than one characteristic map may be assigned to the same common axis. The axis values can be adjusted, which changes the characteristics of all maps referencing it.

Axis type	Description

Fixed axis	Fixed axes are virtual axes, i.e. the axis values are not part of the characteristic map data in the controller, but are only calculated based on settings in the project database. The individual fixed axis values of an equidistant fixed axis are thus calculated from the two parameters **Offset** and **Distance** as follows:
	
	`Axis value(i) = Offset + ((i - 1) * Distance)`
	`where i = { 1...N }`
	
	Non-equidistant fixed axes can also be defined in the ASAP2 Editor.
	
	The axis values cannot be changed in the calibration window.
	
	If the **working point** in a map is to be displayed during the measurement, an input parameter must be assigned to each axis. This assignment is made for shared axes in the dialog of the referenced shared axis. For fixed or standard axes the input variable is assigned via the **Edit X axis** or **Edit Y axis** of the map dialog. It is not possible to display the working point if curves or maps use the axis type **none**.

Depending on the axis type, additional settings can be made, such as the number of sampling (axis) points, selecting shared axes, etc. The options for each axis type are discussed in the Online Help.

The Y axis is configured on the next dialog page (see Figure 35). The inputs are the same as those for the X axis.

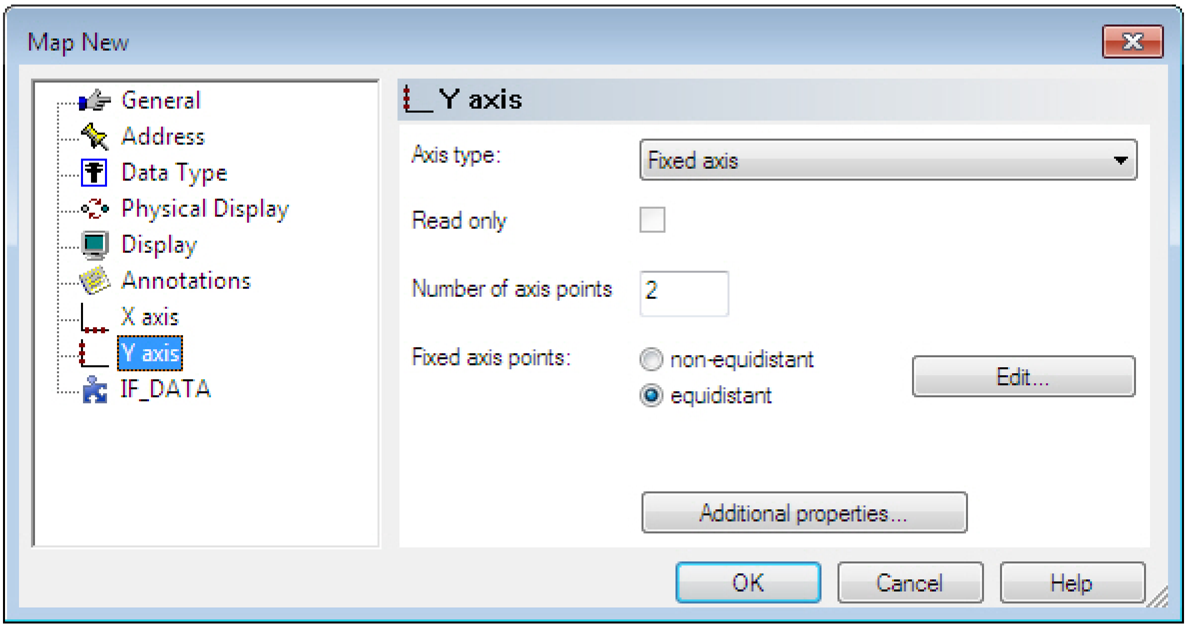

Figure 35: Dialog page "Map - Y Axis"

The properties of an already existing map can be changed by highlighting it and right clicking and selecting **Properties**, which reopens the dialog page "General" (Figure 33, Seite 43) via the main menu item **Edit | Properties** or by double clicking it.

A map can be converted to another data type (measurement value, parameter, map, curve or value block) via the popup menu. The new object keeps its name and receives the respective new symbol.

Please refer to the online Help function for more detailed information about this dialog.

4.1.8 Curves

Info
A characteristic curve is regarded by the ASAP2 Editor as a special case of a map (see Section 4.1.7, "Maps") where there is no Y axis and the number of rows of functional values is equal to 1.

To add a new curve to an existing group, the group must be selected. Then the entry **New | Curve** is selected in the pop-up menu or the menu item **Edit | New | Curve** is selected. This opens the dialog shown in Figure 36.

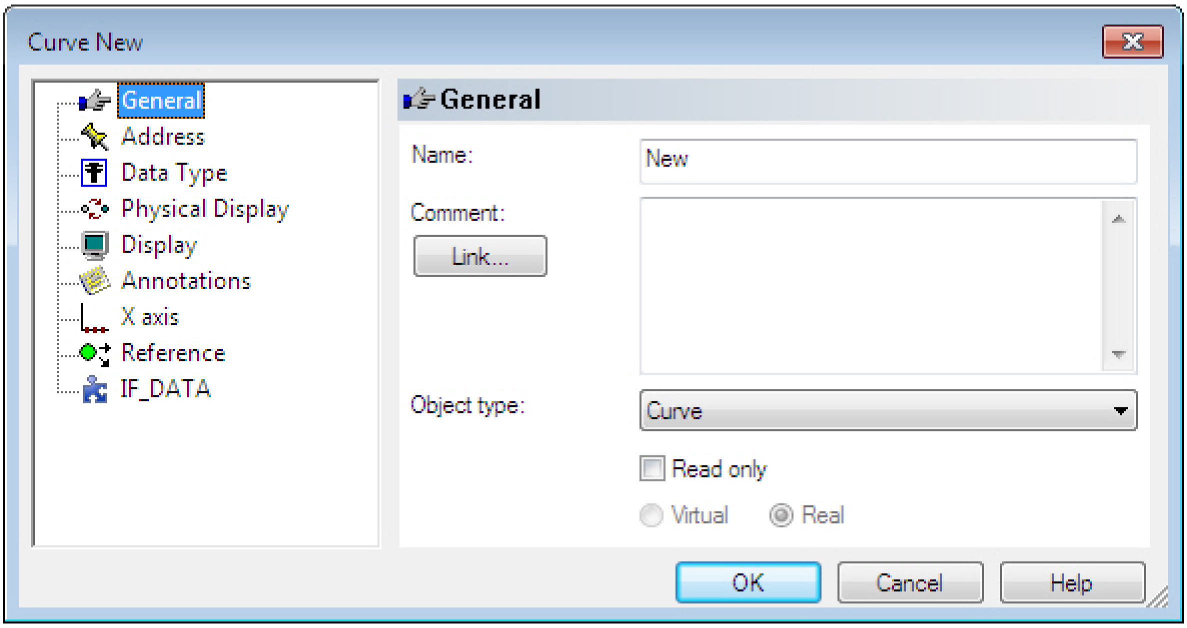

Figure 36: Dialog page "Curve - General"

The dialog pages are consistent with those of a parameter (see section 4.1.4, on page 32) so only the dialog page "Reference" will be explained here.

This dialog page (see Figure 37) shows information on the use of the characteristic curve, i.e. for which objects (maps) the characteristic curve is used as X or Y axis.

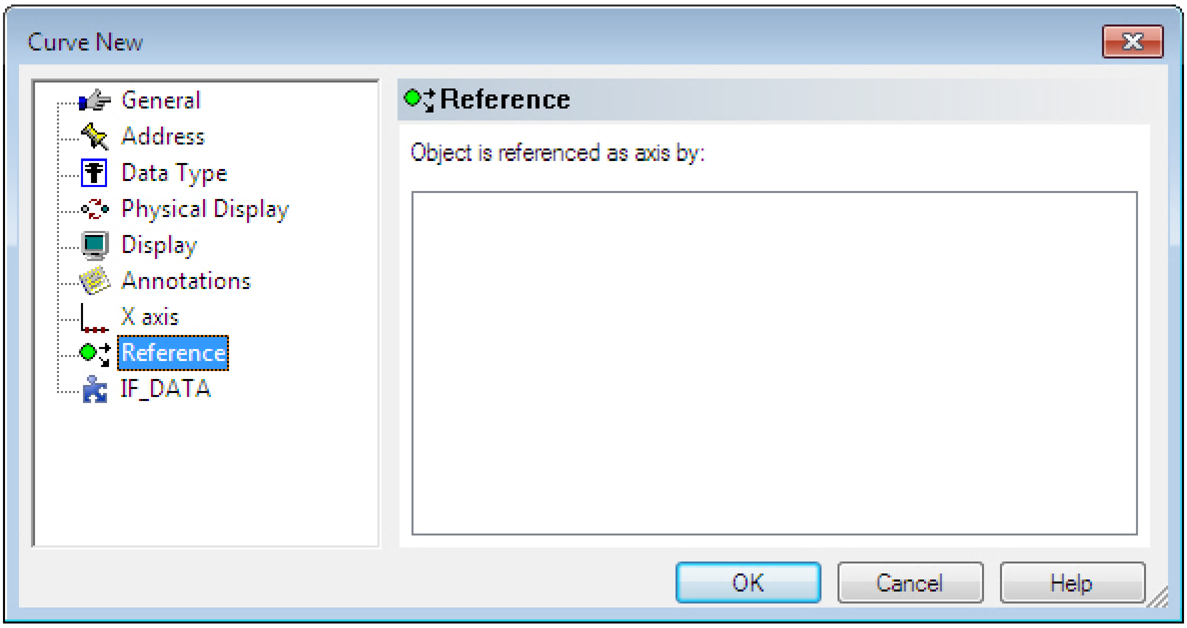

Figure 37: Dialog page "Curve - Reference"

When the characteristic curve is newly created, this page is empty. After the characteristic curve was assigned as an axis to a map, the respective object is displayed .

The page is for information only. The assignment to the respective object cannot be changed here. A change must be made in the "**X Axis**" or "**Y Axis**" dialog of the respective object.

The properties of an existing curve can be changed by highlighting it and right clicking and selecting **Properties**, which reopens the dialog page "General" (Figure 36) via the main menu item **Edit | Properties** or by double clicking it.

A curve can be converted to another data type (measurement value, parameter, map, curve or value block) via the popup menu. The new object keeps its name and receives the respective new symbol.

Please refer to the online help function for more detailed information about this dialog.

4.1.9 Value Block

Value blocks can be seen as maps or curves depending on the number of rows) that do not contain axes, i.e. are matrices.

To add a new value block to an existing group, the group must be selected. Then the entry **New | Value block** is selected in the pop-up menu or the menu item **Edit | New | Value block** is selected. This opens the dialog shown in Figure 38.

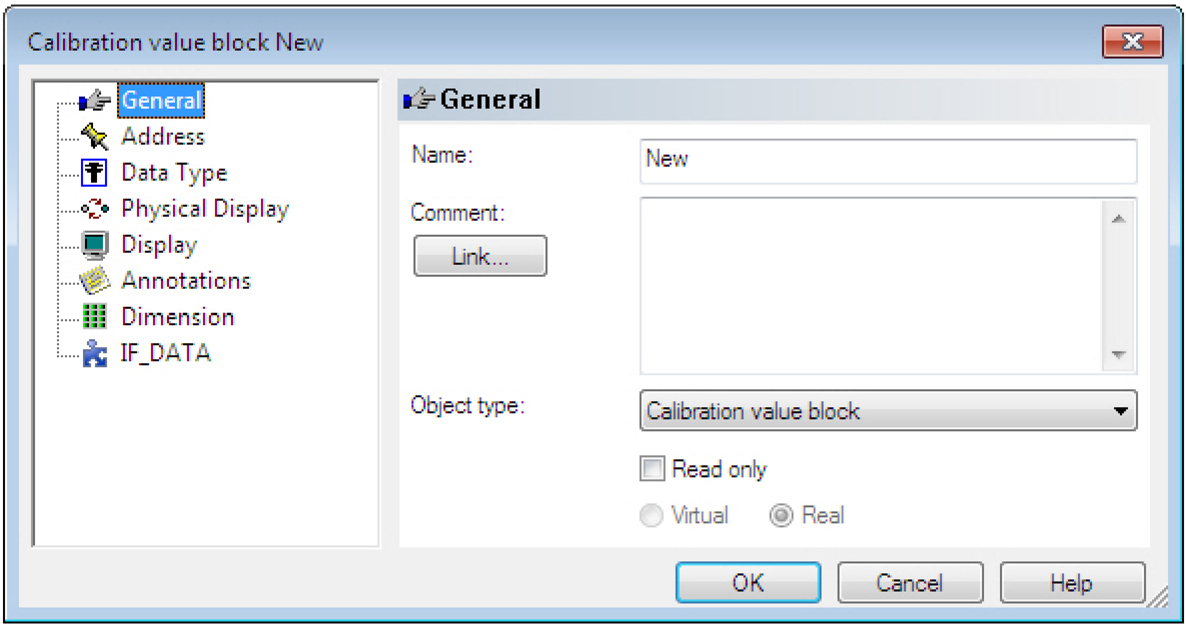

Figure 38: Dialog page "Value block - General"

A symbolic name for the value block must be entered in the **Name** input field. A short description can also be entered in the **Comment** field.

In the field **Object type** the user can decide between **Calibration value block** (in readwrite mode) and **Measurement value block** (in readonly mode). The object types are displayed with different icons**.**

The dialog pages are consistent with those of a parameter (see section 4.1.4, on page 32) so only the dialog page "Dimension" will be explained here.

At this dialog page (see Figure 39) the dimensions are determined.

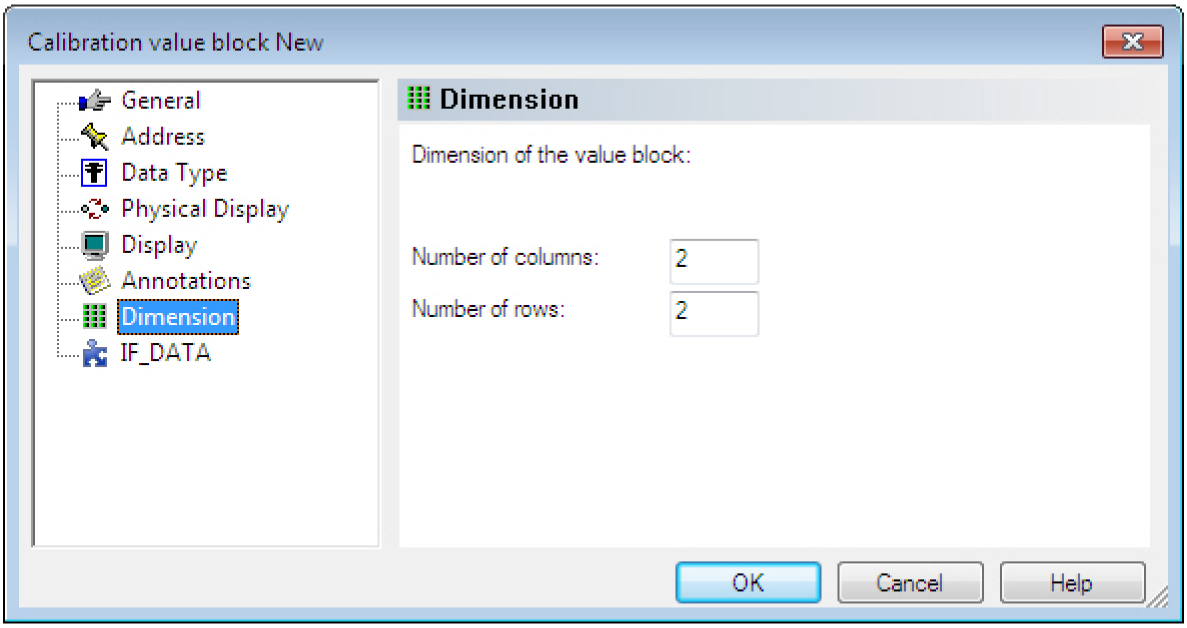

Figure 39: Dialog page "Value block - Dimension"

If the number of rows is 0 (Null), the value block corresponds to a curve with no axes.

If the number of rows and columns are both 1, the value block corresponds to a calibration object.

The properties of an already existing value block can be changed by highlighting it and right clicking and selecting **Properties**, which reopens the dialog page "General" (Figure 38) via the main menu item **Edit | Properties** or by double clicking it.

A value block can be converted to another data type (measurement value, parameter, map, curve or value block) via the popup menu. The new object keeps its name and receives the respective new symbol.

Please refer to the Online Help function for more detailed information about this dialog.

4.1.10 Virtual Measurement Signals

Virtual signals can be calculated from measured input signals in real-time using algebraic formulas. These virtual signals can be displayed - like other controller variables - in display elements of a measurement and calibration system such as CANape.

To add a new virtual signal to an existing group, the group must be selected. Then the entry **New | Virtual measurement signal** is selected in the pop-up menu or the menu item **Edit | New | Virtual measurement signal** is selected. This opens the dialog shown in Figure 40.

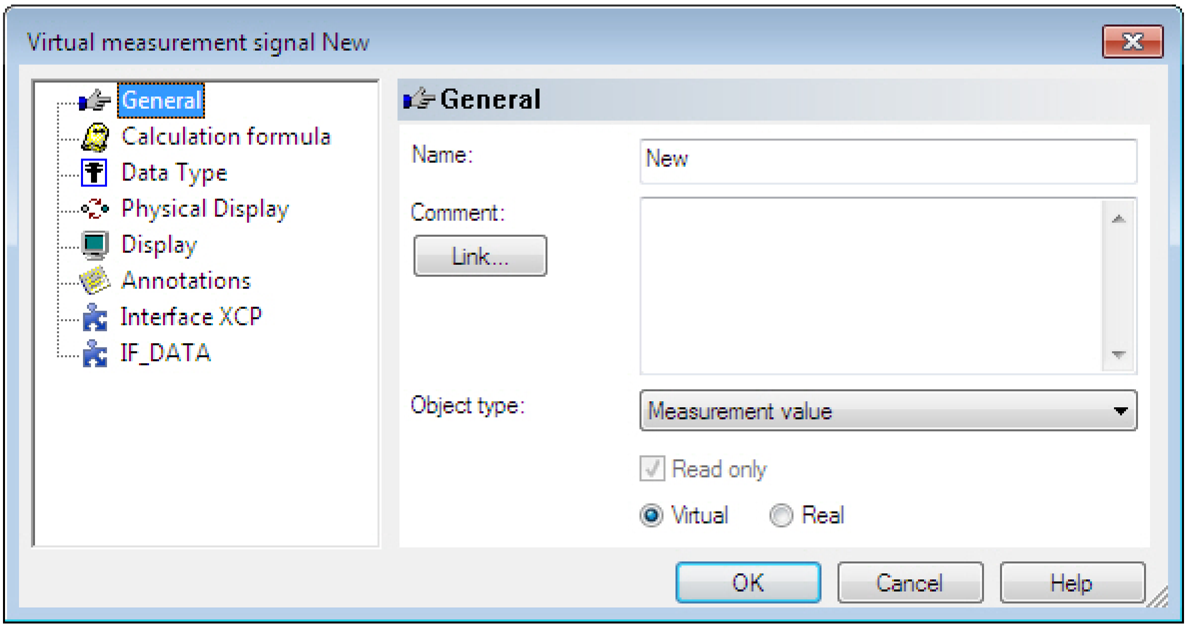

Figure 40: Dialog page "Virtual signal - General"

The settings of the several dialog pages correspond with those of a measurement value (see paragraph 4.1.3, "Measurement Values"), with the option **Virtual** pre-selected (see Figure 40). This removes the page "Address" and adds the "Calculation formula" page.

The properties of an already existing virtual measurement value can be changed by highlighting it and reopening the dialog page "General" (Figure 40) by right clicking and selecting **Properties**, the main menu item **Edit | Properties** or by double clicking it.

A virtual signal can be converted to another data type (measurement value, parameter, map, curve or value block) via the popup menu. The new object keeps its name and receives the respective new symbol.

Please refer to the online Help function for more detailed information on each of these dialogs.

4.1.11 ASCII Strings

An ASCII string object can e.g. read out error messages from the ECU via the KWP2000 Protocol. An ASCII String is a variable of an ECU program. It is described by e.g. a name and its address. The data type is always String.

Info
ASCII string objects are only supported by ECUs with KWP2000 drivers.

To add a new ASCII string block to an existing group, the group must be selected. Then the entry **New | ASCII string** is selected in the pop-up menu or the menu item **Edit | New | ASCII string** is selected. This opens the dialog shown in Figure 41.

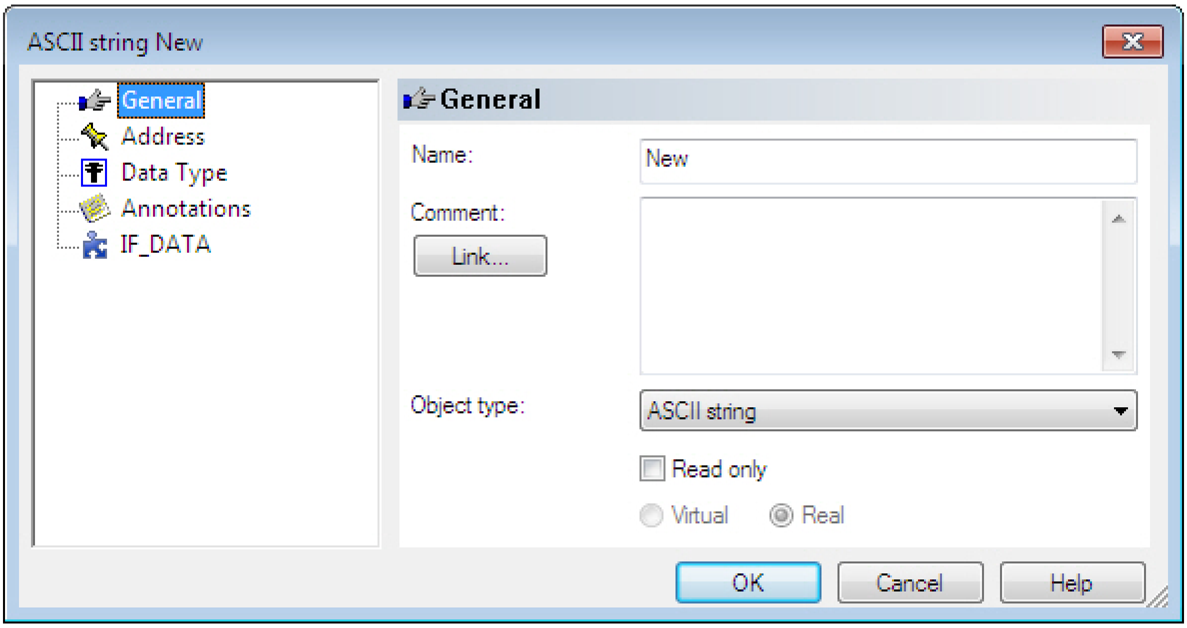

Figure 41: Dialog page "ASCII string - General"

The several dialog pages are already described in the "Parameter" paragraph (see section 4.1.4, "Parameters").

The properties of an already existing ASCII string can be changed by highlighting it and right clicking and selecting **Properties**, which reopens the dialog page "General" (Figure 41) via the main menu item **Edit | Properties** or by double clicking it.

An ASCII string can be converted to another data type (measurement value, parameter, map, curve or value block) via the popup menu. The new object keeps its name and receives the respective new symbol.

Please refer to the online Help function for more detailed information on each of these dialogs.

