MOST

Media **O**riented **S**ystems **T**ransport

Multimedia and Control Networking Technology

MOST Editor – User Manual

06/2018

Document Version *3.4.0.12*

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

Legal Notice

COPYRIGHT

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

LICENSE DISCLAIMER

Nothing on any MOST Cooperation Web Site, or in any MOST Cooperation document, shall be construed as conferring any license under any of the MOST Cooperation or its members or any third party's intellectual property rights, whether by estoppel, implication, or otherwise.

CONTENT AND LIABILITY DISCLAIMER

MOST Cooperation or its members shall not be responsible for any errors or omissions contained at any MOST Cooperation Web Site, or in any MOST Cooperation document, and reserves the right to make changes without notice. Accordingly, all MOST Cooperation and third party information is provided "AS IS". In addition, MOST Cooperation or its members are not responsible for the content of any other Web Site linked to any MOST Cooperation Web Site. Links are provided as Internet navigation tools only.

MOST COOPERATION AND ITS MEMBERS DISCLAIM ALL WARRANTIES WITH REGARD TO THE INFORMATION (INCLUDING ANY SOFTWARE) PROVIDED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

In no event shall MOST Cooperation or its members be liable for any damages whatsoever, and in particular MOST Cooperation or its members shall not be liable for special, indirect, consequential, or incidental damages, or damages for lost profits, loss of revenue, or loss of use, arising out of or related to any MOST Cooperation Web Site, any MOST Cooperation document, or the information contained in it, whether such damages arise in contract, negligence, tort, under statute, in equity, at law or otherwise.

FEEDBACK INFORMATION

Any information provided to MOST Cooperation in connection with any MOST Cooperation Web Site, or any MOST Cooperation document, shall be provided by the submitter and received by MOST Cooperation on a non-confidential basis. MOST Cooperation shall be free to use such information on an unrestricted basis.

TRADEMARKS

MOST Cooperation and its members prohibit the unauthorized use of any of their trademarks. MOST Cooperation specifically prohibits the use of the MOST Cooperation LOGO unless the use is approved by the Steering Committee of MOST Cooperation.

SUPPORT AND FURTHER INFORMATION

For more information on the MOST technology, please contact:

MOST Cooperation
Administration
Emmy-Noether-Str. 14
D-76131 Karlsruhe
Germany

Tel: (+49) (0) 721 966 50 00

E-mail: contact@mostcooperation.com
Web: www.mostcooperation.com

MOST Editor – User Manual



MOST Cooperation

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

Contents

1 INTRODUCTION 6

2 GETTING STARTED 7

2.1 System Requirements 7
2.2 Installation....7
2.3 MOST Editor – A quick walk-through....11
2.3.1 Startscreen 11
2.3.2 Components 12
2.3.3 Create your first function catalog 13

3 WORKING WITH THE MOST EDITOR 20

3.1 Using elements in the MOST Editor 20
3.1.1 Create elements 20
3.1.2 Edit elements 22
3.1.3 Delete elements....23
3.1.4 Single shallow copy of an element....23
3.1.5 Multiple copies of an element 24
3.1.6 Deep copy of an element....25
3.1.7 Compare elements 26
3.1.8 Unmerging elements....28
3.2 Editing a function catalog....28
3.2.1 Open an existing function catalog....28
3.2.2 Create a new function catalog 30
3.2.3 Importing from existing function catalogs 31
3.2.4 FCat MOST Import Errors and Warnings 31
3.2.5 FCat MOST Export Errors and Warnings....34
3.2.6 FIBEX Export Errors and Warnings....37
3.2.7 FIBEX Import Errors and Warnings....38
3.2.8 MOST Editor Window Title 38
3.3 Working with MOST FBlocks....39
3.4 Working with functions....41
3.5 Working with parameters....43
3.5.1 Stream parameter....44
3.5.2 Number parameter 46
3.6 Working with signals....49
3.7 Saving a function catalog....49
3.8 Saving as new function cataloge 49
3.9 Exporting....50
3.10 Generating html output 52
3.11 Analyzing a function catalog 53
3.11.1 Quality....53
3.11.2 Identifier 55
3.11.3 References 56
3.11.4 Identical objects 57
3.12 Toggle between tree/table element viewing modes....58
3.13 Comparing two FBlocks that are stored in different function catalogs....58
3.14 Command line converter....59