4.1.12 Conversion Rules

All conversion rules defined in the database are shown on the left side of the main window (see Figure 5). A dialog to prepare a globally defined conversion rule can be opened via the menu command **Edit | New**, see Figure 42**Fehler! Verweisquelle konnte nicht gefunden werden.**:

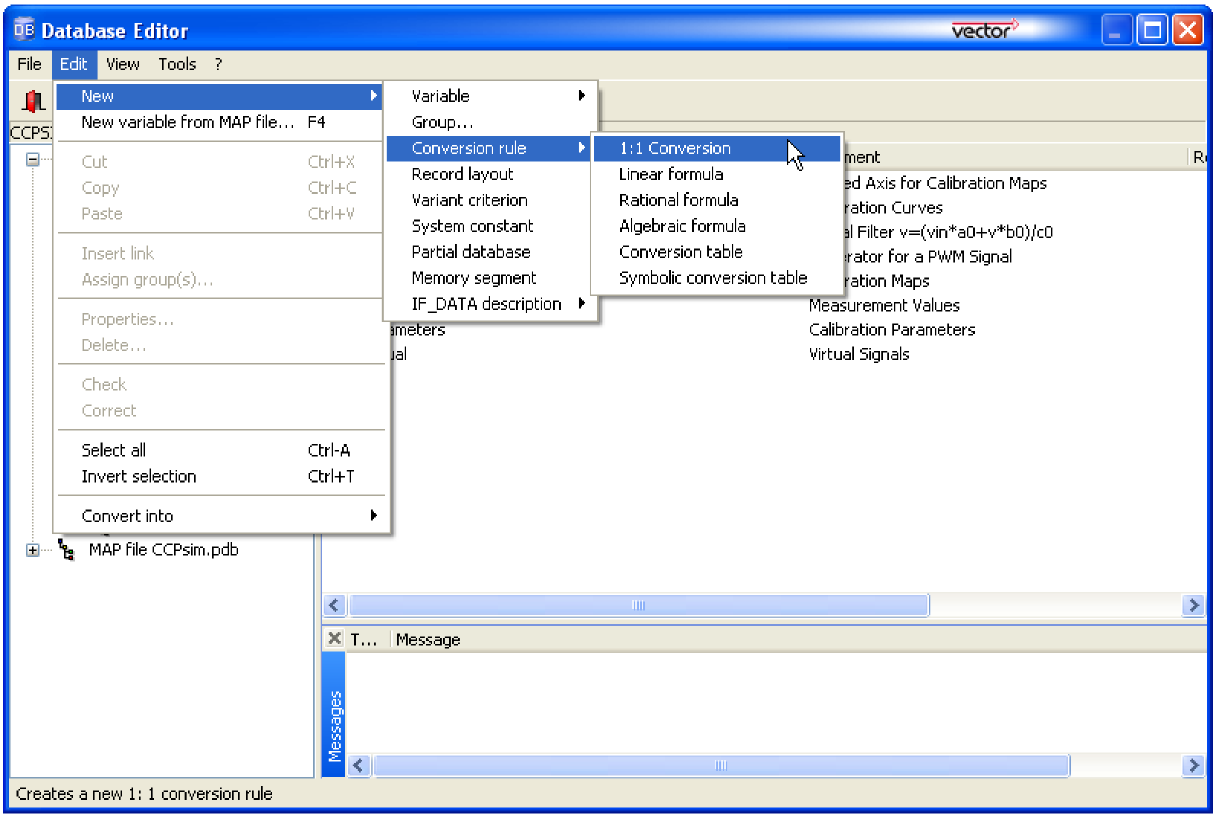

Figure 42: Defining Global Conversion Rules

The dialogs for creating a conversion rule have different pages where the respective properties can be set. All dialogs have a "General" page where general properties can be configured (see Figure 43). This page is the same as the "General" page for defining database objects.

In addition all dialogs have a "Status text table" page (see Figure 44). A status text table (ASAP2 key word: STATUS_STRING_REF) can divide a control unit-internal range of values into two parts: the "normal" range of values, for which may be defined a conversion rule, and an additional range, for which strings stored in the status text table are defined.
This makes it possible to use parts of the range of values for verbal status messages, for example. The online help explains the use of status text tables by means of an example.

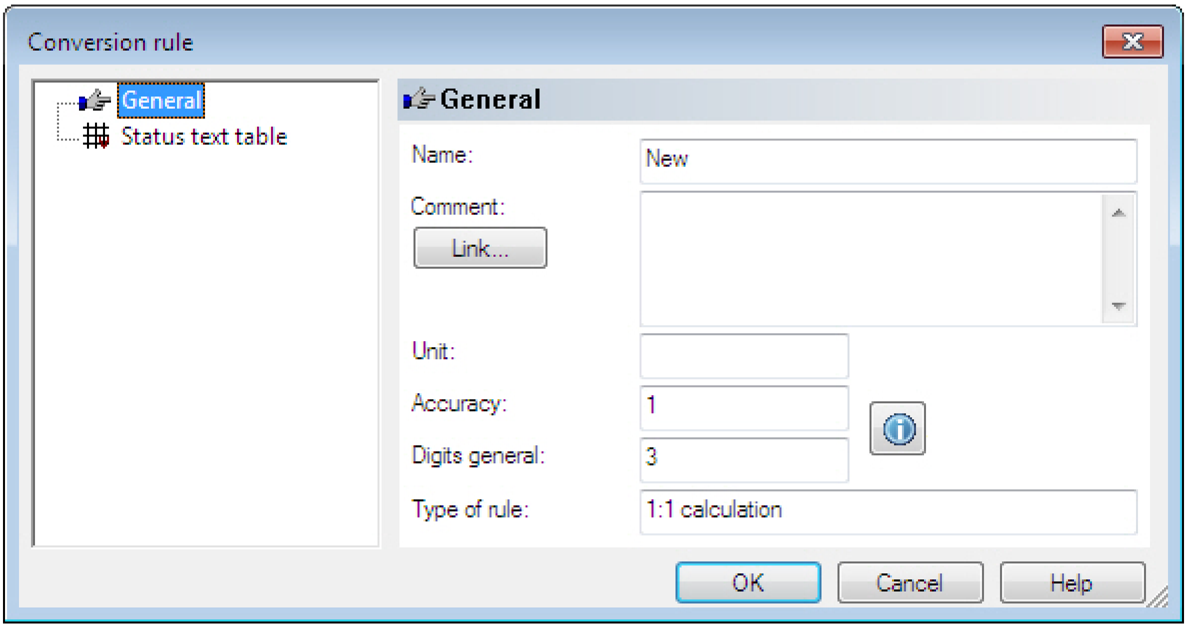

Figure 43: Dialog page "Conversion rule – General"

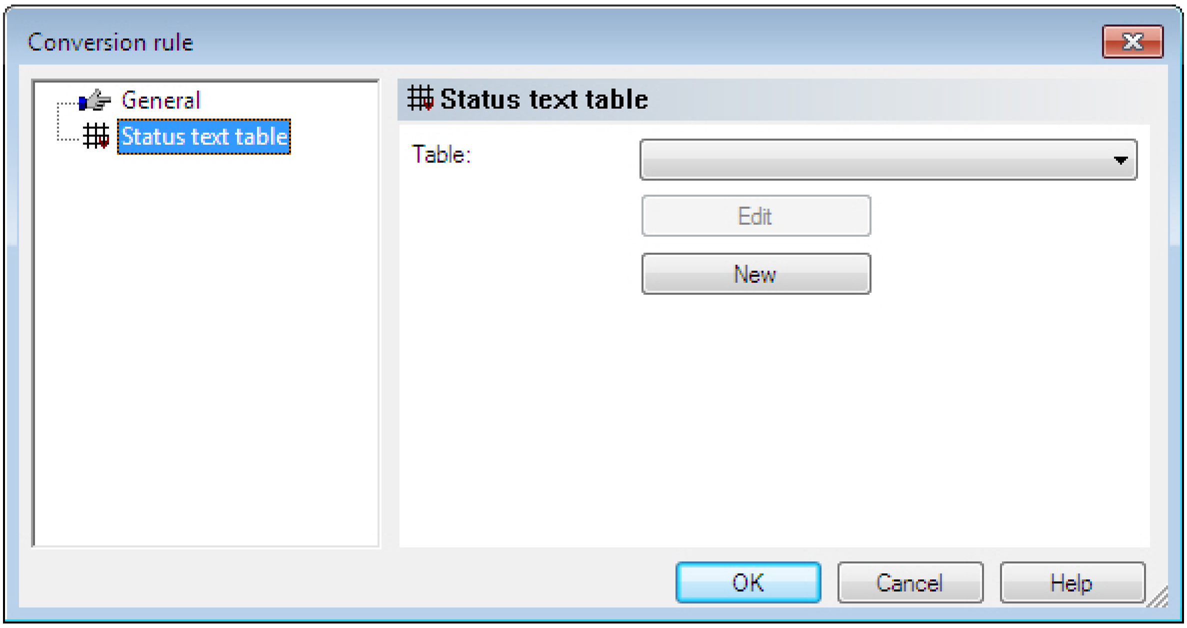

Figure 44: Dialog page "Conversion rule – Status text table"

1:1 Conversion

A 1:1 conversion is equivalent to the special case of a linear conversion with a factor of 1 and an offset of 0. This can be used to assign a predefined unit of measurement to a database object without having to do any additional conversions.

Info
The unit of measurement defined in the **New Conversion Rule** dialog is assigned to the conversion rule permanently. When selecting the conversion rule in the editor window for database objects this unit of measurement is automatically accepted and can no longer be changed there.

This dialog has no additional pages.

Linear Formula

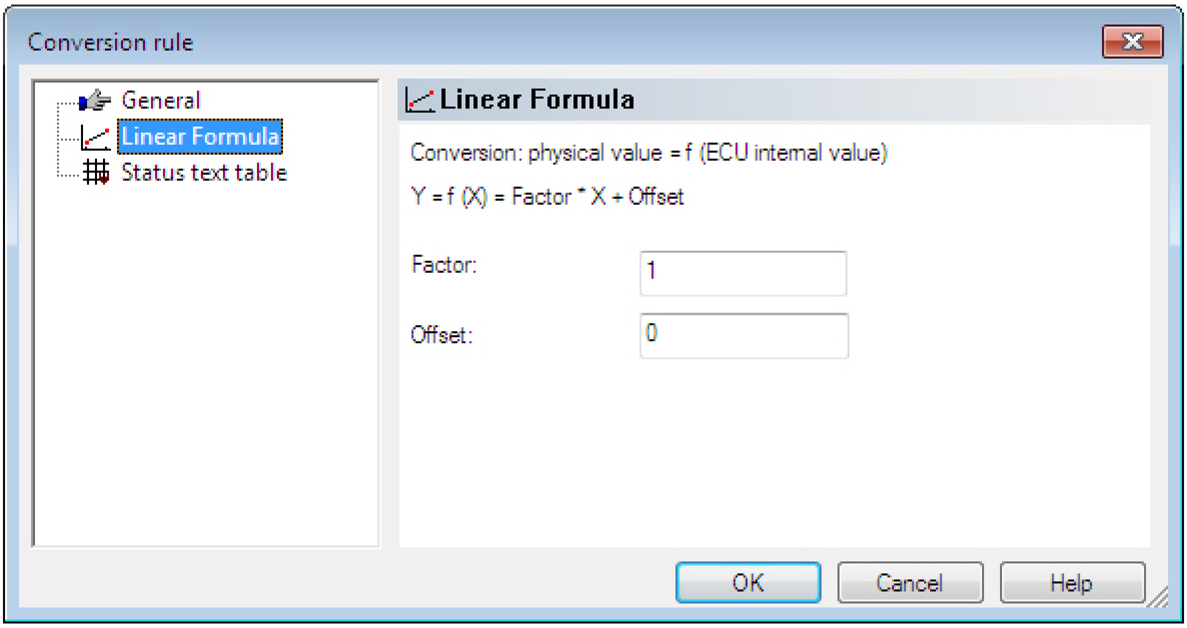

Figure 45: Formula editor for linear conversion

The parameters **Factor** and **Offset** can be set any way desired, but **Factor** must be assigned a numeric value (not zero).

Rational Formula

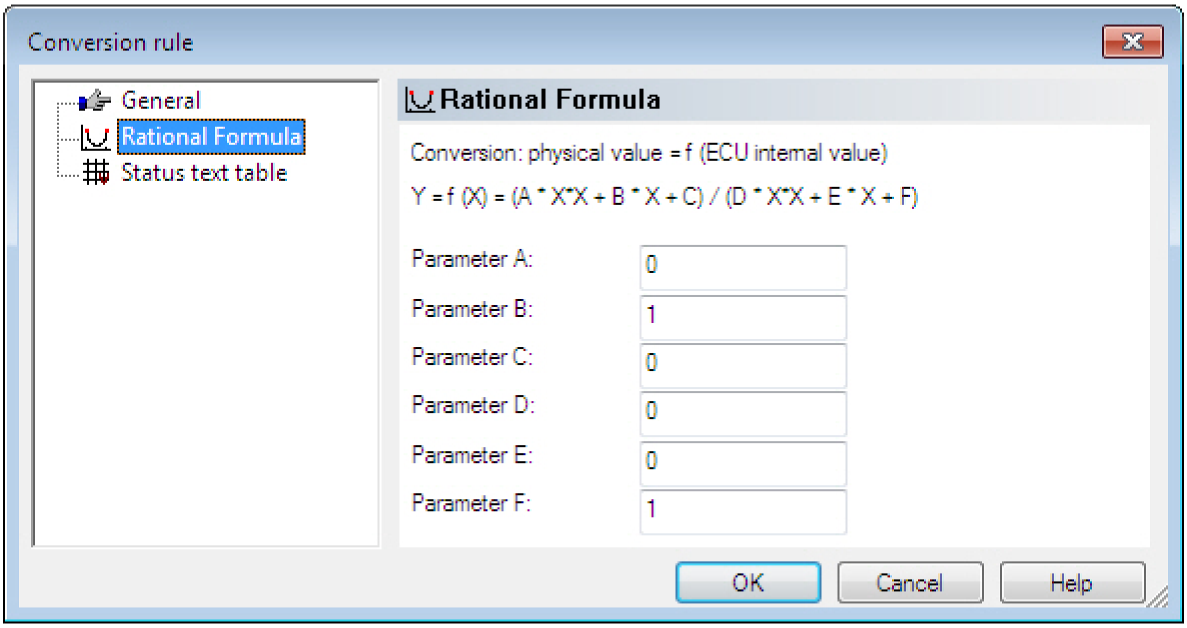

Figure 46: Formula editor for rational conversion

A rational conversion formula is entered in the form $f(x) = (A*x^2 + B*x + C) / (D*x^2 + E*x + F)$. Parameters A through F can be entered any way desired, but the denominator of the formula must not be zero!

Algebraic Formula

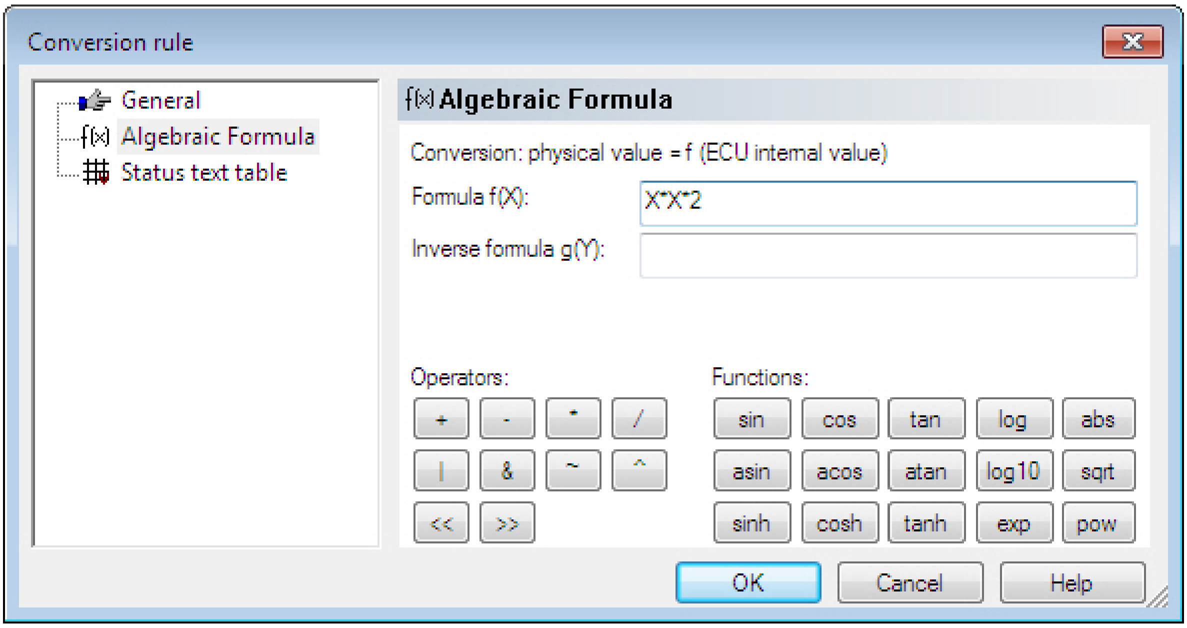

Figure 47: Formula editor for algebraic conversion

An algebraic or Boolean expression with the operators listed in the table can be entered into an algebraic conversion formula. There are many functions available as well. Brackets are also possible.

During conversion the current variable value is accessed with X. In the above example (Figure 47) the current signal value is squared and multiplied by 2.

Info
There is no distinction made between upper case and lower case with "X".

Since algebraic conversion specifications are not always clearly invertible, an algebraic expression can also be entered for the inversion formula. The inversion formula, which calculates the associated ECU internal value from a physical value, is for example required when parameters in physical representation are modified in a calibration window. Entering an inversion formula is optional, but it is recommended, since an iteration process is used when it is not present which under unfavorable circumstances either converges poorly or not at all.

Caution
Parameters with an algebraic conversion rule <u>without</u> an inverse formula cannot be calibrated in the physical format. To calibrate such parameters you have to switch to the decimal format or add an inverse formula.

The current signal value which is to be converted is used as the input. The letter “X” is used as the signal value.

Info
The new rational formula is only available after restarting the ASAP2 Editor. The actual conversion is done during this restart after confirmation by the user.

Numeric Conversion Table

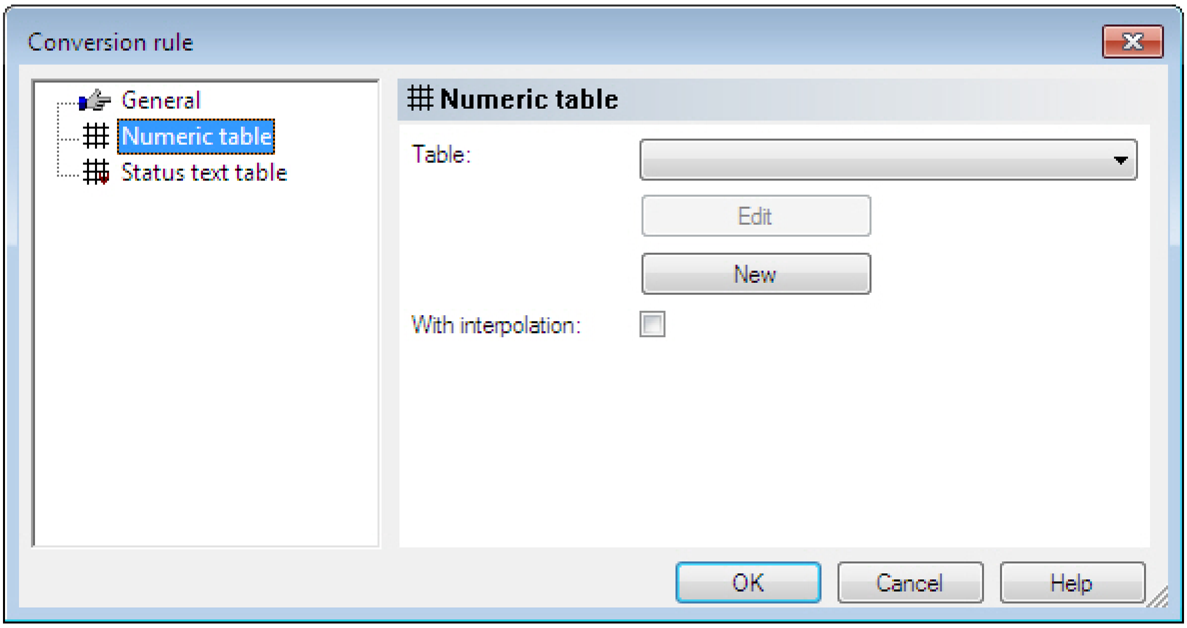

Figure 48: Conversion table

A conversion table assigns physical values to ECU-internal signal values (in hexadecimal and decimal representation).

In the field **Table** an already existing conversion table can be selected. This table can be changed by the **[Edit]** button.

Any number of hexadecimal signal values can be entered in the table. When the hexadecimal value is input, the corresponding decimal value appears in its column automatically, and vice versa. Every value may only be in the table once. The values can be input in any sequence. The values are automatically sorted in ascending order when the window is closed.

The button **[New]** opens an additional dialog to create a conversion table (see Figure 49).

The checkbox **With interpolation** indicates whether or not interpolation should be done for determining a physical value between the hexadecimal ECU-internal signal values if necessary. If this box is not checked, the physical value of the previous hexadecimal signal value is assigned to the respective ECU-internal signal value.

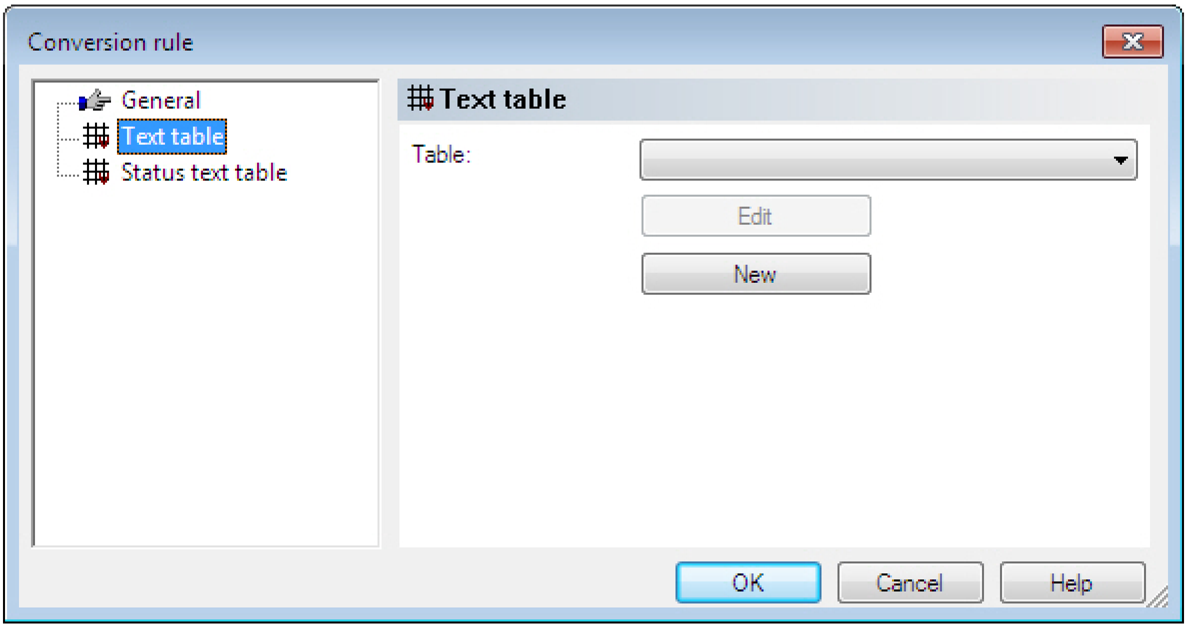

Figure 49: Dialog page to create a conversion table

Any number of hexadecimal signal values can be entered in the table. When the hexadecimal value is input, the corresponding decimal value appears in its column automatically, and vice versa. Every value may only be in the table once. The values can be input in any sequence. The values are automatically sorted in ascending order when the window is closed.

A text can be entered in the **Default text / Default value** input fields which are be used for displaying all signal values which are not in the table. If no default value is input, the current signal value will be output as a numerical sequence.

When a numerical value is to be set, the option **Default value** must be activated, otherwise the value is interpreted as string and not as numeric. A string with the option **Default value** is interpreted as 0 (zero).

Clicking the **[==> Clipboard]** button copies the entire conversion table to the clipboard, where it can be processed further. We recommend using Microsoft Excel be used for this.

Clicking the **[<== Clipboard]** button imports the contents of the clipboard.

Symbolic Conversion Table

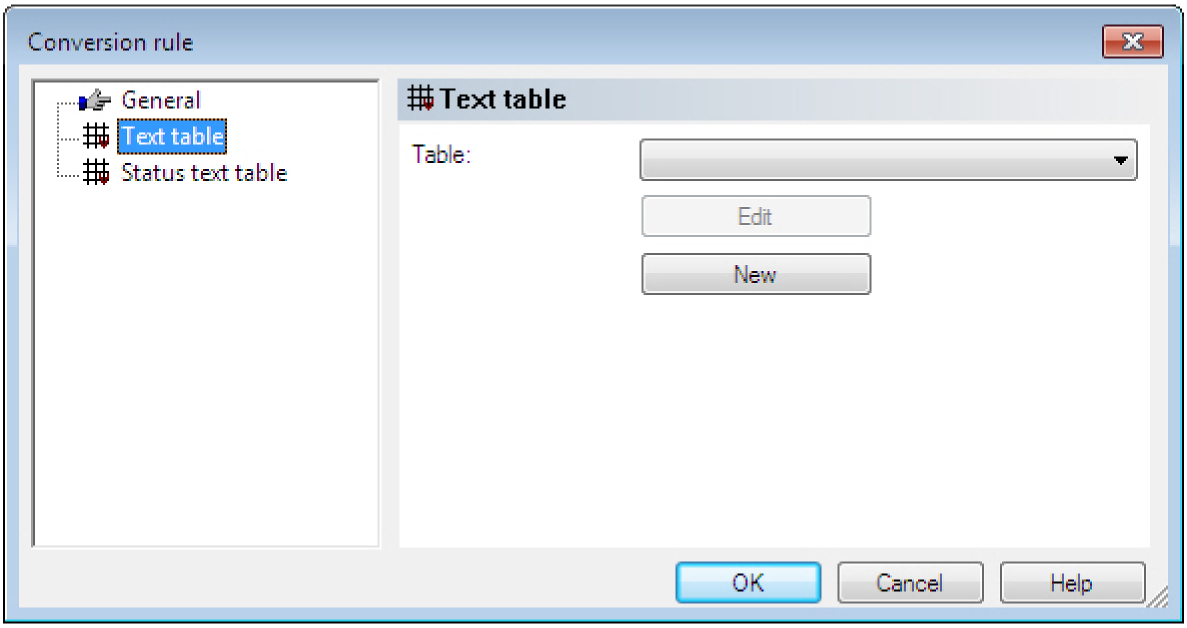

Figure 50: Symbolic conversion table

A symbolic conversion table assigns a text to a value which is output when the variable falls within that range of values. Any number of value/text pairs can be entered in the table, but every value may only be in the table once. The values in the table are sorted in ascending order.

In the field **Table** an already existing conversion table can be selected. This table can be changed by the **[Edit]** button.

The button **[New]** opens an additional dialog to create a conversion table (see Figure 51)

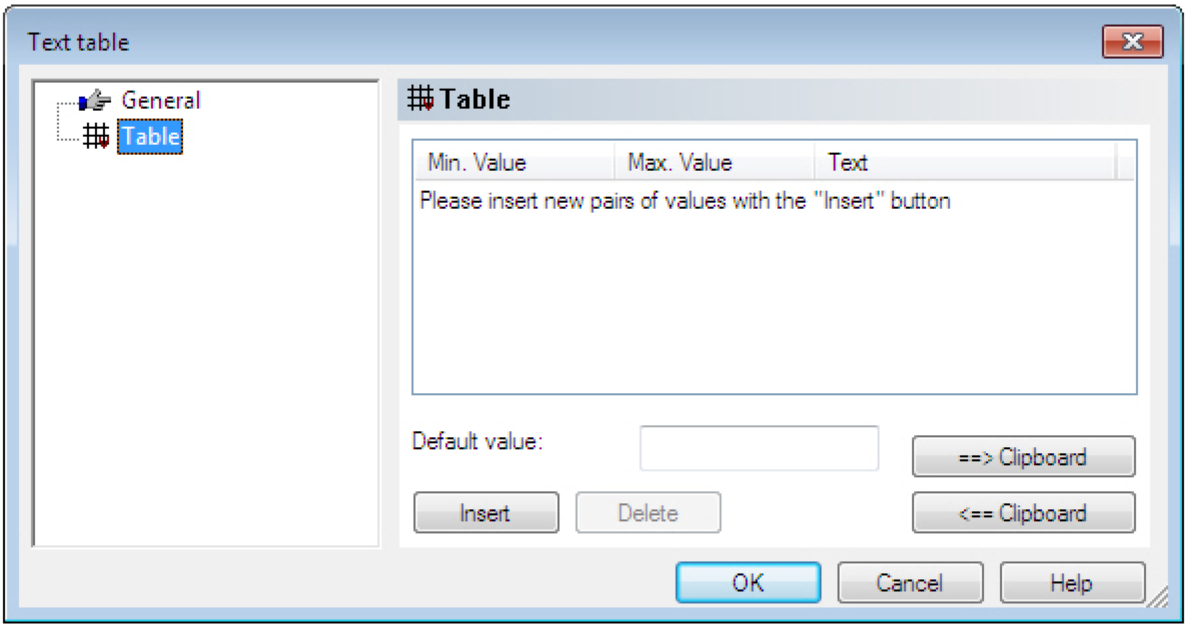

Figure 51: Dialog page to create a symbolic conversion table

Clicking the **[==> Clipboard]** button copies the entire conversion table to the clipboard, where it can be processed further. It is recommended that Microsoft Excel be used for this.

Clicking the **[<== Clipboard]** button imports the contents of the clipboard.

A text can be entered in the **Default** input field that will be used for displaying the signal values which are not in the table. If no default value is input, the current signal value will be output as a numerical sequence.

Info
A macro can also be input in this field, if for example a conversion needs to be defined for all values that are not defined in the table.

Example
A macro for the linear conversion of a measurement value with a factor of 12 and an offset of 34:

```
{EVAL "{X}*12 + 34"}
```

4.1.13 Record Layouts

A record layout represents a structure definition and is comparable to a `struct` declaration in the C programming language. Any number of objects can be derived from this structure definition, i.e. the controller program may contain any number of elements of the described structure. A record layout describes the structure of a characteristic map or curve in the controller program:

- What elements is the characteristic parameter composed of (e.g. interpolation point values, table values)?
- At which position, relative to the object's start address, are these elements stored?
- What data types do the individual elements have?
- Which supplemental properties and attributes do the elements have (e.g. sequence in lists: interpolation point values or table values)?

The ASAP2 Editor implementation covers all properties and configuration options that are declared with the keyword RECORD_LAYOUT in ASAP2 Standard Version 1.4. That document contains a detailed explanation of all attributes and configuration options.

All record layouts defined in the database are displayed on the left side of the main window (see Figure 5, page 20).

Record layouts can be defined by selecting the menu command **Edit | New | Record layout**. This opens the first page of the dialog seen in Figure 52.

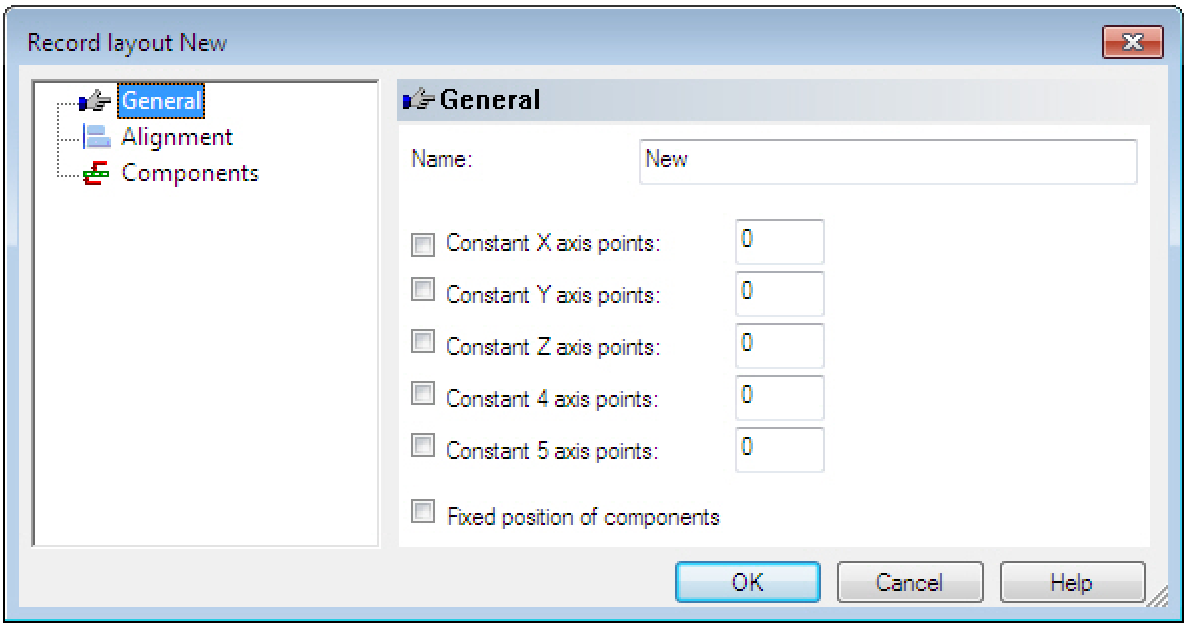

Figure 52: Dialog page "Record layout - General"

Activating one of the checkboxes **Constant axis points** indicates that the number of axis points defined in the input field to the right of it will be assigned to all derived parameters. This eliminates the possibility that the number of axis points might be contained in the corresponding parameter data record of the controller program.

When the option **Fixed axis points** is activated a shifting of the components in the memory is suppressed when the number of the respective components is reduced. This leads to empty memory areas.

At the next dialog page the alignments are defined (see Figure 53).

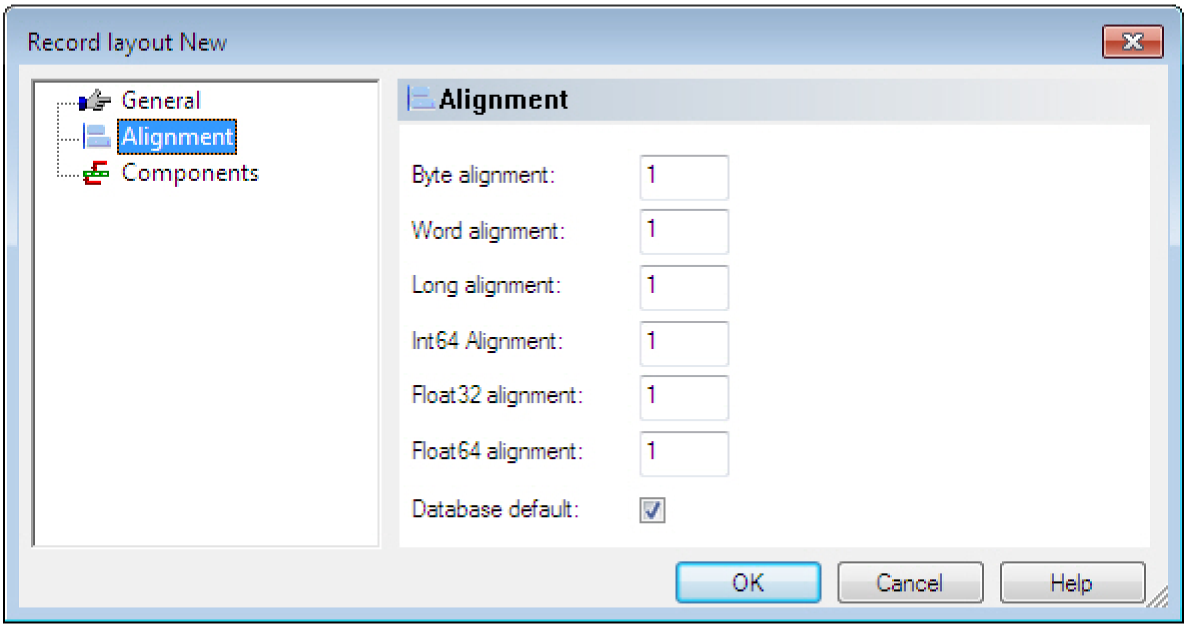

Figure 53:Dialog page "Record layout – Alignment"

The alignment values specify the addresses where the values for derived parameters begin – depending on data type.

The checkbox Database default has three possible settings, which are read out from the database. These settings can be changed by clicking.

- Selected (☑): The alignment settings of the database apply; the input fields in the dialog are therefore disabled.
- Cleared (☐): The specified alignment settings apply locally.
- Partially selected (■): Only individual alignment settings are specified in the ASAP2 file.

The next dialog page is for inputting the components of the record layout (see Figure 54).

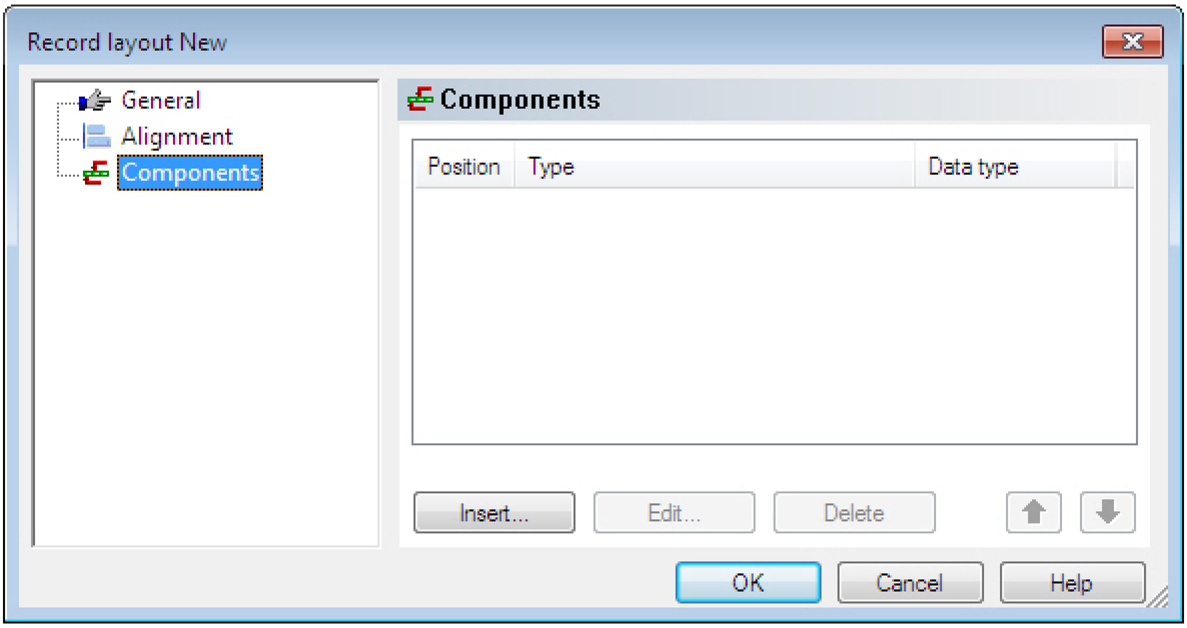

Figure 54: Dialog page "Record layouts - Components"

Clicking the **[Insert]** button opens a selection list of the available components. How to add components is described in the Online Help, and there is an example in Section 4.2.5, "Example 5: Record Layout".

4.1.14 Variant Criteria

Variant coding cannot be specified for individual characteristic parameters; this is done by groups. The same dependency applies to all characteristic variables in a group.

An example of a variant-coded controller is a transmission controller which supports - with regard to the variant criterion Transmission Type - the variants Four-Gear Transmission (Variant name: FourGearTrans), Five-Gear Transmission (Variant name: FiveGearTrans) and Six-Gear Transmission (Variant name: SixGearTrans). The controller program contains a copy of certain characteristic variables for FourGearTrans, FiveGearTrans and SixGearTrans in its data area. There is another characteristic variable, the so-called variant selector, in the controller program. The current value of the variant selector determines which variant is currently active.

As long as no variant criteria have been defined in the database, the variables are accessed in the controller's memory using constant addresses in the database which were either input directly or which resulted from references to symbols in the MAP file.

If you identify certain variables as variant-coded by assigning a variant criterion to each of them, then these variant-dependent characteristics are accessed via addresses whose values are determined at run time by the value of the relevant variant selector.

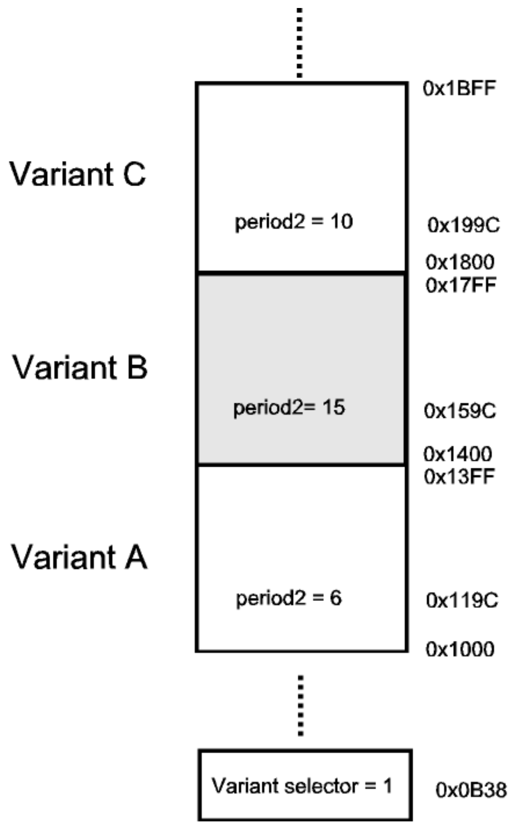

Figure 55: Variant-coded ECU

As can be seen in Figure 55, oben, the three variants - referred here as simply A, B and C - are located in three different data areas of the flash memory. The value of the variant selector defines and points to the active variant, i.e. the controller accesses the values of this variant at run time. If a variant-coded parameter is to be calibrated, the measurement and calibration system (e.g. CANape) automatically uses the variant-dependent object address for the write access.

We now illustrate how this variant-dependent object address is calculated in a measurement and calibration system such as CANape from the object address of the MAP file.

The program performs the address calculation according to these formulas:

Variant-dependent object address := Variant-independent object address + variant offset

Variant-independent object address := object address from the map file & mask

Depending on which variant-coded object addresses are located in the MAP file, there are two methods available to the controller developer for defining the mask.

Case 1: Object Addresses of Variant A Are Located in the MAP File

The variant offsets of variants B and C are found to be:

Variant offset B: = 0x0400

Variant offset C: = 0x0800

The address calculation to be performed by the program is simplified to:

Variant-dependent object address := Object address of variant A + Variant offset of active variant

Since in this case the variant-independent object address matches the object address of variant A, the mask must be set to the value: 0xFFFF FFFF.

Example

Let Variant C be the active variant. The object address from the MAP file for the calibration object **period2** is 0x119C. The variant offset of variant C is 0x0800. Then the variant-dependent object address of **period2** for Variant C is found to be

Variant-dependent object address = 0x119C + 0x0800 = 0x199C.

Case 2: Object Addresses of Variant B or C Are Located in the Map File

To allow to calculate the variant-dependent object address of an object address from the MAP file, a mask and all variant offsets are required (see address calculation formulas above).

If the length of the variant blocks is a power of two and the start address is an integer multiple of the length, the following method for determining the mask can be applied:

mask := length of the flash address range for a variant - 1

In this example, the mask is found to be 0x0400 – 0x1 = 0x03FF.

Then only the variant offsets need to be determined for the different variants:

Variant offset A:=	Start address of first variant in flash memory (here: 0x1000)
Variant offset B:=	Variant offset A + Length of the flash address range of a variant (here: 0x1000 + 0x0400 = 0x1400)
Variant offset C:=	Variant offset B + Length of the flash address range of a variant (here: 0x1400 + 0x0400 = 0x1800)

If the flash memory ranges are located one after another, the following applies:

Variant offset N:=	Variant offset (N-1) + Length of flash address range of a variant

Example

Let Variant B be the active variant. The object address of calibration object **period2** from the Linker MAP file is 0x199C. The variant offset for variant B is, as above, 0x1400. Then the variant-dependent object address of **period2** for Variant B is found to be

```
  0x199C           0001 1001 1001 1100
& 0x03FF         & 0000 0011 1111 1111
----------------------------------------
= 0x019C         = 0000 0001 1001 1100
+ 0x1400         + 0001 0100 0000 0000
----------------------------------------
= 0x159C         = 0001 0101 1001 1100
```

Defining Variant Criteria

All variant criteria defined in the database are displayed on the left side of the main window (see Figure 5, Seite 20).

To define variant criteria in the Database Editor, select the menu command **Edit | New | Variant criterion**. This opens the first page of the dialog seen in Figure 56.