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

4 PRINCIPLES AND CONCEPTS GUIDING THE MOST EDITOR 61

4.1 Top-down approach 61
4.2 Infix-Filter 61
4.3 Reuse of existing elements of a function catalog 61
4.4 Predefined elements 62
4.4.1 Units 62
4.4.2 OPTypes 62
4.5 References 63

5 HOW-TO / TIPS AND TRICKS / FAQ 64

5.1 How can I incorporate the MOST Editor in the context menu of my explorer? 64
5.2 Can I use shortcuts? 66
5.3 How can I quickly access previously edited elements? 68

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

1 Introduction

The MOST Editor is a state-of-the art software for editing MOST function catalogs. As ease of use was one of the main goals in the development of the MOST Editor the structure and handling uses mechanisms, icons, menus etc. common in today's editors to make basic functionality self-explanatory. The following XML formats for saving and loading of function catalogs are supported: FIBEX, FCat DTD and FCat XSD.

This manual describes how the MOST Editor works. It is not an introduction to the MOST Specification nor does it explain the meaning of MOST elements.

This manual refers to version 3.4.0.12 of the MOST Editor distributed by the MOST Cooperation.

Please contact mosteditor@mostcooperation.com for any remarks or requests concerning the MOST Editor.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

2 Getting started

2.1 System Requirements

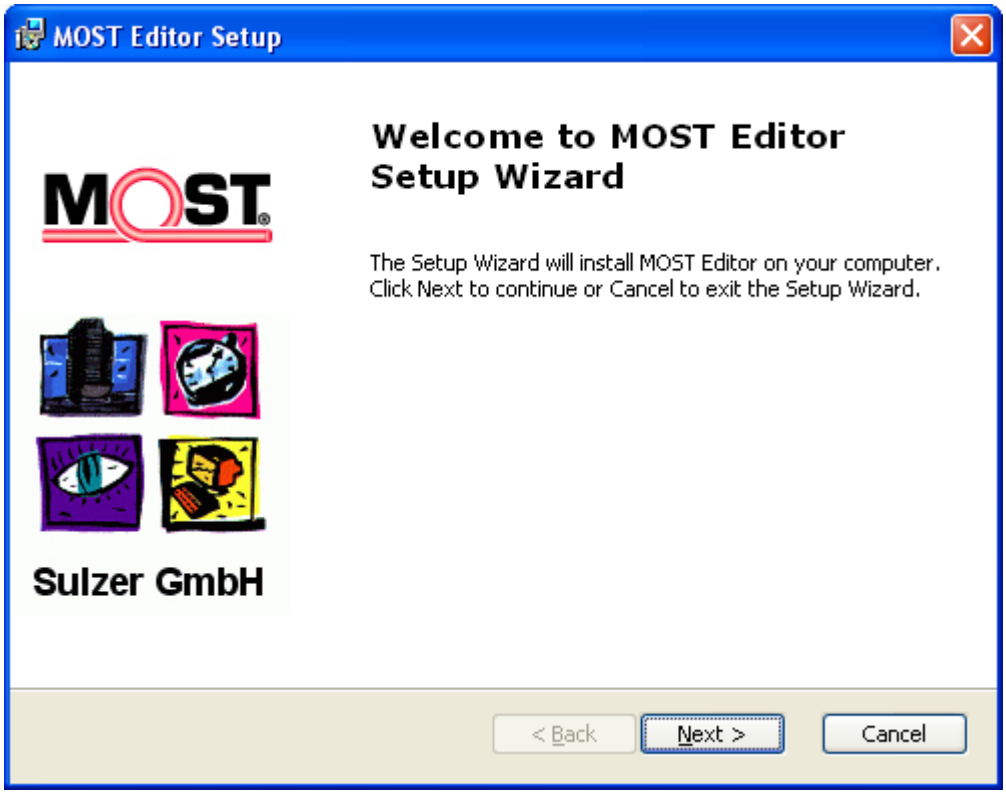
To use the MOST Editor you need a Windows System with Windows 2000 or higher.