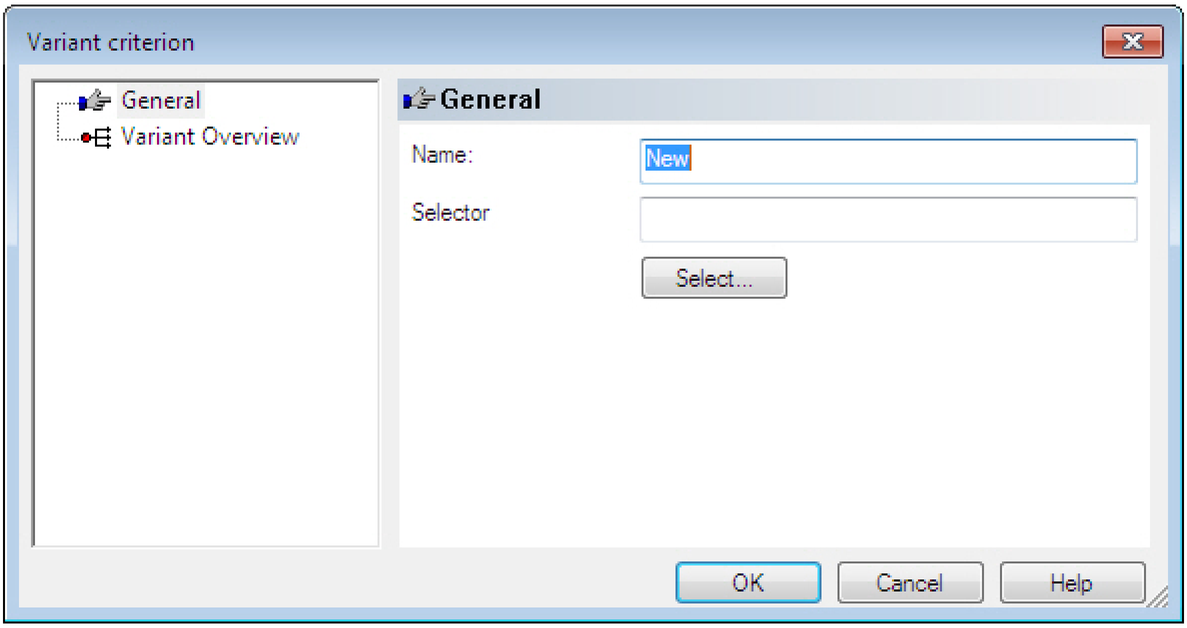

Figure 56: Dialog page "Variant criterion - General "

In this example, we will define a new variant criterion "TransmissionType". Enter "Transmission type" in the **Name** field, select a parameter to be the Selector and open the dialog page "Variant overview". Click the **[Insert]** button to define three variants, as shown in Figure 57**Fehler! Verweisquelle konnte nicht gefunden werden.**:

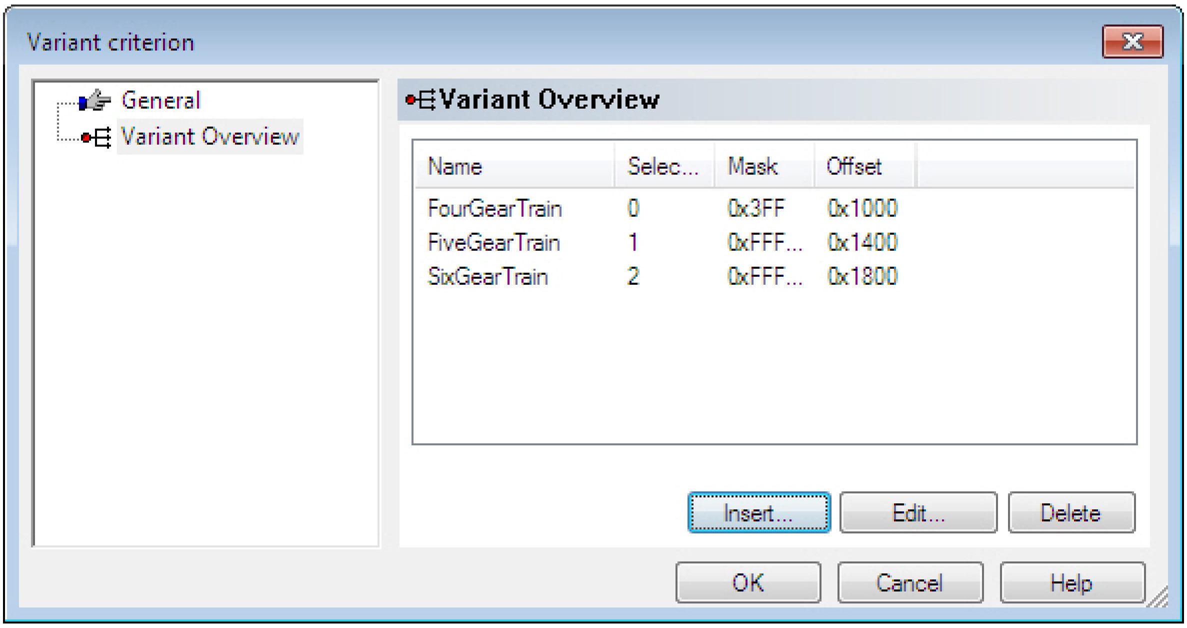

Figure 57: Dialog page "Variant criterion - Variant overview "

A new variant criterion can be created in this dialog. Every variant criterion is characterized by these attributes:

- **Identifier:** Name for the variant criterion
- **Variant selector:** Characteristic parameter whose value indicates which variant in the controller is active
- **Variants** have the following attributes:
    - Variant (the identifier, e.g. FourGearTrans)
    - Selector value (e.g. 2)
    - Mask (see above)
    - Offset (see above)

The dialog contains all entries for defining the Transmission Type variant criterion from the introductory example of this chapter.

Assigning a Variant Criterion to a Group

As long as no variant criteria have been defined in the ECU database, the program proceeds by assuming that no group contains variant-coded calibration objects (Criterion: disabled).

After variant criteria have been defined in the controller database, they can be assigned to the appropriate groups. To do this, open the Database Editor and select the group to which a variant criterion is to be assigned. Selecting **Properties** from the popup menu opens the following:

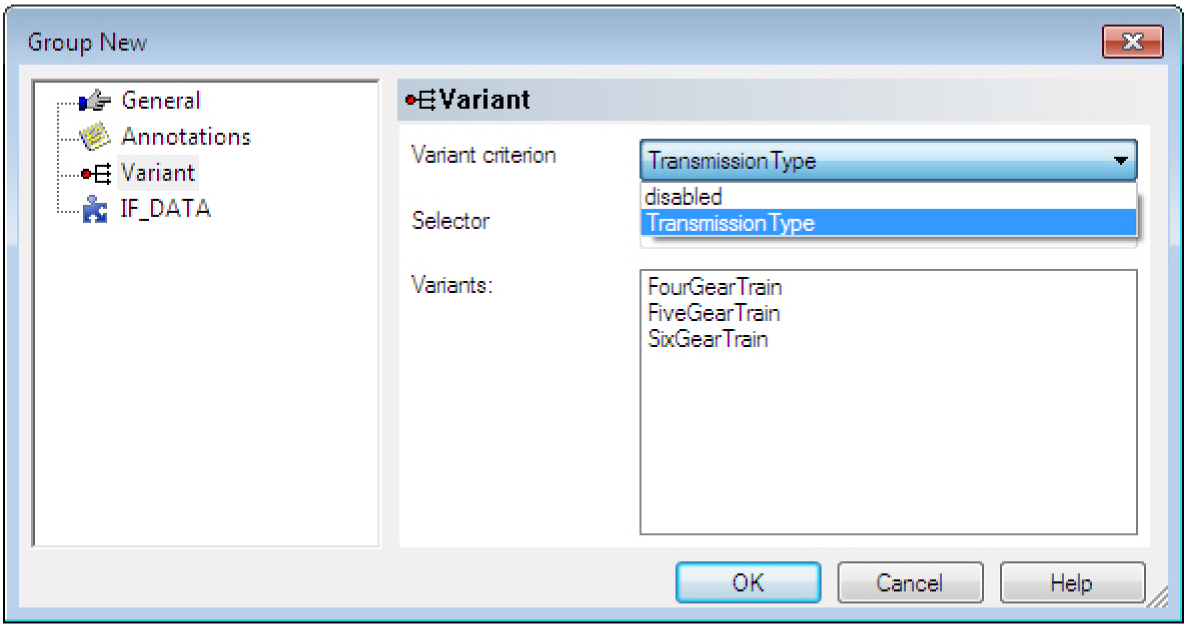

Figure 58: Assigning a Variant Criterion to a Group

Now you can assign a variant criterion from the selection list to the group. As can be seen in Figure 58, one variant criterion is available in this example: TransmissionType.

4.1.15 System Constants

All conversion rules defined in the database are displayed on the left side of the main window (see Figure 5, page 20). By selecting the menu command **Edit | New | System constant** you can define additional system constants (see Figure 59).

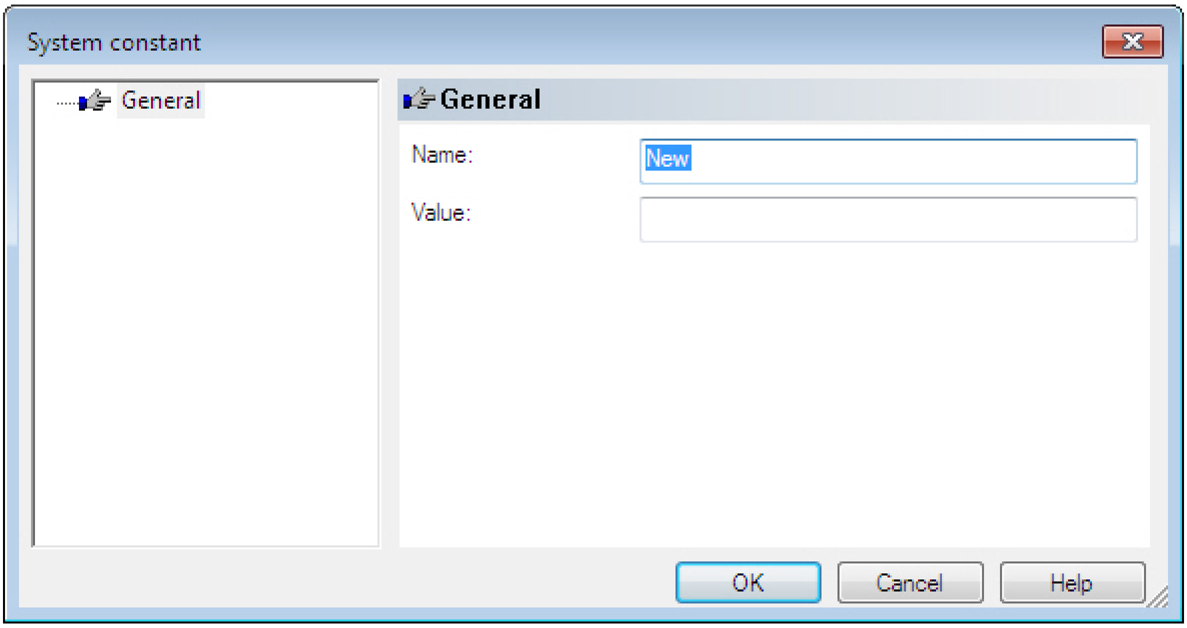

Figure 59: Dialog page "System constant"

System constants are stored in the database, and a name is assigned to each constant value. Another system constant can be used as a value.

Example

```
Name=PI
Value=3.14159
Name=PI_HALF
Value=PI/2
```

Info

Only the period (.) is to be used as the decimal separator.

4.1.16 Memory Segments

When the memory cache for offline calibration should be used the respective memory areas of the ECU – segments – must be given and configurated.

Info

Only in the standalone version of the ASAP2 Editor memory segments can be created and changed. When the ASAP2 Editor is opened from CANape the memory segments are changed in the Device Configuration dialog. In this case the properties dialog in the ASAP2 Editor is for information only.

By selecting the menu command **Edit | New | Memory segment** you can define additional memory segments (see Figure 60).

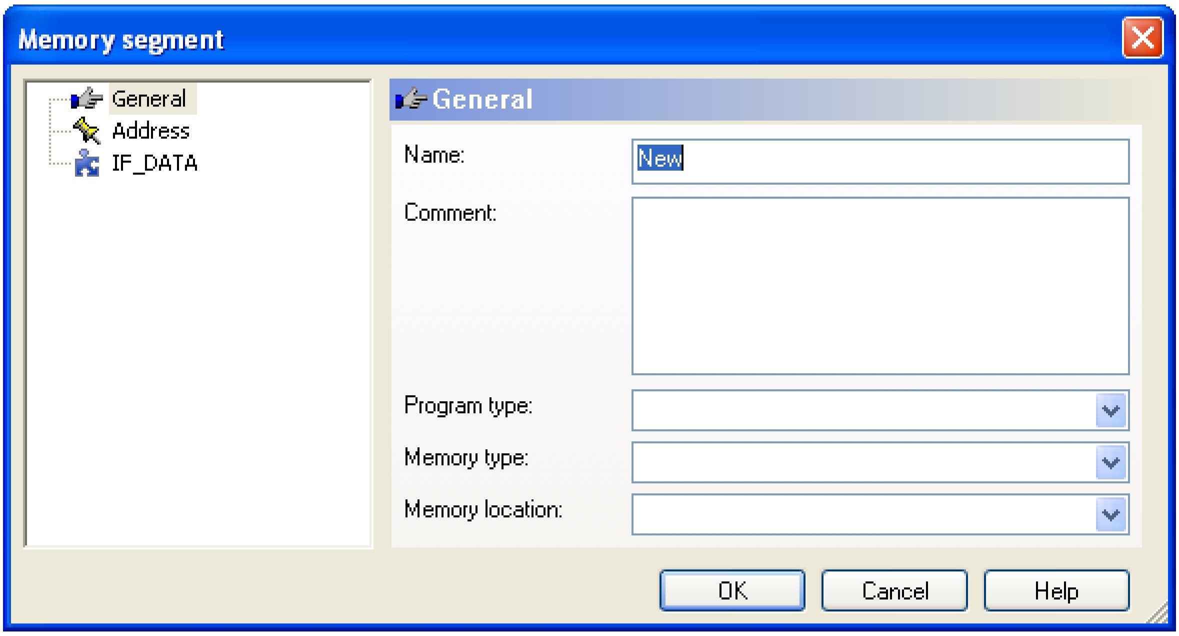

Figure 60: Dialog page "Memory segment – General"

On this dialog page the general parameters of a memory segment can be edited. behind the name and an optional comment the program type of the memory segment, the memory type in the ECU (hardware) and the memory location must be selected. The several settings are described in detail in the online help.

On the Address page (see Figure 61), the settings of the address range where the memory segment is located can be edited. The address range is defined by the start address and the size of the memory segment.

If copies of the memory segment are to be stored, appropriate offsets must be assigned to them The data "FFFFFFFF" (corresponds to decimal -1) means that no offsets are assigned (inactive).

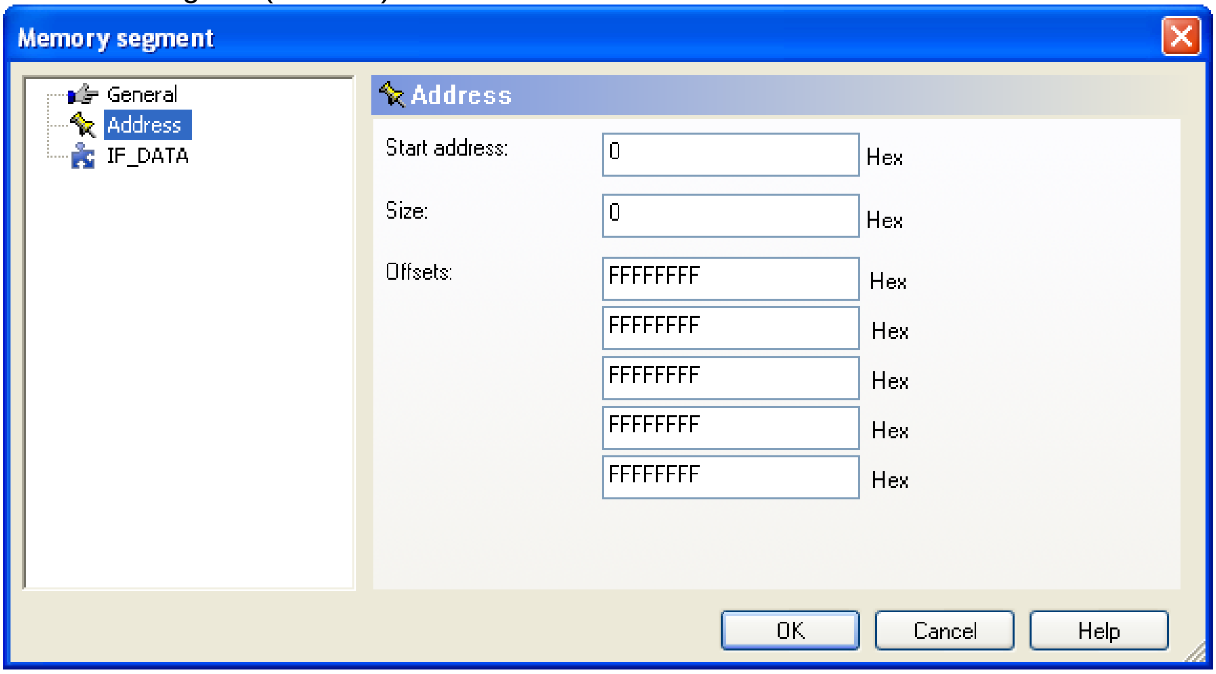

Figure 61: Dialog page "Memory segment – Address"

On the last dialog page (see Figure 62) all IF_DATA sections of the respective memory segment are displayed. Via the button **[Insert]** additional IF_DATA section can be added. A double click on the entry opens a dialog page to edit it in case it is of an editable data type. The data types of IF_DATA sections are explained in detail in the on online help.

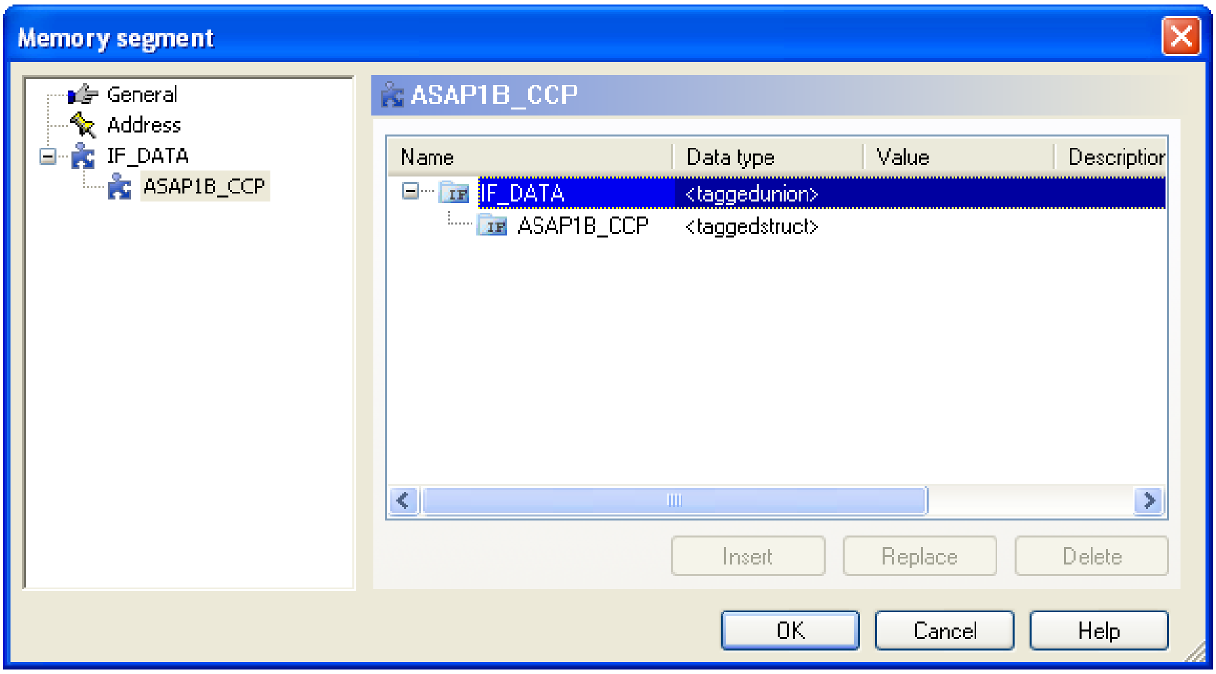

Figure 62: Dialog page "Memory segment – IF_DATA"

4.1.17 Frames

Frames are another structuring means, in addition to groups and functions. They are read in from the ASAP2 file and visualized in the ASAP2 Editor.

A frame is characterized by a name and a comment, a scaling unit, and a rate. It can be used only for grouping of measurement values (and thus not for grouping of calibration values). Frames can also contain IF_DATA information.

The frames are displayed in the tree view in the ASAP2 Editor. There is one main node for the frames – the frames themselves cannot be nested like groups and functions.

The menu command **Edit | New | Frame** opens the General page (see Figure 63):

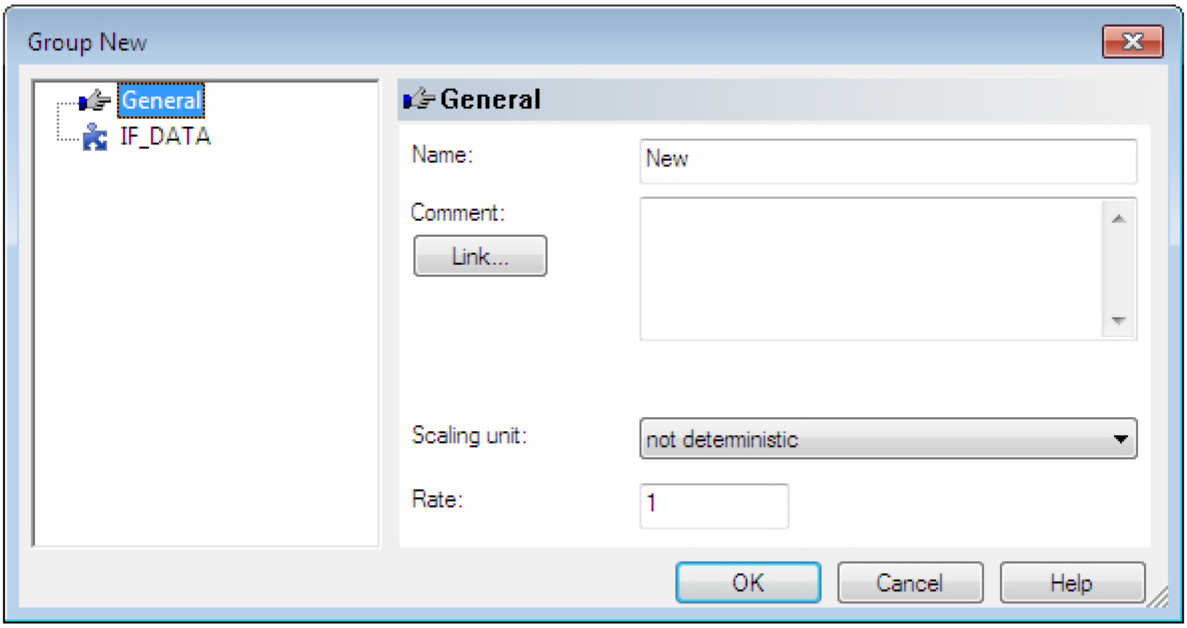

Figure 63: Dialog page „Frame – General“

On this dialog page you can input the name and an optional comment. The **[Link]** button next to the comment field opens a dialog in which a link to a file on the local hard disk or another database object in the current database can be made.

In addition the scaling unit and the associated rate are displayed. The scaling unit defines a measurement raster. At the moment this setting is read from the A2L file (when it is defined there) and displayed here but will not be evaluated. With the value of „rate“ the scaling unit is multiplied. This value is not evaluate by, too.

On the second dialog page (see Figure 64) all IF_DATA sections of the respective memory segment are displayed. Via the button **[Insert]** additional IF_DATA section can be added. A double click on the entry opens a dialog page to edit it in case it is of an editable data type.

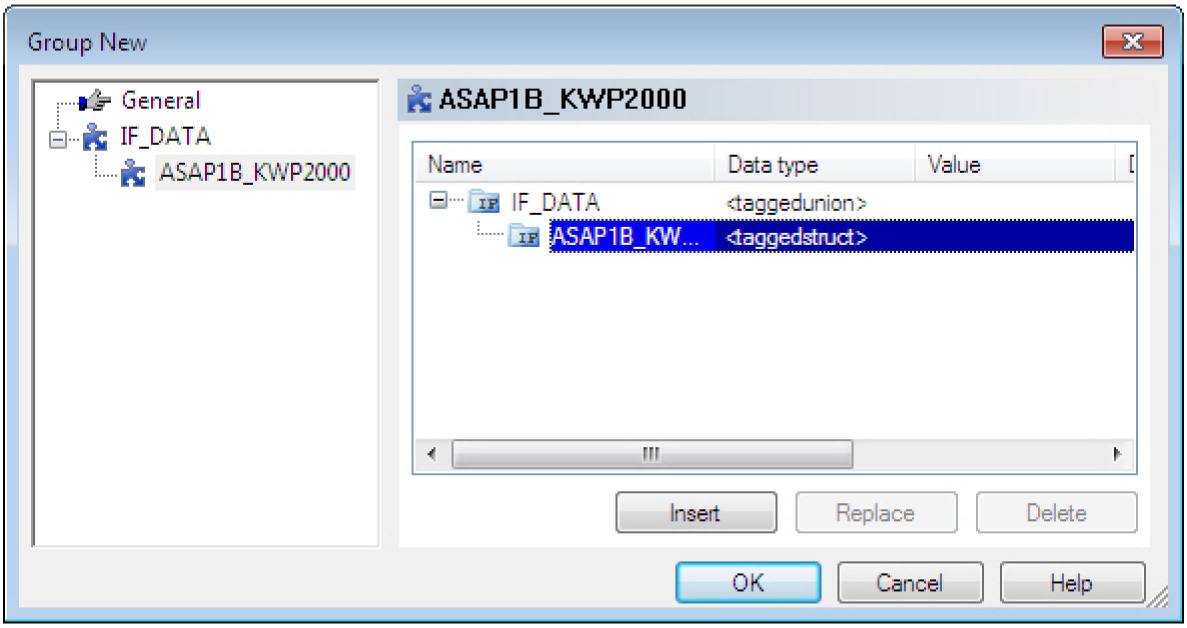

Figure 64: Dialog page „Frame – IF_DATA“

4.1.18 Partial Databases

In principle a partial database is a special display method of the database. So, when in a partial database a new conversion rule is defined it is available for the whole database. In the ASAP2 file the partial database is defined by an `INCLUDE` statement.