2.2 Installation

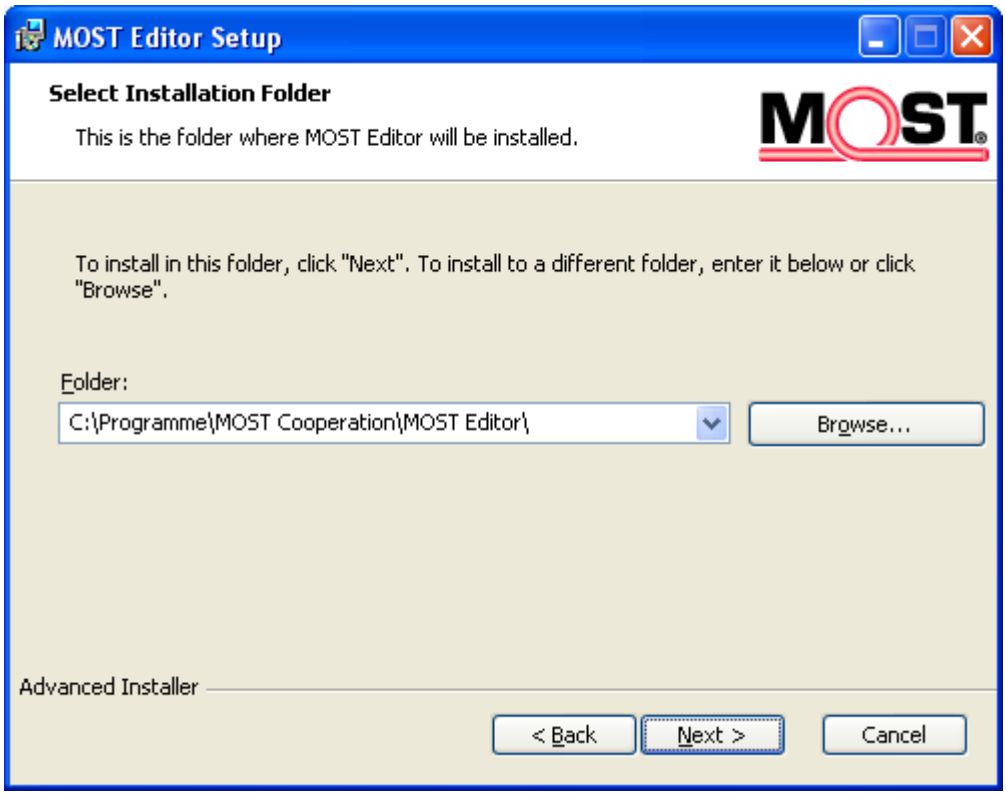
You start the installation by double-clicking the MOST Editor install file

```
MOSTEditor-<Version>.msi.
```

A standard windows installer starts. The only two things you have to do during installation are to select the location where you want to install the MOST Editor and to accept the license agreement.

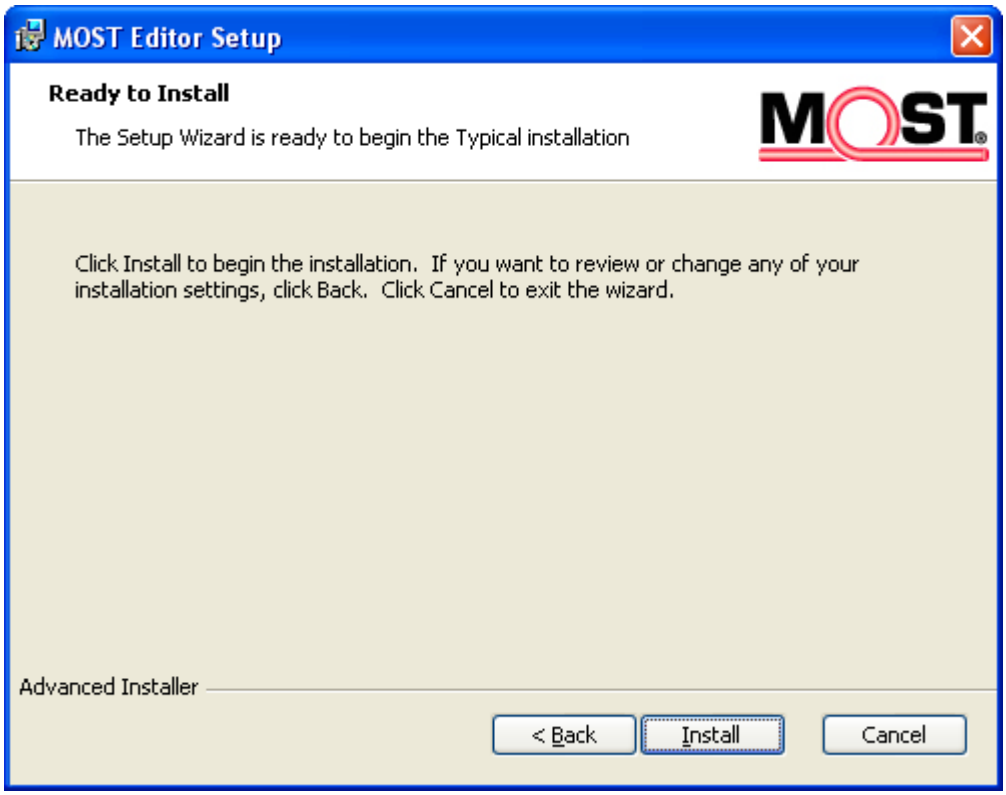


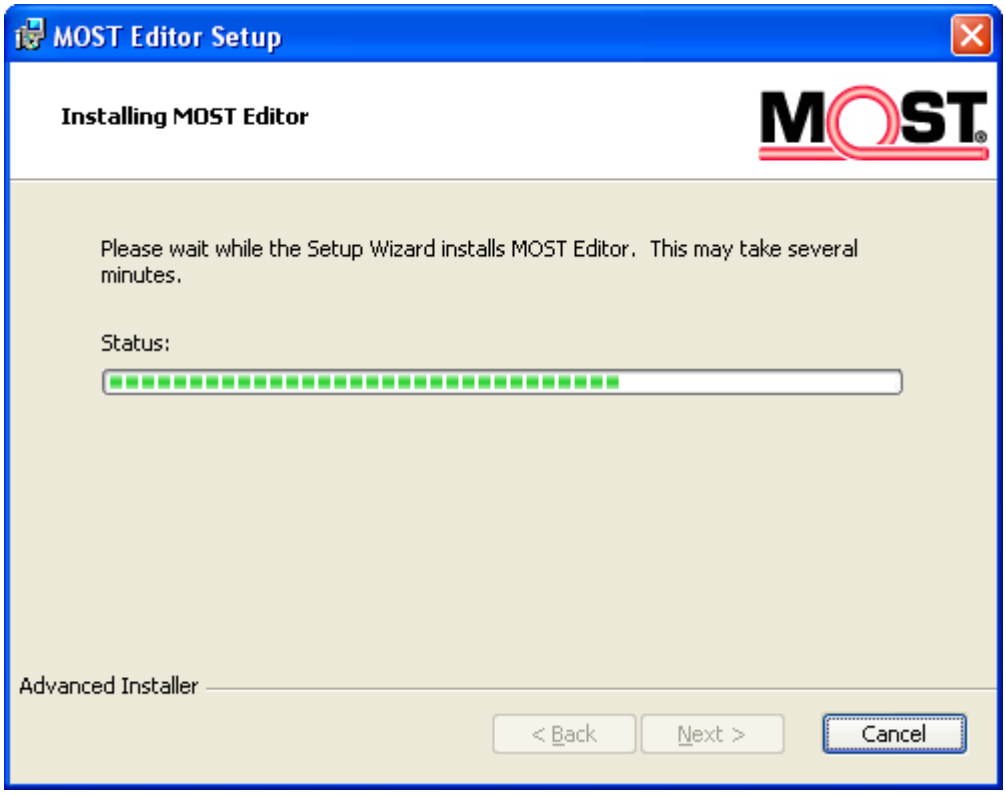
© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.