The menu command **Edit | New | Partial database** opens a dialog where the name of new partial database is to be set (see Figure 65).

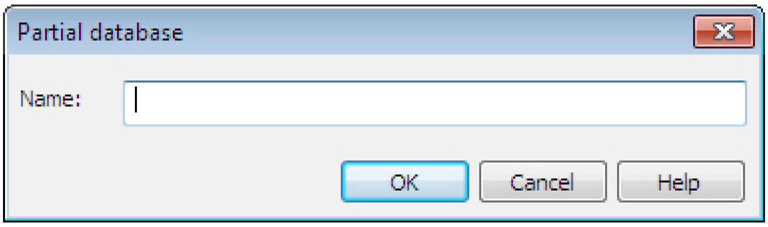

Figure 65: Add a new partial database

The new partial database appears in the tree view at the element Partial databases. Together with the new partial database the original database is displayed once more as a partial database.

Afterwards objects of the main database can be assigned to the partial database. A partial database can contain variables, conversion rules and record layouts but no groups.

4.1.19 IF_DATA Information

Beside general descriptions an ASAP2 file can contain additional interface specific information (e.g., driver settings). This information are described in a general (global) IF_DATA block in the header an in (local) IF_DATA blocks at every measurement and calibration object. Local IF_DATA blocks can also be defined for groups, functions and memory segments.

Depending on the interface type different data are needed. So, IF_DATA information are containers inside the ASAP2 file with supplements to the information in the ECU description file that have been defined in the ASAP2 specification. The syntax of this supplements is described in the AML file.

Info

Only in case when IF_DATA information are not used by the current device driver they can be created and edited in the ASAP2 Editor. Failing this the values must be changed in the Device Configuration dialog and are only displayed for information in the ASAP2 Editor.

In the standalone version of the ASAP2 Editor IF_DATA information can always be edited and changed.

The menu command **Edit | New | IF_DATA description** the respective popup menu in the **IF_DATA information** section of the tree view opens a selection menu that contains all available IF_DATA descriptions (see Figure 66).

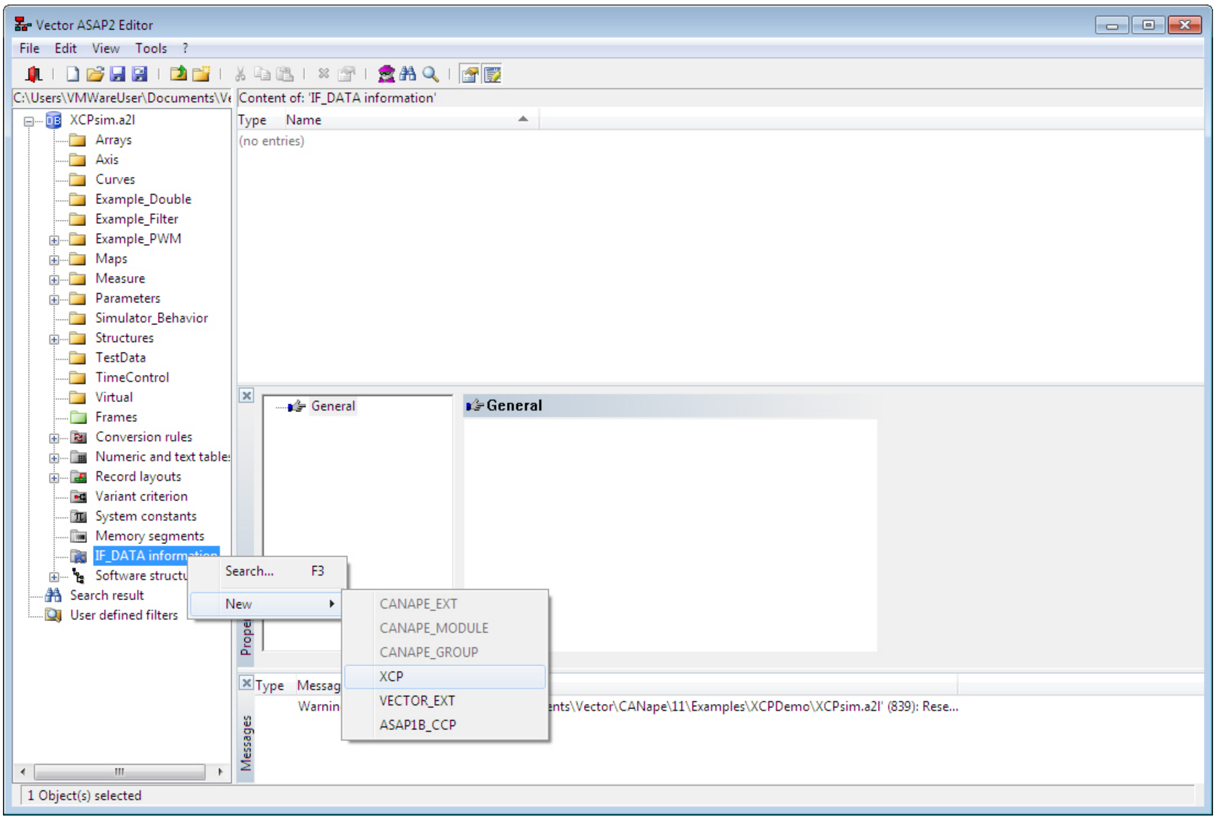

Figure 66: Selection of an IF_DATA description

A description can be inserted only once, additional descriptions can be added to the list by the menu command **File | Add AML**.

On the right part of the editor window the loaded IL_DATA information are listed. A double click on an entry opens the edit dialog (see Figure 67).

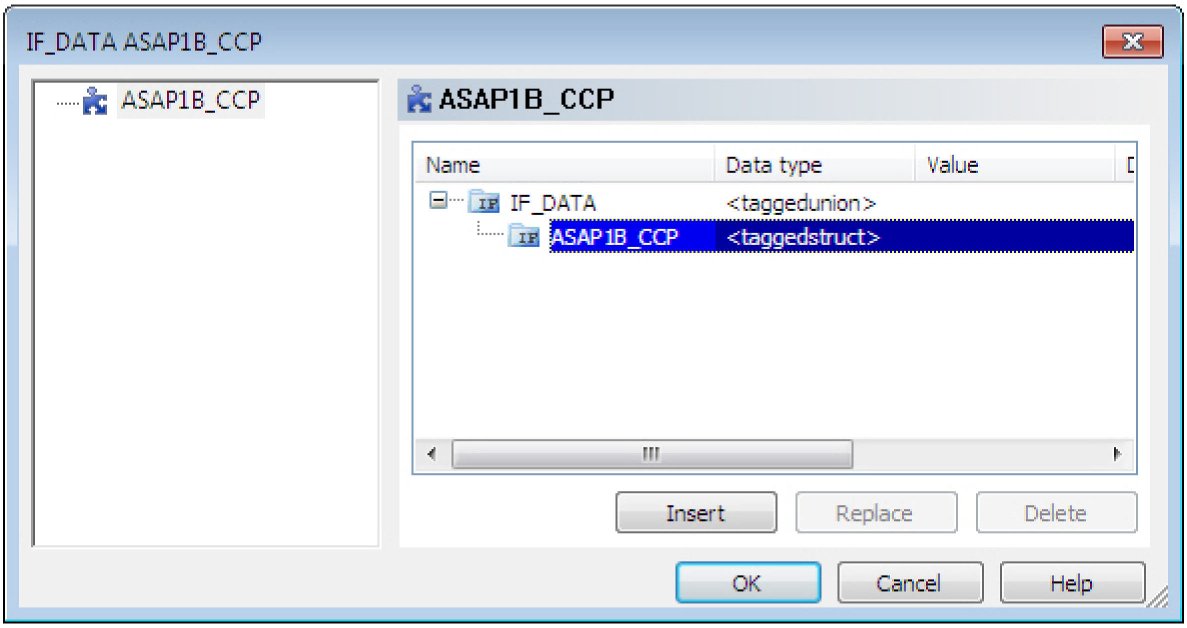

Figure 67: Dialog page "IF_DATA"

The elements of this dialog are described in detail in the online help.

4.1.20 Software Structure

A software structure (also called a data structure) consists of a block of database objects, whose address ranges are connected as a rule. When the name fragments are divided by a dot, the structure is build based on this names.

Example

```
structure.measure.cal1
structure.measure,cal2
```

builds the following structure:

```
structure
   measure
      cal1
      cal2
```

Because a software structure is based on the MAP file or the object names, it cannot be changed in the database editor.

The structure can be measured together, whereby the whole address range of all object is measured. In the measurement signal list the structure seems to be a single object, all measurement options (measurement mode, cycle time...) are valid for all objects contained in the structure.

In the Properties dialog all Elements of the software structure are listed (see Figure 68).

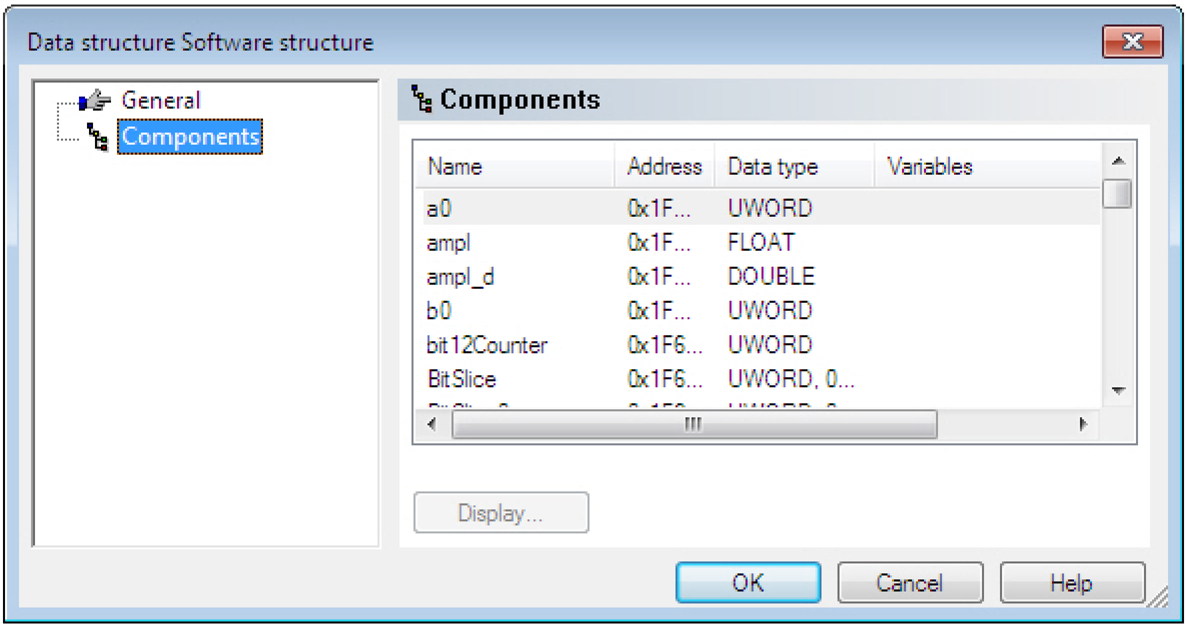

Figure 68: Elements of the data structure

Info

In the database editor several data structures from the folder Software structure can be copied into other groups (via popup menu or drag & drop).

When a sub-element is copied the name of its parent element will be prefixed, separated by a dot.

Doing so only a link to the original element in the data structure is created, the data structure remains.

So only such links can be deleted.

Using of data structures is in such cases useful when a lot of measurement objects are to be handled and they can be combined in structures.

4.2 Examples for Database Objects

In the following, we will show - based on several examples - how maps/characteristic curves have to be described in the controller database.

4.2.1 Example 1: Characteristic Curve, No Axis

The following table defines a characteristic curve with no axis:

X Axis	X0	X1	X2	X3	X4	X5	X6	X7
Value	11	12	13	14	15	16	18	22

These table values are stored in the controller and require 8 bytes for the data type `unsigned char`.

Characteristic curves are considered to be special maps, characterized by having the row count parameter set to the value 1.

Use the **record layout** drop down list to indicate how the map will be stored in the controller's address space. You must choose between a row-by-row, column by-column or a user-defined record layout.

For row-by-row storage in the controller's address space, first the individual function values are saved one after another for the first row, and then the function values for the second row, etc. With column-by-column storage, the controller's memory first contains the individual function values of a column, then function values of a second column, etc. For characteristic curves, row-by-row and column-by-column storage are the same.

The **X Axis** and **Y axis** selection list can be used to specify whether an axis should be assigned to the map as an X axis or a Y axis (see Example 2). Since in this first example a characteristic curve with no axis is being created in the controller data-base, the **No axis** entry is selected.

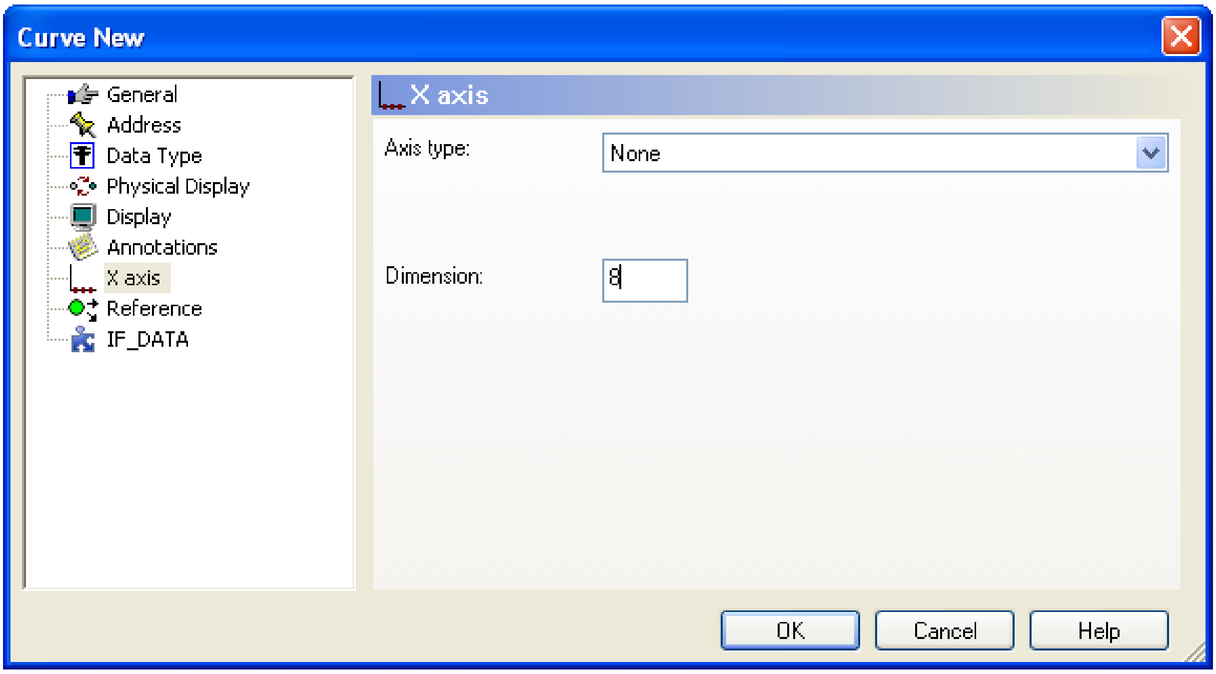

Figure 69: Dialog Curve1

Enter the number of curve values (here: 8) in the **Dimension** input field on the "X axis" dialog page (see Figure 69, . Since this is a curve, there is no "Y axis" dialog page.

Info
Since the name of a newly created database object is only accepted when the dialog is closed, the default name for a newly created data-base object appears in all screen shots in this section.

When the dialog is exited with **[OK]**, the ASAP2 Editor saves this characteristic curve in the controller database.

4.2.2 Example 2: Characteristic Curve With Fixed Axis

In this example, the following characteristic curve is to be defined with a fixed axis:

X Axis	26	28	30	32	34	36	38	40
Value	11	12	13	14	15	16	18	22

The table represented above is stored in the controller and it requires a total of 8 bytes for the data type `unsigned char`.

Add a new characteristic curve to the group of characteristic curves in the database (e.g. Curve2). Do this in a way analogous to Example 1. To assign a fixed axis to the X axis, the Fixed axis item must be selected from the **X Axis** selection list.

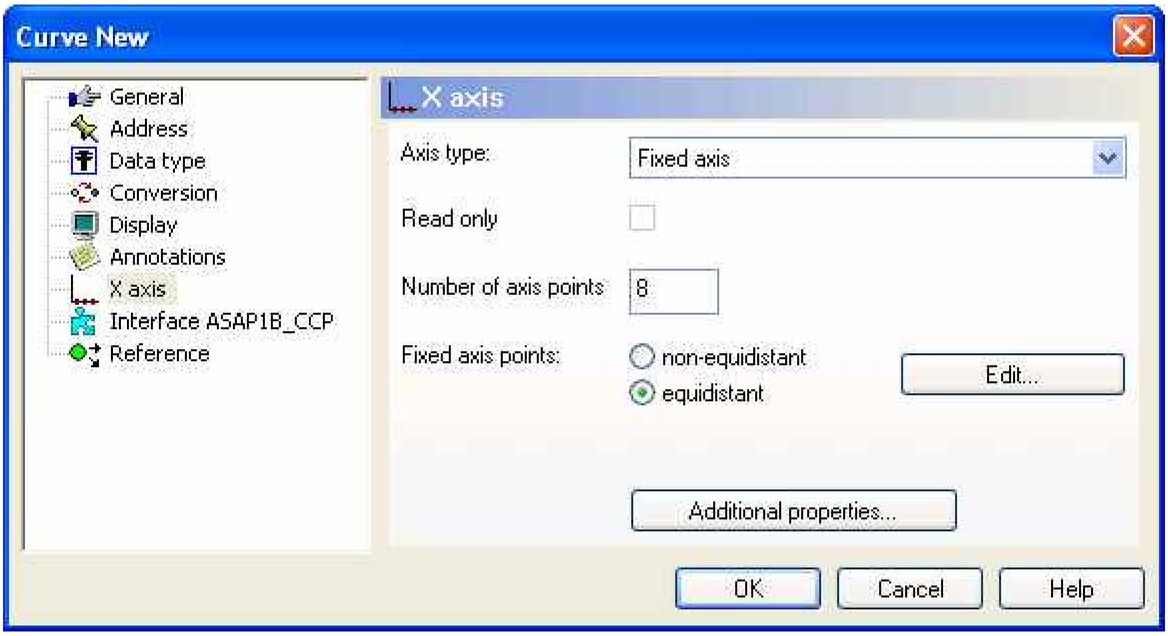

Figure 70: Configuration of Curve2

Clicking the button **[Additional properties]** opens a dialog with several pages about the X axis coordinates, where you can configure the fixed axis (see Figure 71, Seite 77). In this dialog you can assign an input quantity to the fixed axis by clicking the **[Select]** button.

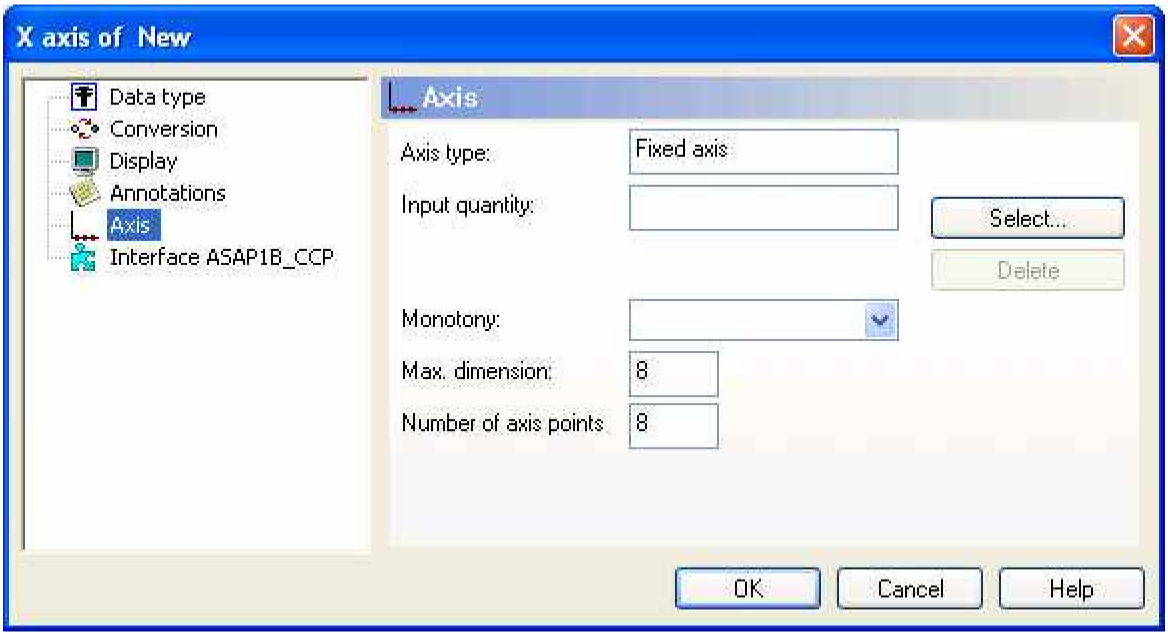

Figure 71: Axis coordinates dialog

Fixed interpolation points can be defined as equidistant or not equidistant. If the fixed interpolation points are equidistant, the start value (**Offset**) and step width (**Distance**) can be set as desired in a subordinate dialog. If not equidistant fixed interpolation points are used, they can be defined with the **[Edit]** button in a table.

Since this example begins with the value 26 instead of 0, and the spacing between two interpolation points is two steps, the fixed axis type is equidistant. In the **Fixed axis points** section, select the type **equidistant** and click on the **[Edit]** button. This opens the **Fixed Axis** dialog, in which the offset (here: 26) and the distance (here: 2) can be entered in their respective input box.

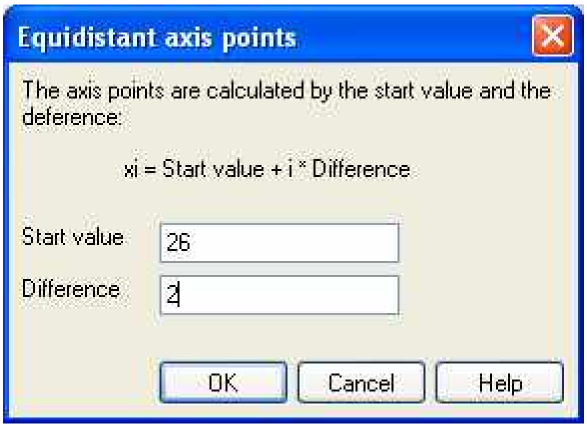

Figure 72: Fixed axis parameters

After you have exited the **Axis Coordinates** dialog with **[OK]**, you will return to the **Curve** dialog.