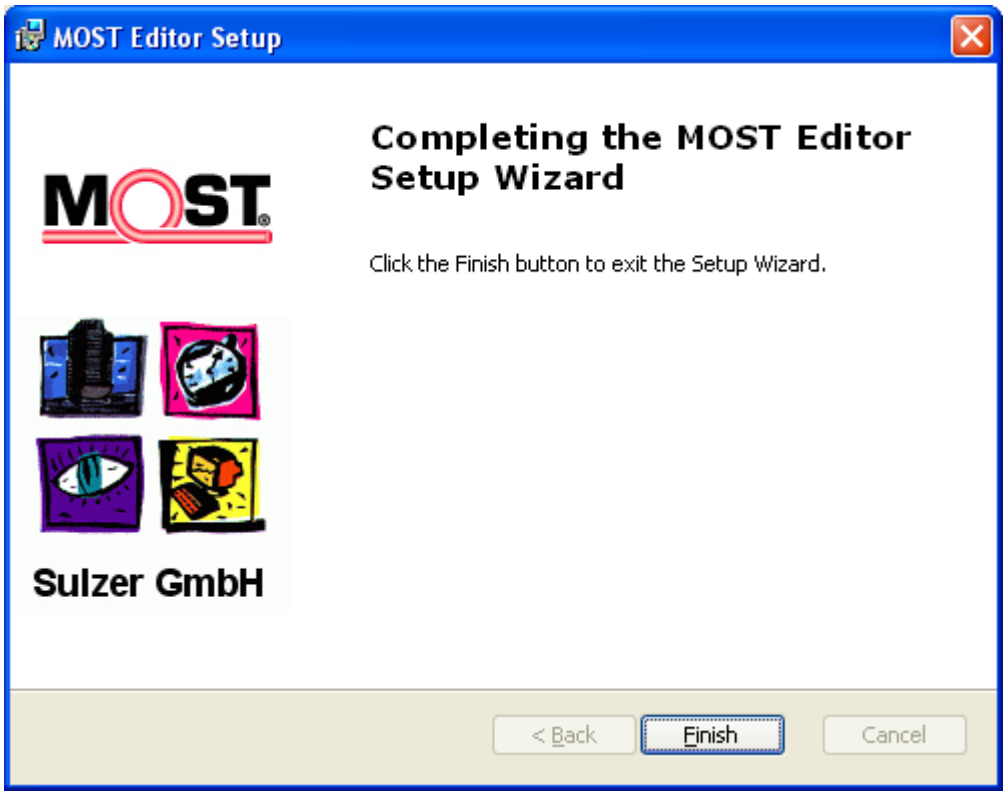


© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.





© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.



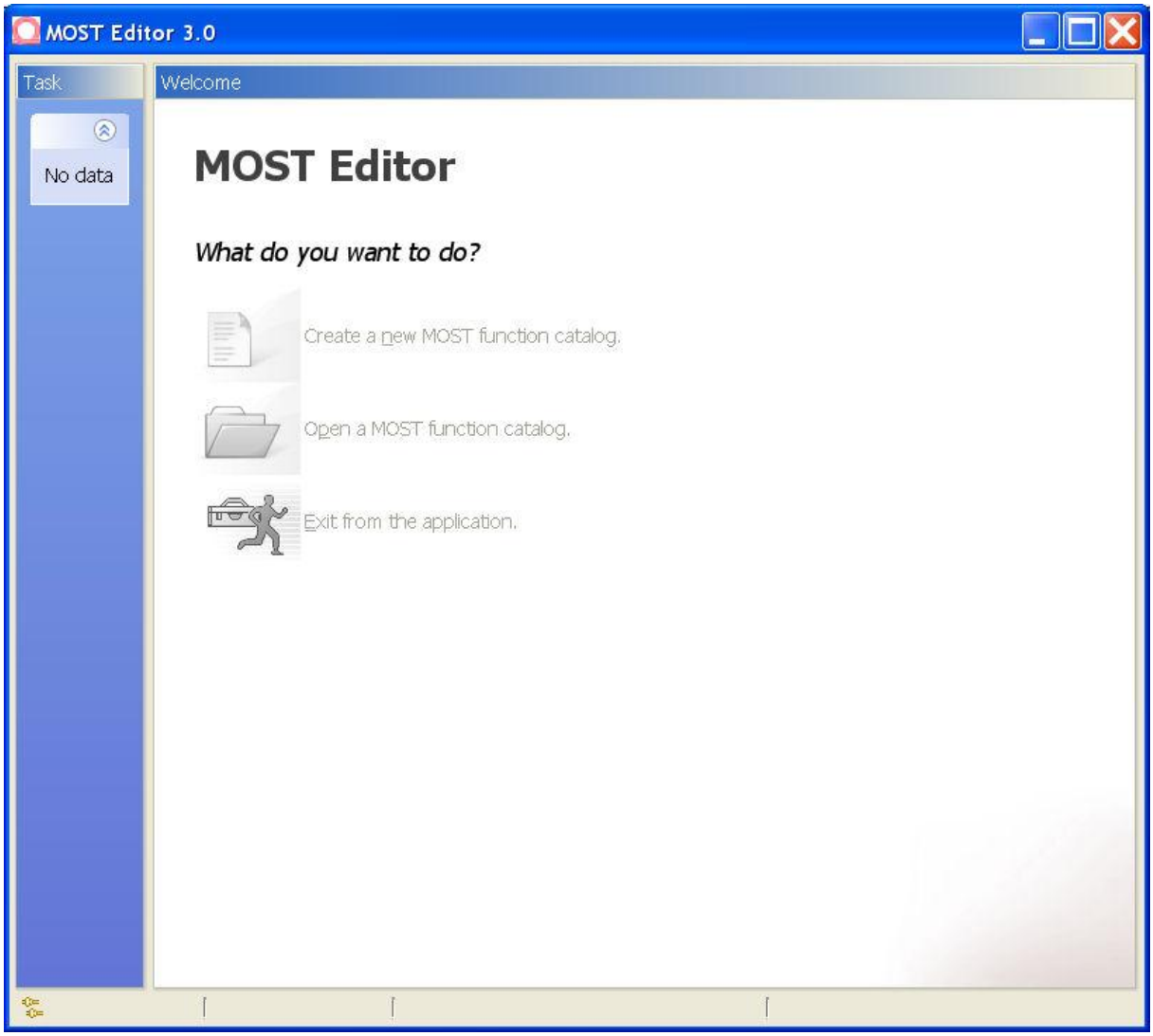
© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