After you have also exited this dialog with **[OK]**, the curve is stored in the ECU data-base.

4.2.3 Example 3: Characteristic Curve With Shared Axis

In this example, the same values are used as in Example 2. However, the interpolation point values are defined for a shared axis.

X Axis	1	2	3	4	5	6	8	12
Value	11	12	13	14	15	16	18	22

Create a new group, in which shared axes can be stored. Create a new shared axis with the name **Axis1**.

The settings made in the dialog **Shared axes** (see Section 4.1.6, "Shared Axes") largely correspond to those of the **Curve** dialog (see Example 1). On the **Axis** dialog page the **[Select]** button can be used to assign an input quantity to the shared axis.

Enter 8 as the number of X values for the axis in the field **Number of axis points** and close the dialog with **[OK]**. KL

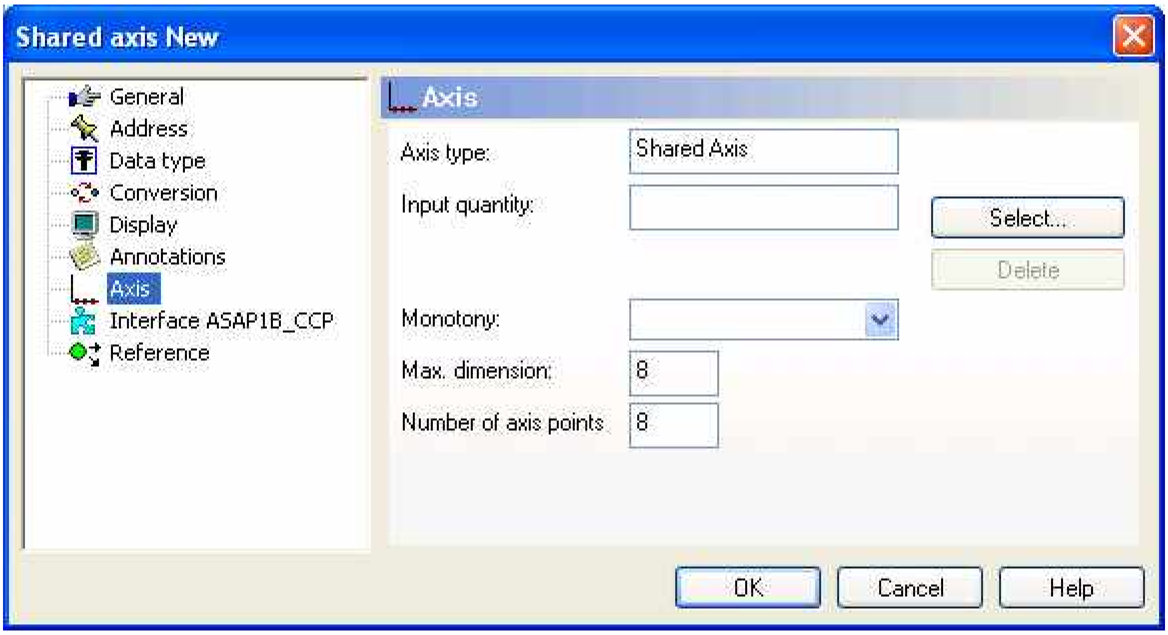

Figure 73: Dialog "Shared axis - Axis"

Now add a new curve to the database (e.g. Curve3), proceeding as in Example 1. Select the axis type Shared axis on the **X axis** dialog page. This opens a dialog in which all of the shared axes defined in the ECU database are displayed in a list. Select the previously created shared axis **Axis1**, to assign it to the X axis of the curve. After you have exited the dialog with **[OK]**, the curve is stored in the ECU database.

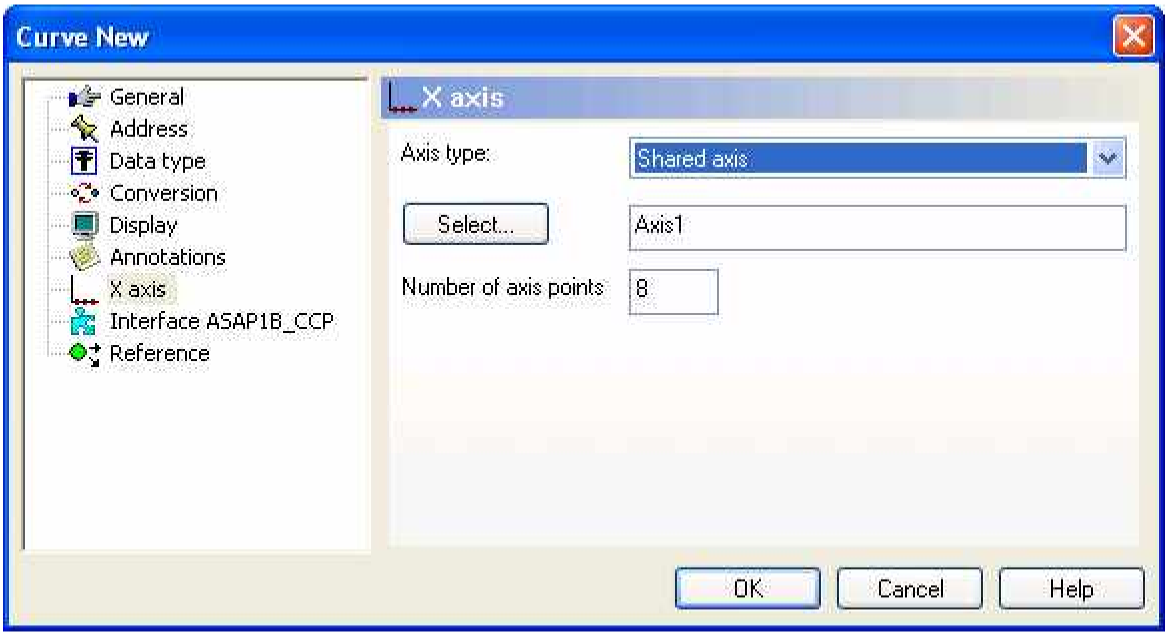

Figure 74: Configuration of the curve Axis3

4.2.4 Example 4: Characteristic Curve With User-defined Record Layout

In this example, the X axis point values are saved in the controller with the characteristic curve values, and together they are considered to be an object. In contrast to shared axes sampling points, they do not have their own address.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Values in ECU	0	2	4	8	12	14	16	20	1	3	8	9	11	7	6	7
	Axis points								Values							

To interpret such tables, so-called record layouts (controller program structures) in the controller database are required.

Add a new curve to the database (e.g. **Axis4**).

On the dialog page **Data type**, select the (below in Example 5) newly created record layout **KL_Layout1** from the **Record layout** selection list. Exit the dialog with **[OK]**.

Maps can be configured in a similar manner.

Detailed explanations of these dialogs can be found in the Online Help.

4.2.5 Example 5: Record Layout

The menu command **Edit | New | Record layout** lets you define a new record layout with the name **KL_Layout1** (see Section 4.1.13, "Record Layouts

"). The table described in Example 4 (Section 4.2.4, "Example 4: Characteristic Curve With User-defined Record Layout") occupies a total of 16 bytes in the ECU memory; the data type of the axis points and function values is unsigned char. The X axis points should be located at the beginning of the table (Index 0...7) and the function values at the end of the table (Index 8…15).
Enter a name for the new record layout to be added in the **Name** input field. Since we are working here with a record layout for curves with a constant number of X sampling points, you have to check the option **Constant X axis points**. For our example, enter the number 8.

The alignment values specify the addresses where the values for derived parameters begin – depending on data type. A value of 2 means that e.g. the values only begin at even numbered addresses. A value of 4 means that the values only begin at addresses divisible by 4, etc. Memory areas within the parameters may be skipped. The dialog page should now look like Figure 75.

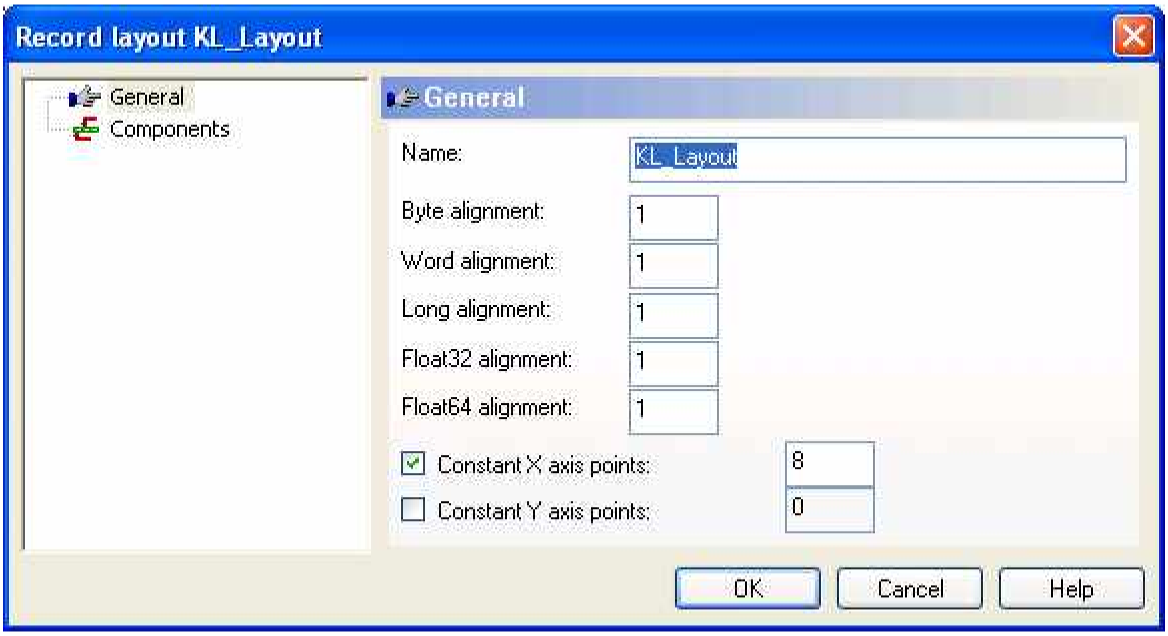

Figure 75: Configuration of an user defined record-layout

The **Components** dialog page is where the user selects an element type of record layout. An Element type describes the semantics of the record layout. Clicking the **[Insert]** button opens a list of predefined values, and selecting one opens a properties dialog to set and display its attributes.

In our example we only need two of these element types: the X axis points and the function values.

Since in our example the X axis points will be stored in the table first, they will be defined first (see Figure 76, Seite 81).

Info

The position of an entry can be changed via the and buttons.

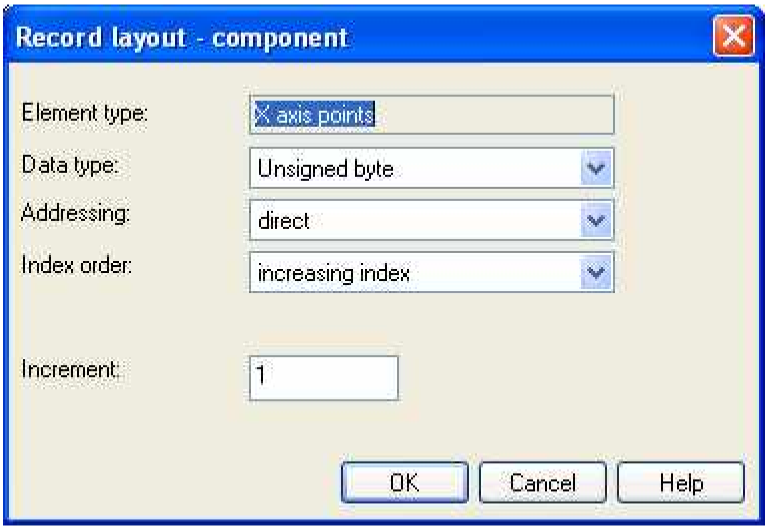

Figure 76: Configuring the attributes of a record layout element

For our example the default values can be used.

Click the **[Insert]** button again to expand the record layout with the element type **Function values**. Again accept the default settings and exit the dialog with **[Ok]**. More specific details about the various different element types can be found in the Online Help.

The record layout has now been completely configured. Exit this and the main dialog with **[Ok]**.

4.3 Special Dialogs

The following functions can be reached from the main menu item **Tools**:

- Search (on page 82)
- Editor settings (on page 87)
- Settings database/MAP file (on page 88)
- Statistics (on page 93)
- Plausibility check (on page 94)
- Compare (on page 97)

4.3.1 Search

You can search through the entire database for individual variables with the Search function. The function is available both via the popup menu and the function key <F3> as well as via **Tools | Search**. The inputs are made in the dialog window in Figure 77, Seite 82:

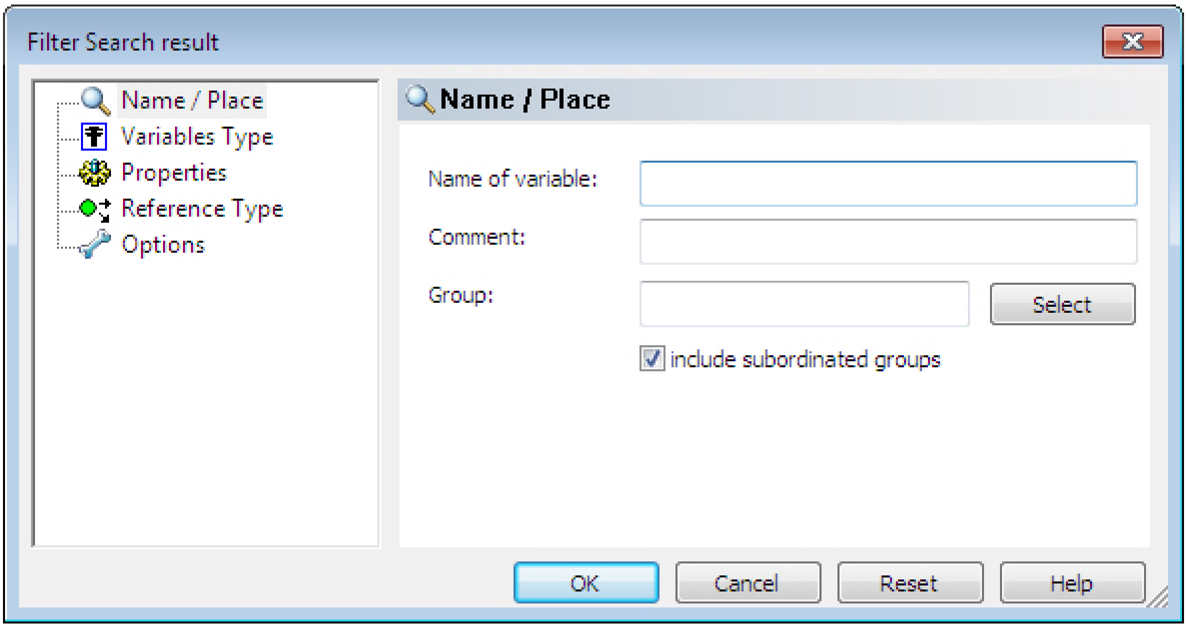

Figure 77: Search Function in the Database

Five tabs delimit different areas of the search.

Info
The options selected under each tab are all simultaneously active during the search.

In the tree view the filter Search result becomes active. Here the current search settings can be saved as filter and used again when necessary. To display all objects of the database the node with the database name (at the top) must be activated.

The objects in the list view right can be processed just as they can in the Database Editor (e.g. converted, properties changed, deleted). The list of results can also be sorted by column. To do this, click with the left mouse button on the desired column heading. Another click inverts the sorting order. A pin in front of the address of a database object means that the object was inserted into the database from the Linker MAP file.

Name/Place Page

On this page a search can be conducted by name or comment about the individual database objects. The wildcards " * " and " ? " can be used. Empty input fields mean the same as inputting " * ", i.e. there are no limitations.

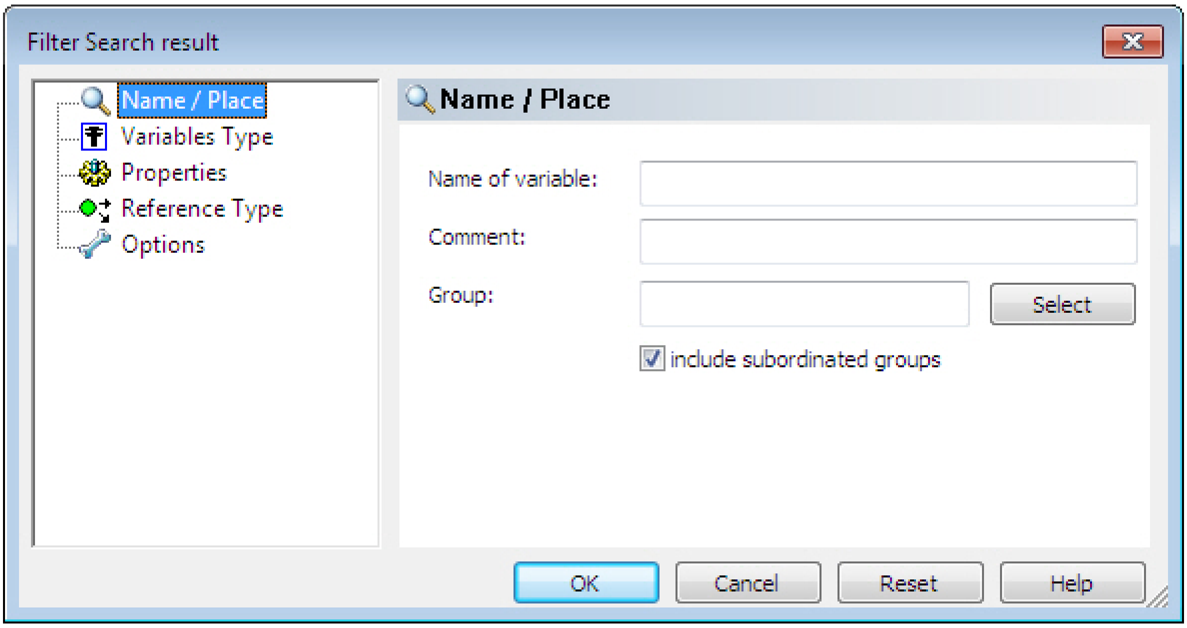

Figure 78: Search for database objects by name

The **[Select]** button opens the **Select group** dialog box (see Figure 79). Here the search can be limited to individual groups. When a group is selected in the database editor it is automatically entered in this field.

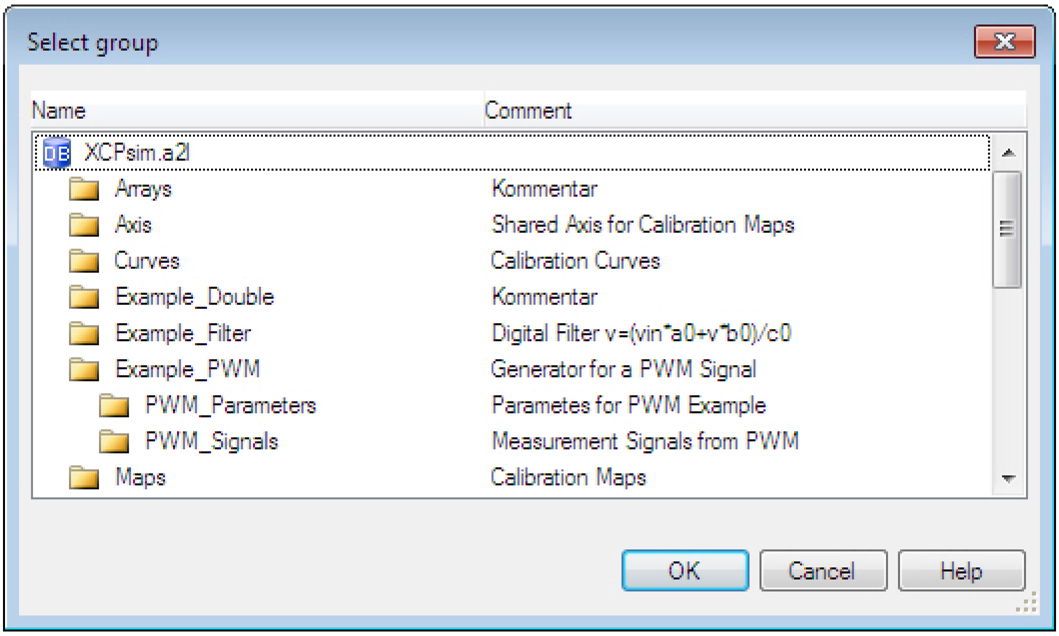

Figure 79: Group Selection

Info
If the entire database is to be searched, the database, not a group, must be selected in the **Select group** dialog. When the database is selected, no entry appears in the **Group** field.

A search of all objects in the entire database results in a display of all available variables in a complete overview (without subdividing them into groups).

Variables Type Page

On this page the search can be limited to certain specific variable types. A search can be conducted for all variable types in the database. Virtual signals and parameters can be selected on the **Properties** tab page.

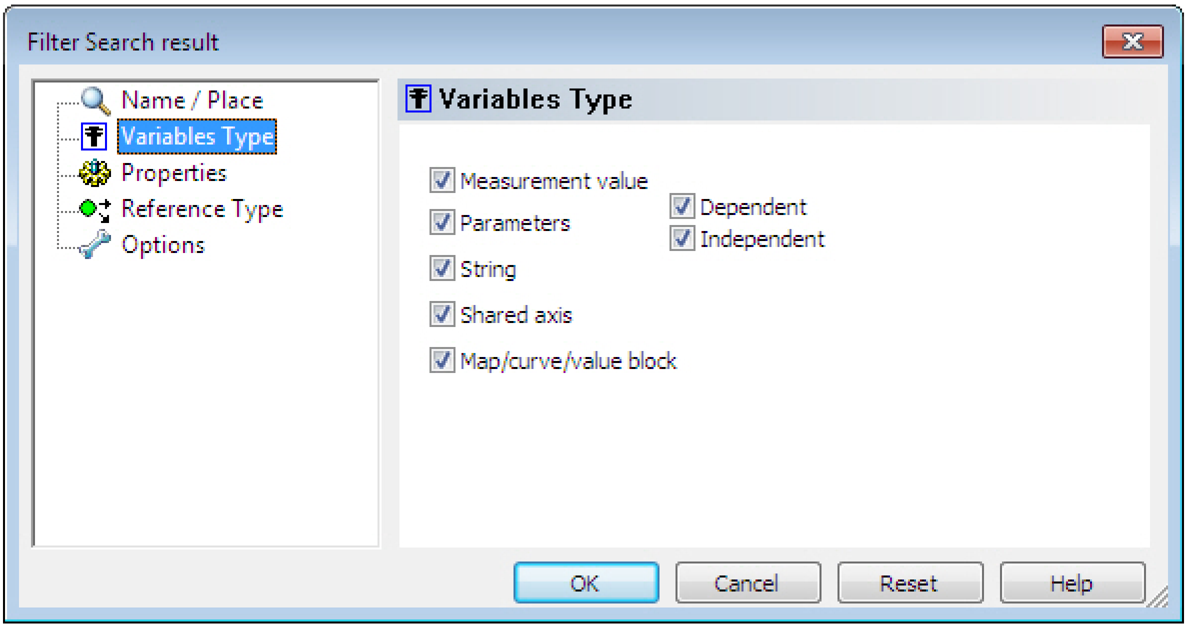

Figure 80: Search by variable types

If the search is not limited to any specific variable type, all checkboxes must be activated.

Info
The property **Dependent** is meaningful only for parameters

Properties Page

On this page the search criteria are put together according to various object properties. The address range for the search can be specified in the two input boxes near the top, **Real, address (hex)** and **to**. If a value is only entered in the first box, the program will only look for variables at that one address. If a value is entered in both boxes, variables will be looked for whose address lies within the range set by both of these values. If you do not wish to limit the address range, both of these boxes must remain empty.