2.3 MOST Editor – A quick walk-through

After installing you can start the MOST Editor either from the program menu of your computer via Start → Programs → MOST Editor → MOST Editor or by double-clicking on the short-cut named “MOST Editor” on your desktop.

2.3.1 Startscreen

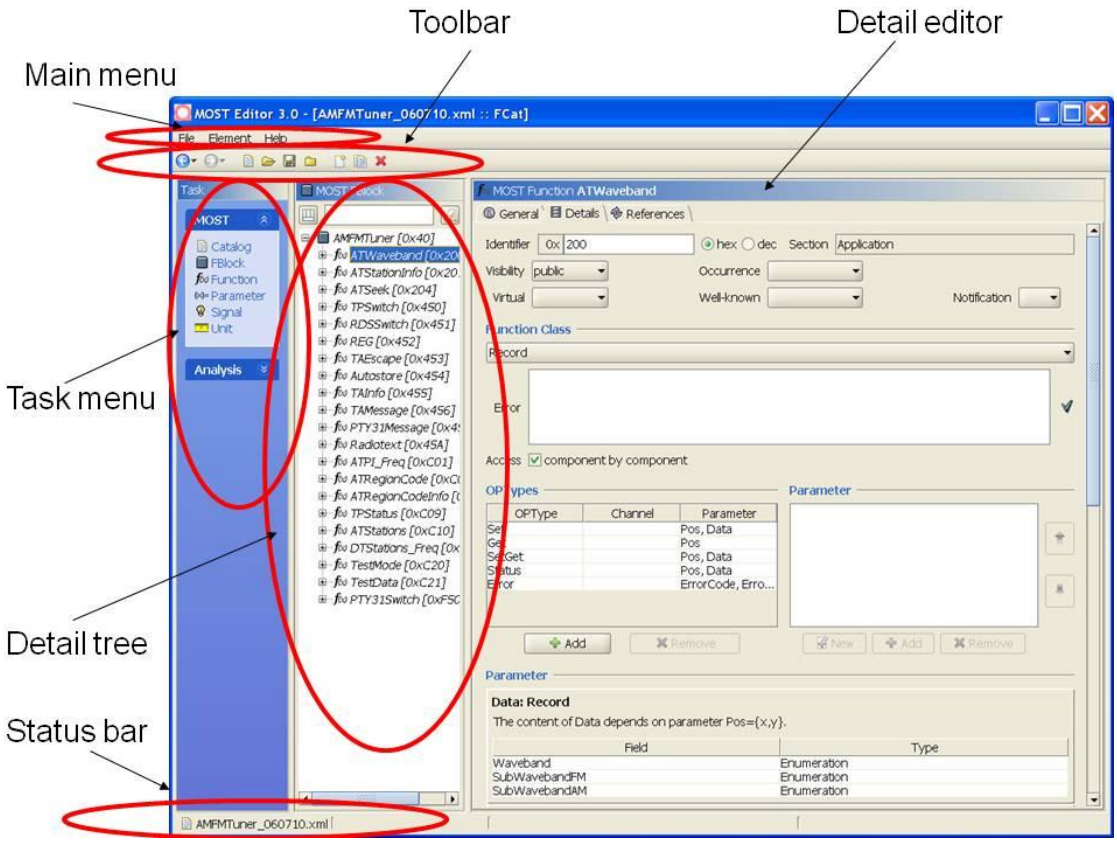
The welcome page is the first page you see when you launch the MOST Editor. You can choose from a clearly arranged menu what do you want to do.



© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

2.3.2 Components

The main window of the MOST editor consists of several different areas.

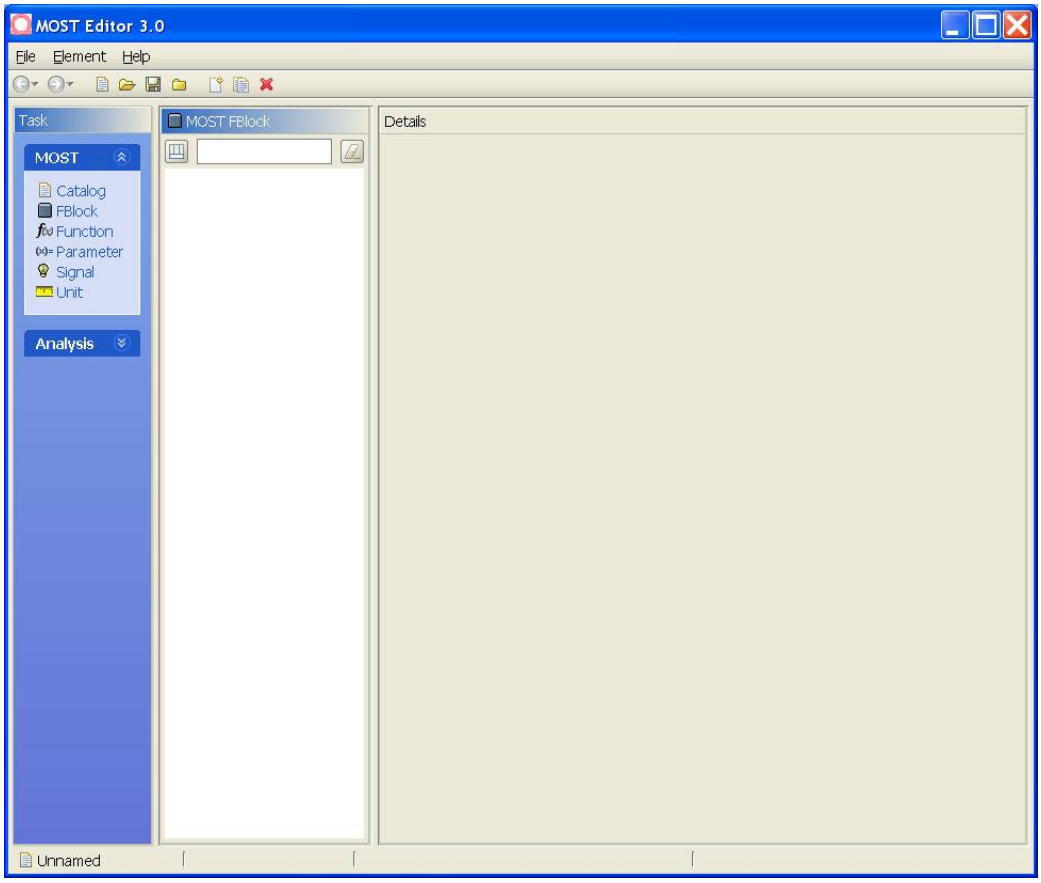


Component	Purpose
Main menu	Using the main menu you can execute all file handling actions and basic operations on elements.
Toolbar	The toolbar is a collection of icons that allows you to directly access often-used commands. If you move your mouse over an icon a tool-tip appears.
Status bar	The status bar shows the name of the file that is currently opened in the MOST Editor.
Task menu	The task menu gives a structured overview of the elements in a MOST file.
Detail tree	The detail tree shows the elements connected to the item that is selected in the task menu. You can navigate through this structure (click on “+” to expand and on “-“ to collapse the substructures). A context menu enables a quick access to the elements in the detail tree by just right clicking on it.
Detail editor	The detail editor consists of one or more tabs showing details of the element that is selected in the detail tree. Which tabs are displayed depends on the type of the selected element.