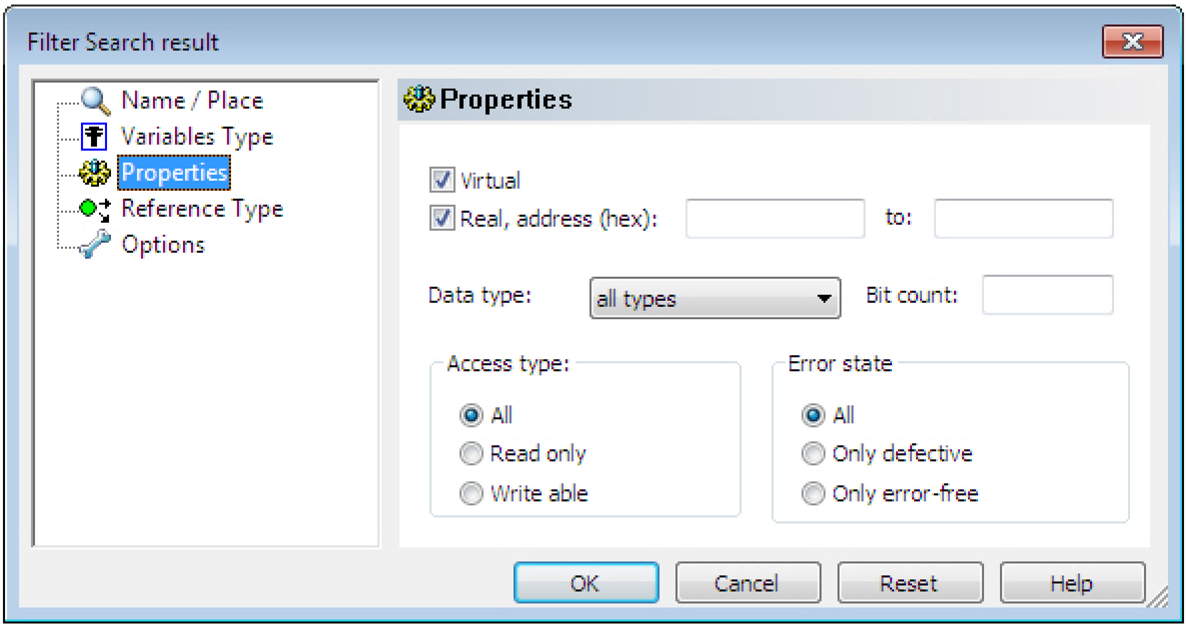

Figure 81: Search by properties

The options for **Access type** make it possible to limit the search to measurement or calibration objects. These settings have the following purpose:

All	Unlimited access
Read only	Finds calibration objects in the Read Only mode and measurement objects without Read/Write mode
Write able	Finds calibration objects with inactivated Read Only mode and measurement objects with the Read/Write mode

Info
The **Virtual** checkbox allows a search for virtual parameters and measurement values in connection with the options on the **Variable type** tab page.

Reference Type Page

Use options for database objects that are summarized in Function groups can be entered on this page.

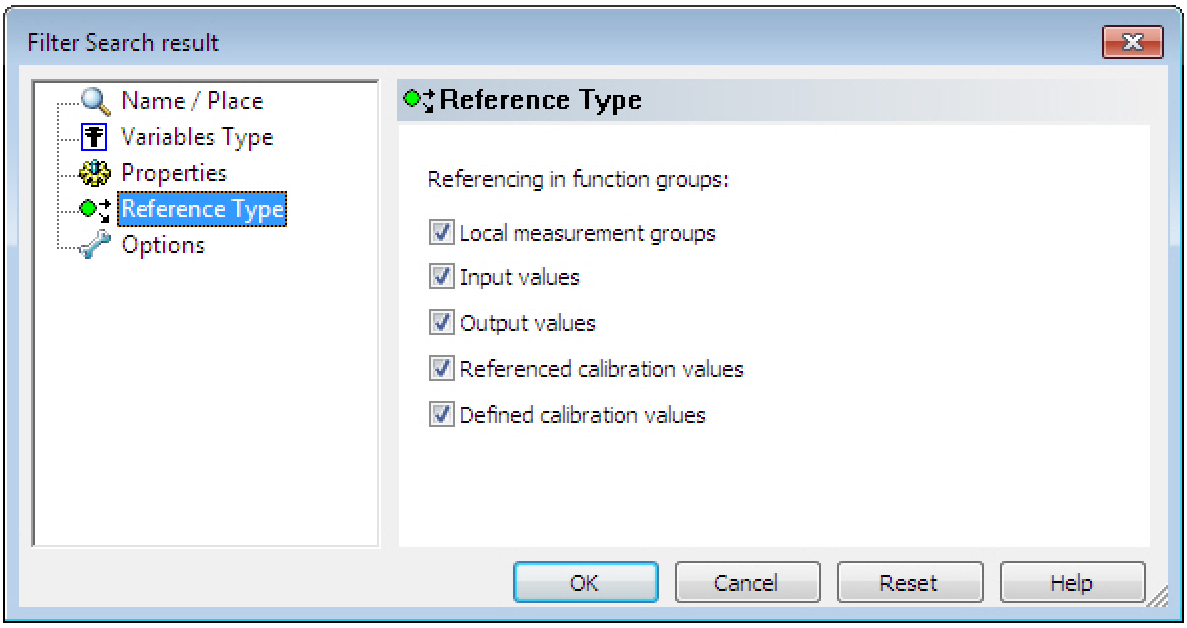

Figure 82: Selection based on use

At present only local measurement values and defined parameters can be created in the Database Editor. If in the ECU database function groups with input and output values or referenced parameters are defined, they can be selected here. Groups and function groups can be transformed one to another and vice versa. During this procedure all measurement values defined or converted measurement values get the property "local" and all parameters get the property "defined".

Info
The search via this tab involves only function groups. Database objects in groups are all shown. That's why it is recommended to specify the search in the tab **Name/location** to the respective function group.

Options Tab

Additional search options can be entered on this page. The **Enable case sensitive text search** checkbox only influences the search criteria **Name** and **Comment**. For database objects assigned to more than one group at any given time the option **List variables separate for every group** can display one line per group in the result list. If this option is not activated, only one line is displayed for such database objects and all groups to which this variable is assigned are listed in the **Group** column one after another separated by a comma. When the option **Start search after opening this dialog** is activated, clicking the search function immediately starts browsing with the settings from the last browse.

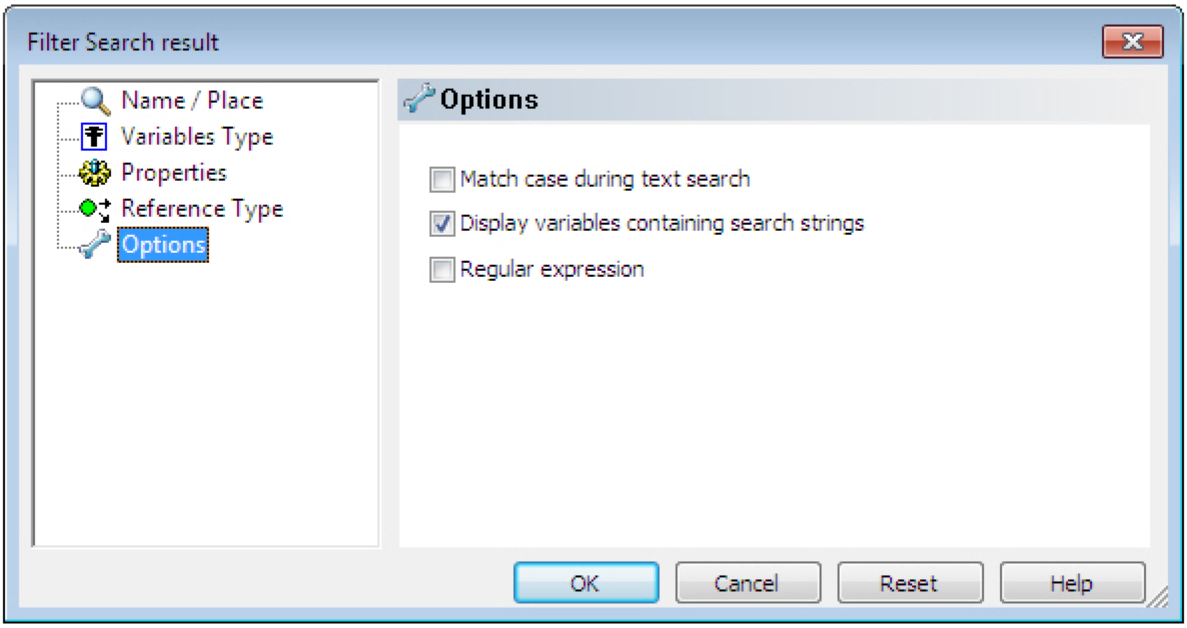

Figure 83: Search - Options Tab

4.3.2 Editor Settings

Selecting the menu item **Tool | Editor settings** opens the **Options** dialog. On this dialog you can display and edit general management and project description data about the database editor (see Figure 84).

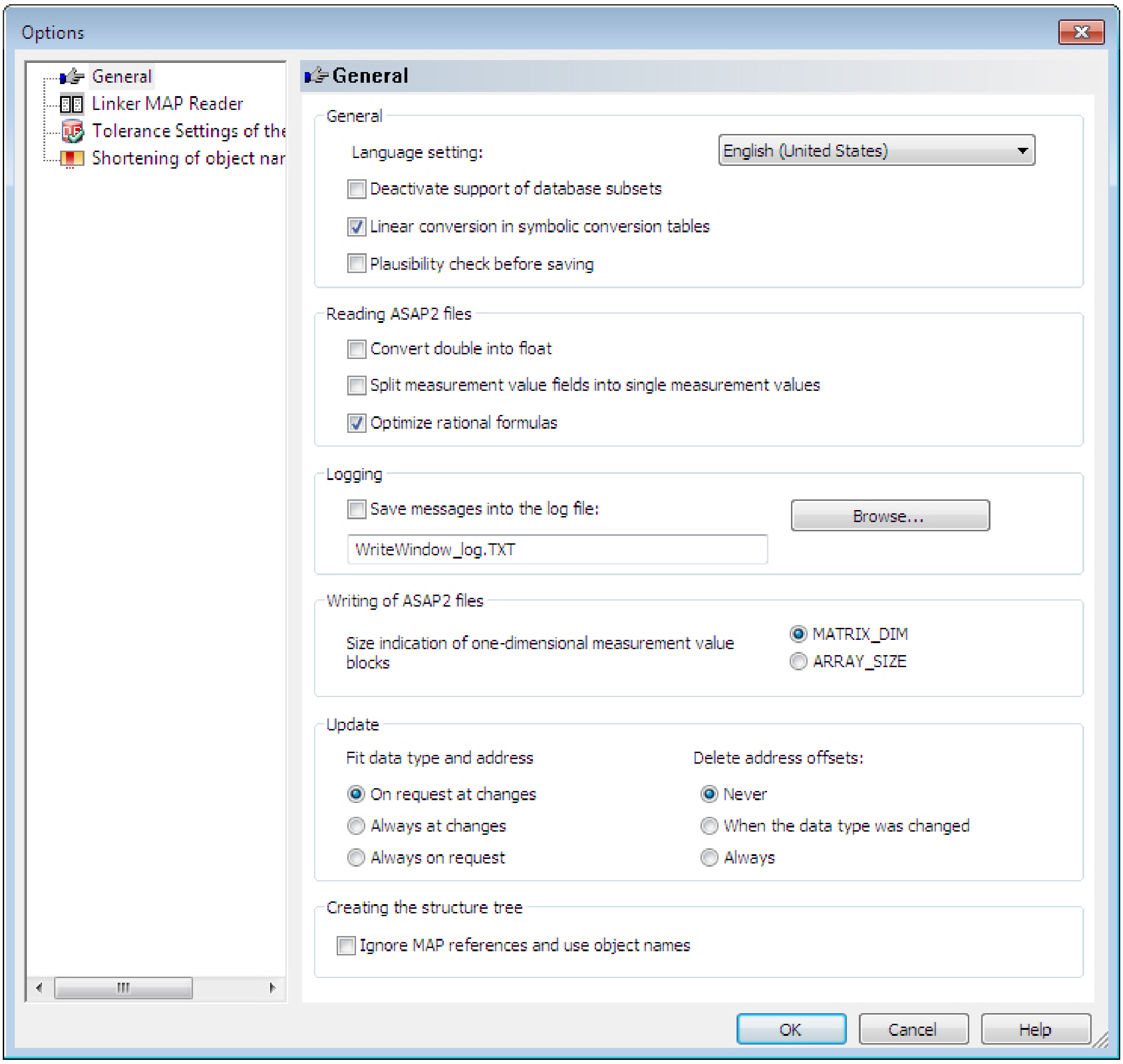

Figure 84: Options dialog

Info

The dialog is described in more detail in the online help. The page "Shortening of object names" is only displayed in the standalone version of the ASAP2 Editor, otherwise it is displayed in the CANape Option dialog.

4.3.3 Settings Database/MAP File

Selecting the menu item **Tools | Settings database/MAP file** opens the first page of the Properties dialog for the whole database (see Figure 85).

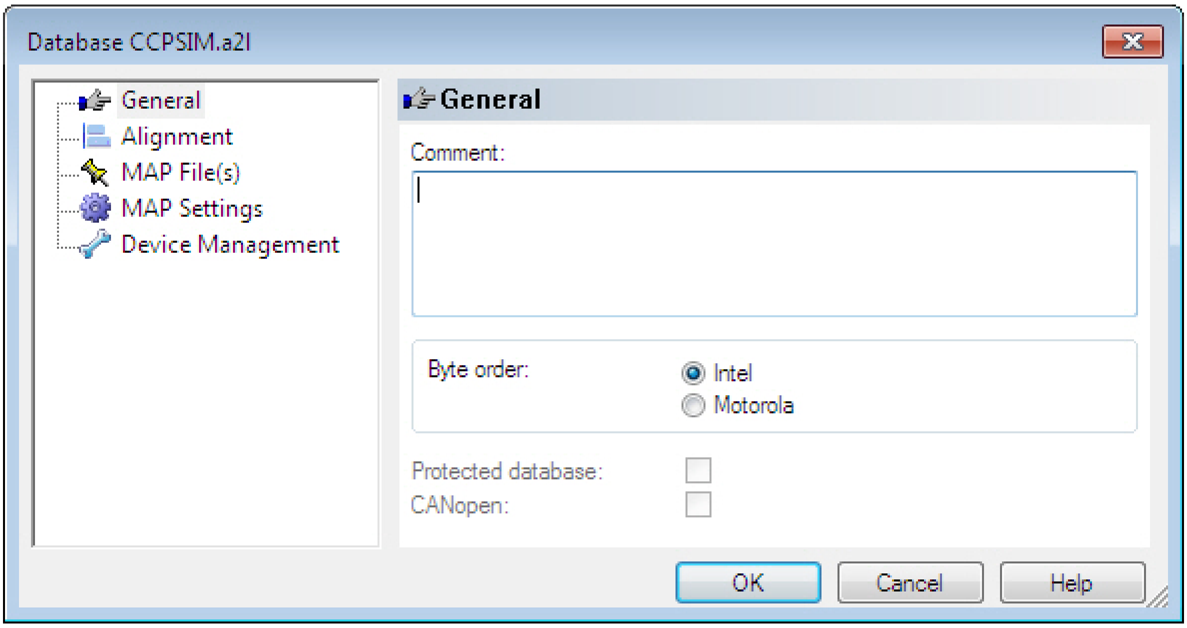

Figure 85: Settings MAP file, General

Besides of a comment you can set here the default byte order.

On the page **Alignment** (see Figure 86) are the addresses set at which the values of a derived parameter start, depending on the data type.

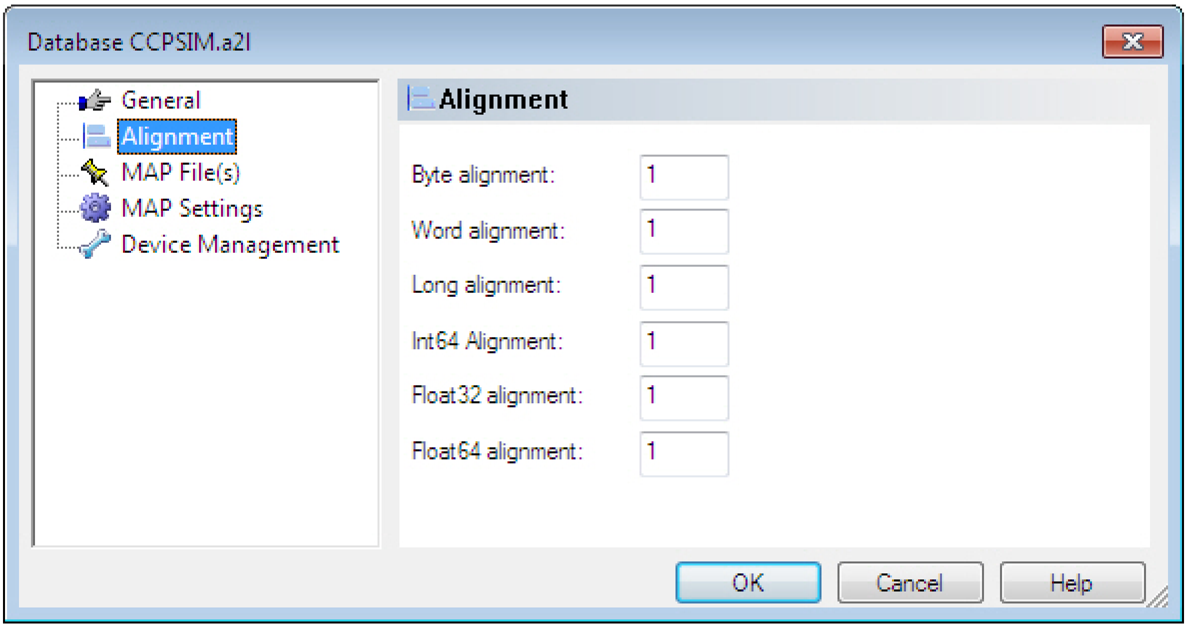

Figure 86: Settings MAP file, Alignment

The value "2" means that the values start only at even-numbered addresses. The value "4" means that the values start only at addresses divisible by 4, and so forth. Consequently, memory areas within the parameters are skipped, if necessary.

On the page **MAP file(s)** you can set the respective assignments (see Figure 87).

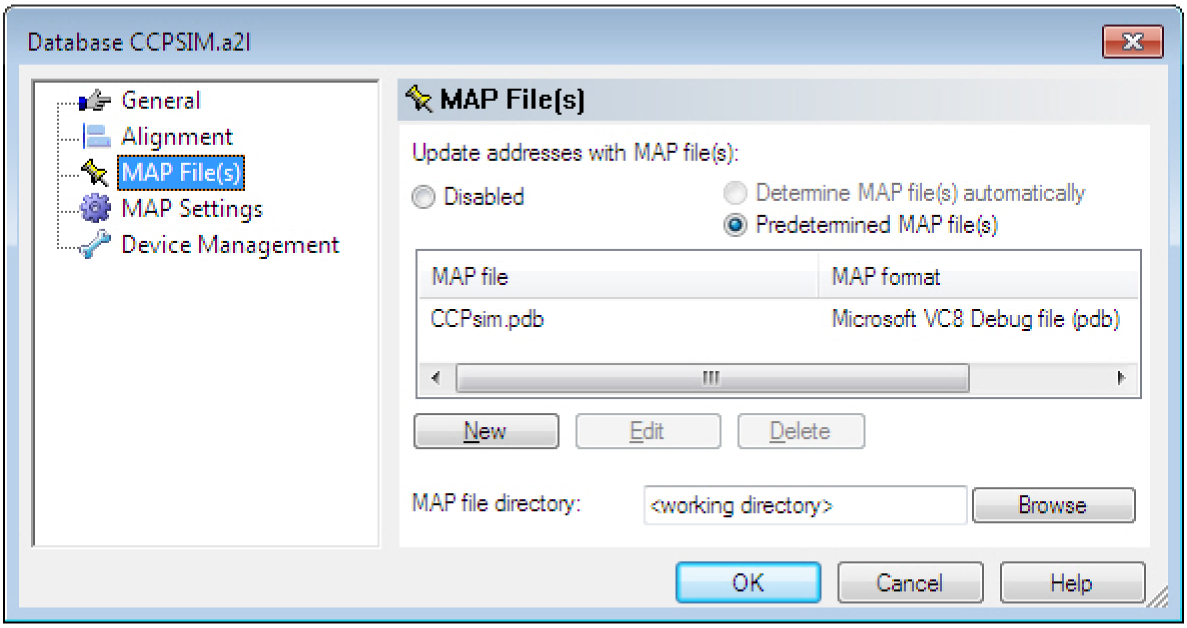

Figure 87: MAP files

The button **[New]** opens the MAP file definition dialog (see Figure 88). The format of the Linker MAP file can be selected from the drop-down list.

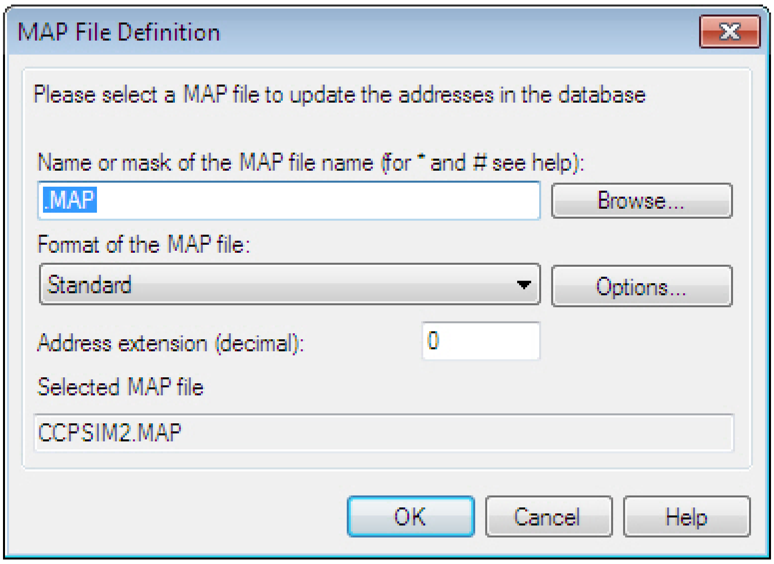

Figure 88: Definition of a Linker MAP file

In the field Format of the MAP file all Linker MAP file formats are listed that are supported by the AASP2 Editor.

Please contact Vector Informatik if your Linker MAP file format is not listed here.

Usually, the Linker MAP file has the same name as the database, but it is possible to assign a Linker MAP file with another name to the database. The name of the Linker MAP file is entered in the **Name or mask of the MAP file name** input box.

Info
If the database is renamed or moved to another directory, the Linker MAP file may also need to be renamed or moved.

If multiple processors are in one ECU, each with its own memory, each program receives its own Linker MAP file. In this case, the file names must be extended by the processor number. Masks can be defined for customizing the file names. If the file names have the character #, the corresponding characters are taken from the name of the ECU database. If a mask contains the character *, then the entire database name is inserted at this position.

Example
Name of database: `ABS_Rel303.A2L`

File name extension: `.MAP`

Mask: **_1* -> generated Linker MAP file name: `ABS_Rel303_1.MAP`

Mask: *MAP#######_1* -> generated Linker MAP file name: `MAP_Rel303_1.MAP`

For differentiation, the symbolic variable names in the database are extended by the character _C<n>, where n is the processor number. The addresses of the objects likewise receive an address extension in the ECU in the form <n>:aaaa, where n is the processor number and aaaa the address of the variable.

On the page **MAP Settings** (see Figure 91) the database file settings can be changed concerning the MAP file. The default settings in this dialog are set in the INI file.

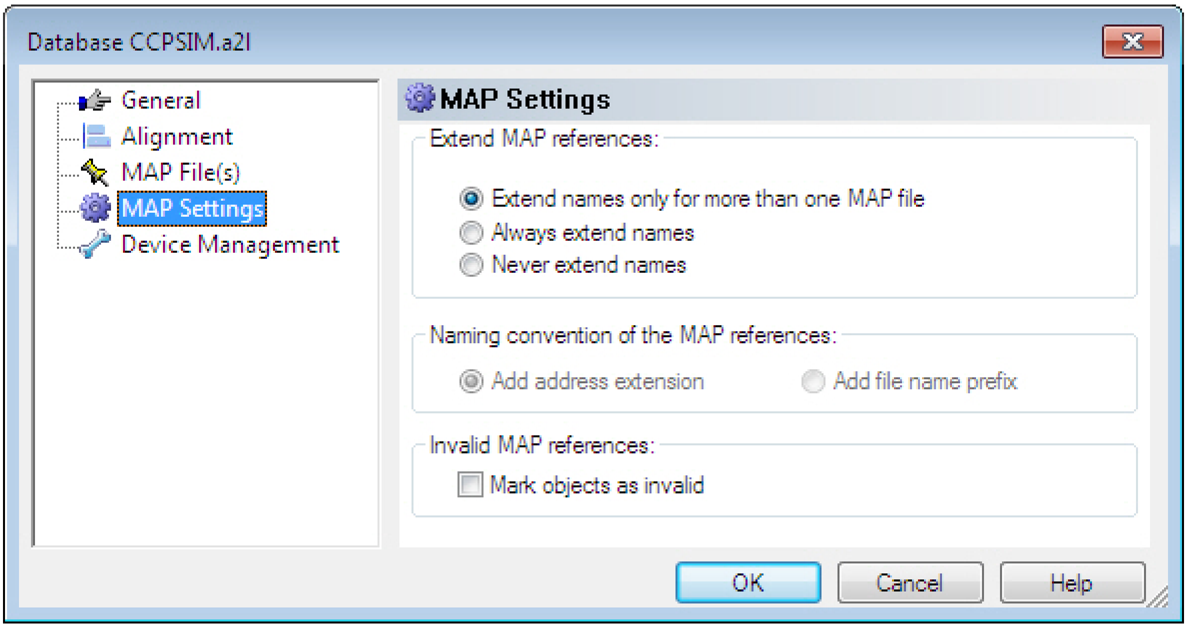

Figure 89: MAP settings

This settings are explained in detail in the Online help.

On the Device Management page (see Figure 90) you can display and edit information for identifying the current database.

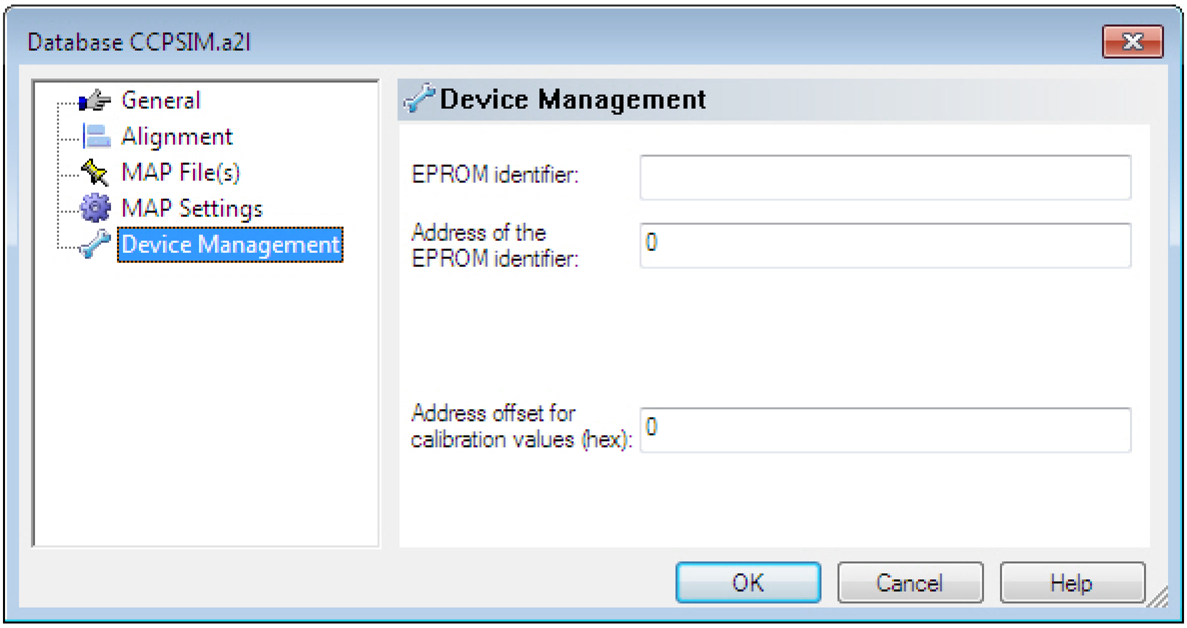

Figure 90: Settings MAP file, Device Management

Info
The option **Address offset for calibration values** is only available for calibration values (of type CHARACTERISTICS), but not for measurement values.

MAP File

This dialog is opened if the **[Link]** button on the **Address** page is clicked when creating or configuring database objects (e.g., parameters or measurement values) .

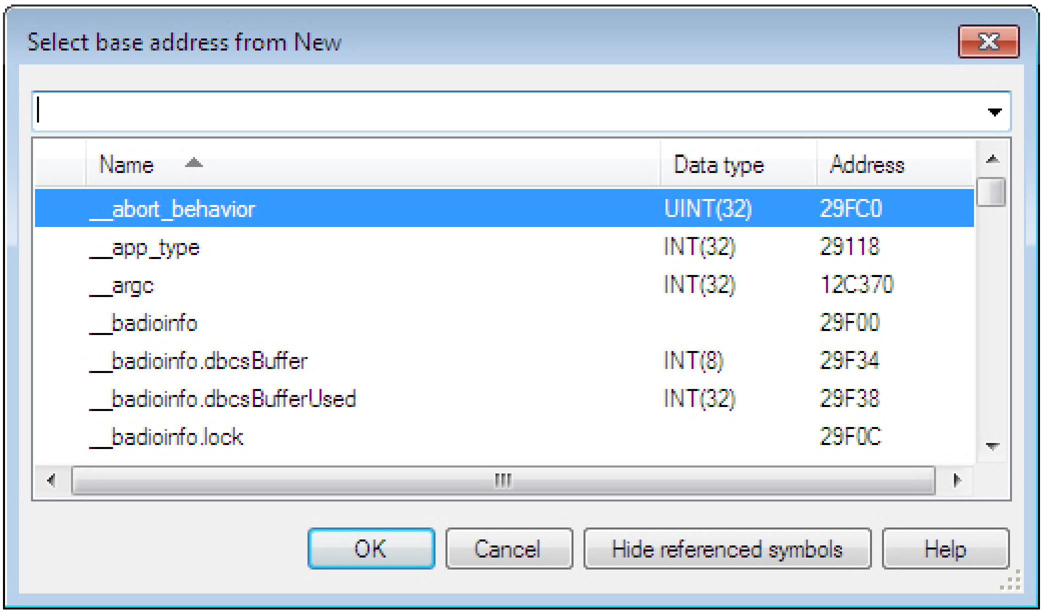

Figure 91: MAP file dialog

All available variables in the ECU are listed here. If two or more Linker MAP files are available (when there are two or more ECU processors), the variables from all the MAP files are listed. The variable names are extended by the characters _Cn (where n is the processor number) to differentiate between them.

The list can be sorted by each column. To do this, click on the corresponding column header. A second click inverts the sorting order. You can also search for words or numbers within a column by inputting the desired character string via the keyboard. If there is a pin in front of the address of a Linker MAP file entry, this means that the object is already being used as a reference by a database variable and should not be used again. The button **[Hide referenced symbols]** turns off the display of these entries.

If no group has been selected in the database editor, the variable is accepted into the database by double-clicking it. If a group has been selected, there will be a confirmation prompt as to whether the variable should be inserted into this group. If you say no, the variable will be inserted directly into the “root” of the database. It is also possible to select multiple variables and insert them into the database.

Before the variable or variables are inserted, you are asked whether the variables are to be inserted as measurement values or parameters.

Info
If a variable is to be added to the database a second time, the existing variable must be renamed, because each variable name may only be found in the database once.

4.3.4 Statistics

Selecting the menu item **Tools | Statistics** displays information about the database objects, see Figure 92, Seite 94. The window only gives information. No input is possible, but highlighted elements can be copied via the popup menu **Copy** command onto the clipboard for further processing, e.g. in a word processor.

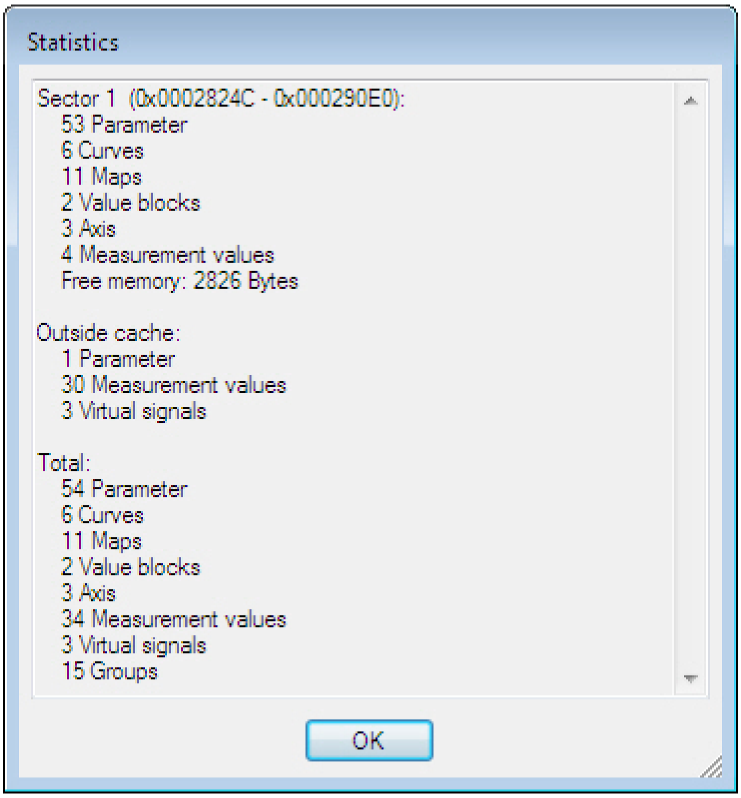

Figure 92: Statistics

4.3.5 Plausibility Check

Selecting the menu item **Tools | Plausibility check** opens the dialog seen in Figure 93, so that the database can be checked for consistency of various factors.
In addition the program checks to see if the record layouts are appropriate for the corresponding objects.

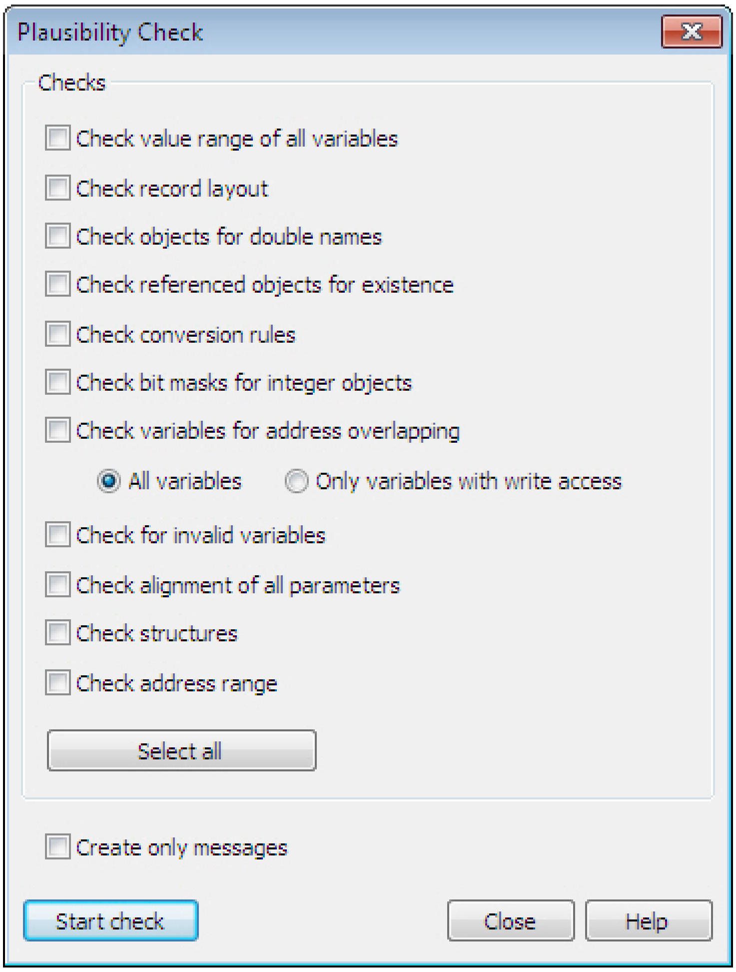

Figure 93: Options of the plausibility check

The individual options in this dialog are explained in detail in the Online Help. Clicking the button **[Start check]** executes the check. In case inconsistencies arise from the check, they can be removed or ignored (see Figure 94). When the option Create only messages is activated, the inconsistencies are only shown in the Write window. Via the respective link in the Write window the "Properties" dialog of the respective database object is opened. Changes in the database will be not done.

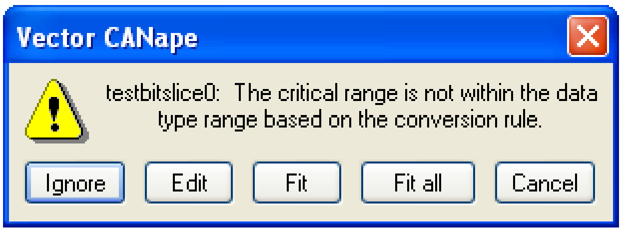

Figure 94: Results of Plausibility check

Selecting the options **[Fit]** or **[Fit all]** the inconsistencies are removed when it is possible. The correction mechanisms are explained in the online help (index entry "ASAP2 Error").
With the exception of the check for referenced objects the database can be corrected manually via the button **[Edit]**. The button opens the **Edit** dialog for the respective database object.

When the plausibility check finds no objects whose value ranges lie outside the data type range or whose record layouts do not fit, an appropriate announcement appears.

4.3.6 Optimizations

In this dialog, various optimizations for individual elements of the ECU database can be selected:

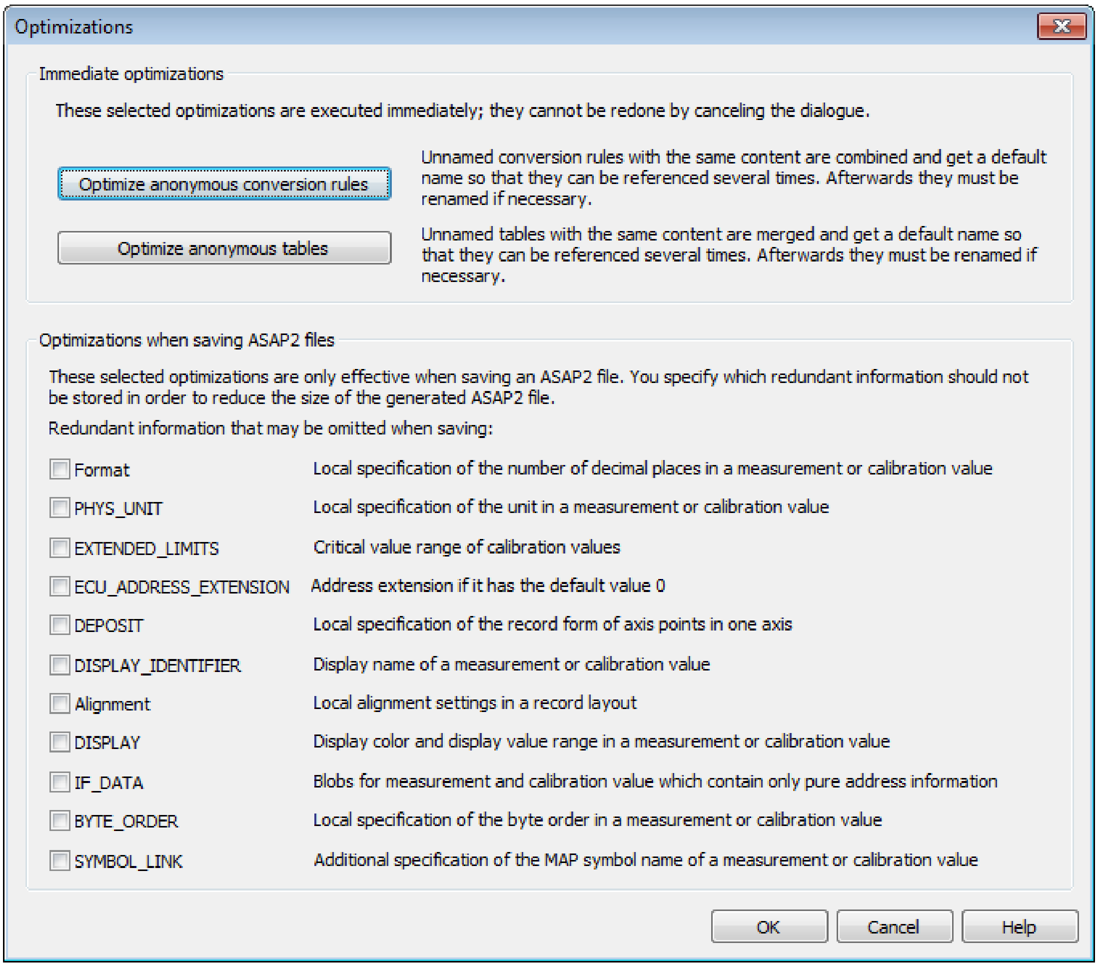

Figure 95: Optimizations

Immediate optimizations: These options will be executed immediately and cannot undone. Unnamed (anonymous) conversion rules or tables are local rules/tables that are directly assigned to a database object. When such a rule/table When such a rule / table appears multiple times with the same content, it is converted into a global conversion rule / table and given the name AnonymousConversionMethodX with X = sequential number. These rules / tables are also shown in the tree view and can be edited.

Optimizations when saving ASAP2 files: These optimizations take place only when the database file is saved. This prevents the program from saving redundant information and the file size is reduced

Apart from the entry EXTENDED_LIMITS these are global settings of the database, which are identical to local settings of the corresponding database object. In case of

EXTENDED_LIMITS the optimization is performed when the values of the critical range, and standard range are identical.

4.3.7 Compare

Selecting the command **Tools | Compare** opens a dialog to select a database for comparison. It is always compared to the currently loaded database. Selecting a database for comparison opens the dialog seen in Figure 96, Seite 98, so that a comparison can be made of the various different elements in the two databases.

For every object category a separated name filter can be given. The wildcards "*" and "?" can be used. The comparison is case sensitive. Default setting for filters is always "*", that means that all objects are compared. In case of variables address differences and differences in the IF_DATA information can be ignored optionally. Two variables that are differ only in their address are seen as identical when this options are active.

IF_DATA differences can be optionally ignored also in case of groups (including function groups) and memory segments.

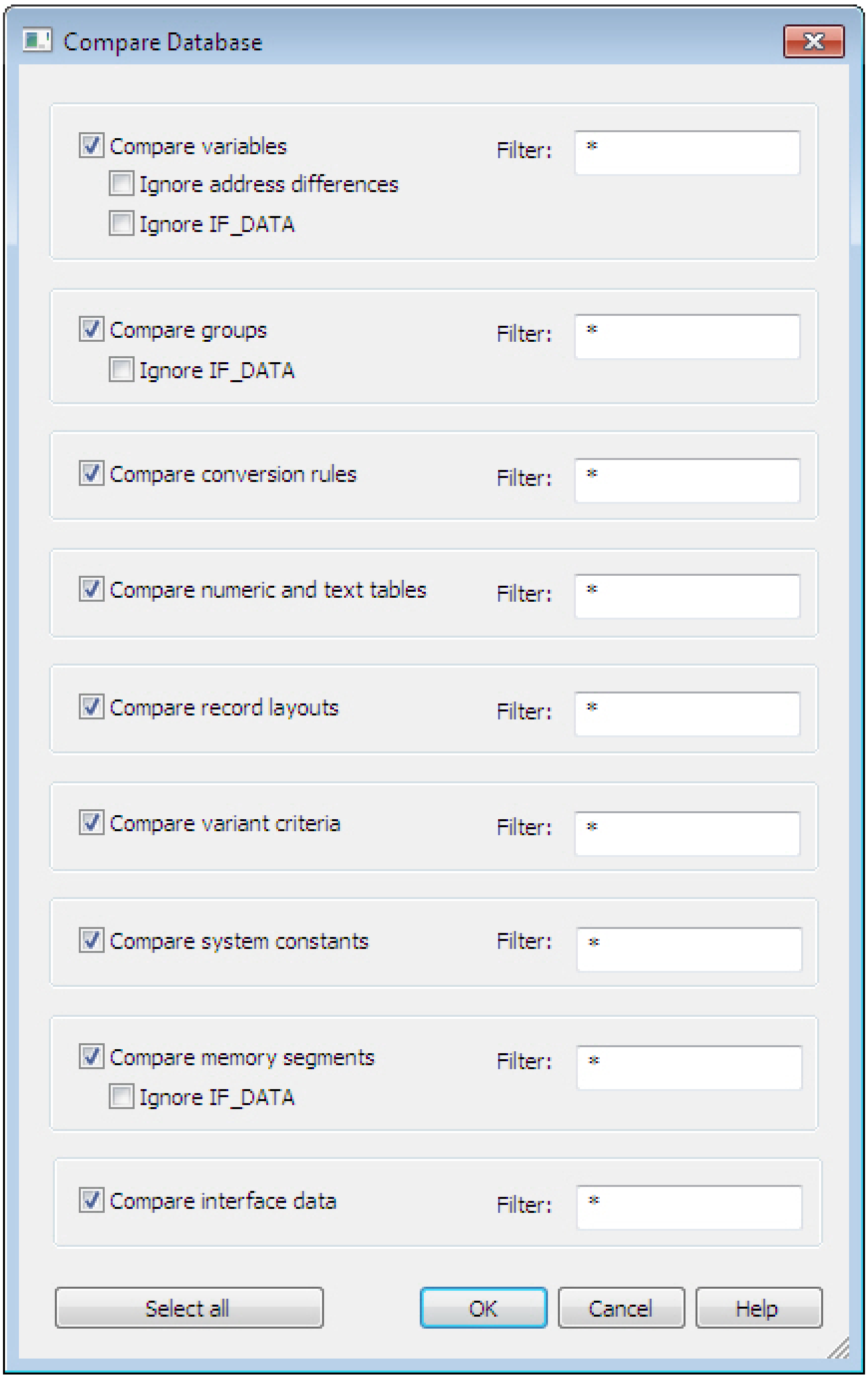

Figure 96: Options for comparing databases

Clicking **[OK]** starts the comparison. Where the databases are different the results of the comparison will be displayed in the Message window of the Database Editor. The text of the message can be copied to the clipboard and to a text editor via the **Copy** command in the popup menu.

5 Index

T

V

Get more Information!

Visit our Website for:

> News

> Products

> Demo Software

> Support

> Training Classes

> Addresses

www.vector-worldwide.com