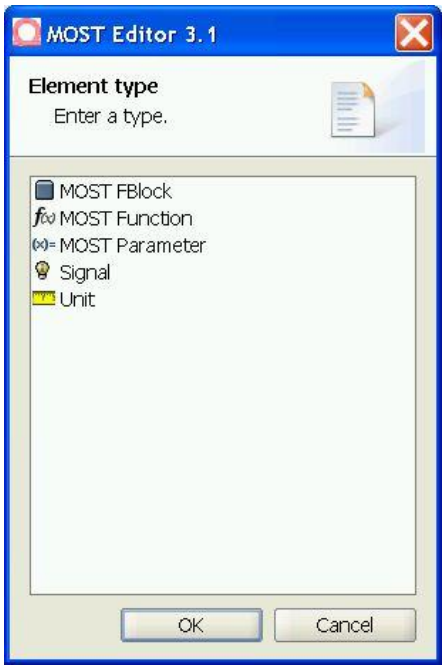
© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

2.3.3 Create your first function catalog

When you have chosen to create a new MOST function catalog from the start screen the following window appears:

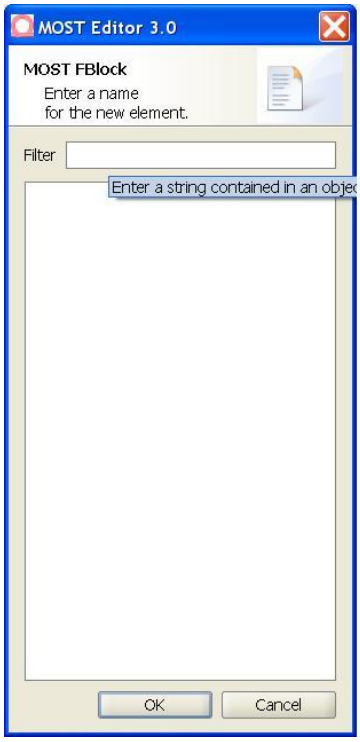


In the main menu choose “Element” → “New”. The following pop-up window appears:

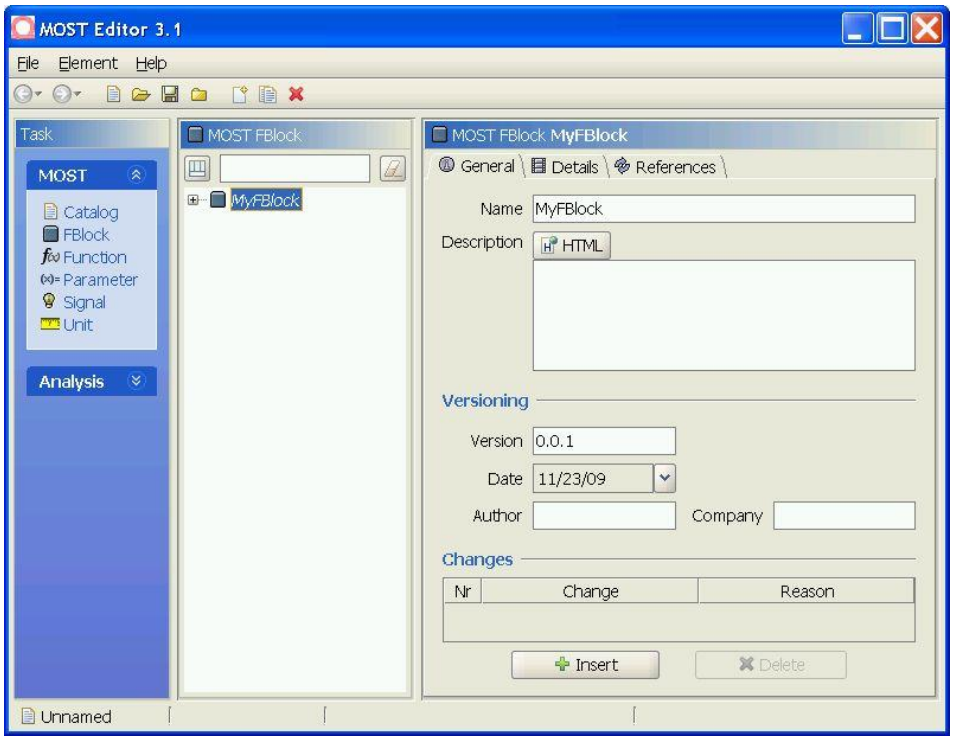


© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

Select “MOST FBlock” and click “OK”.

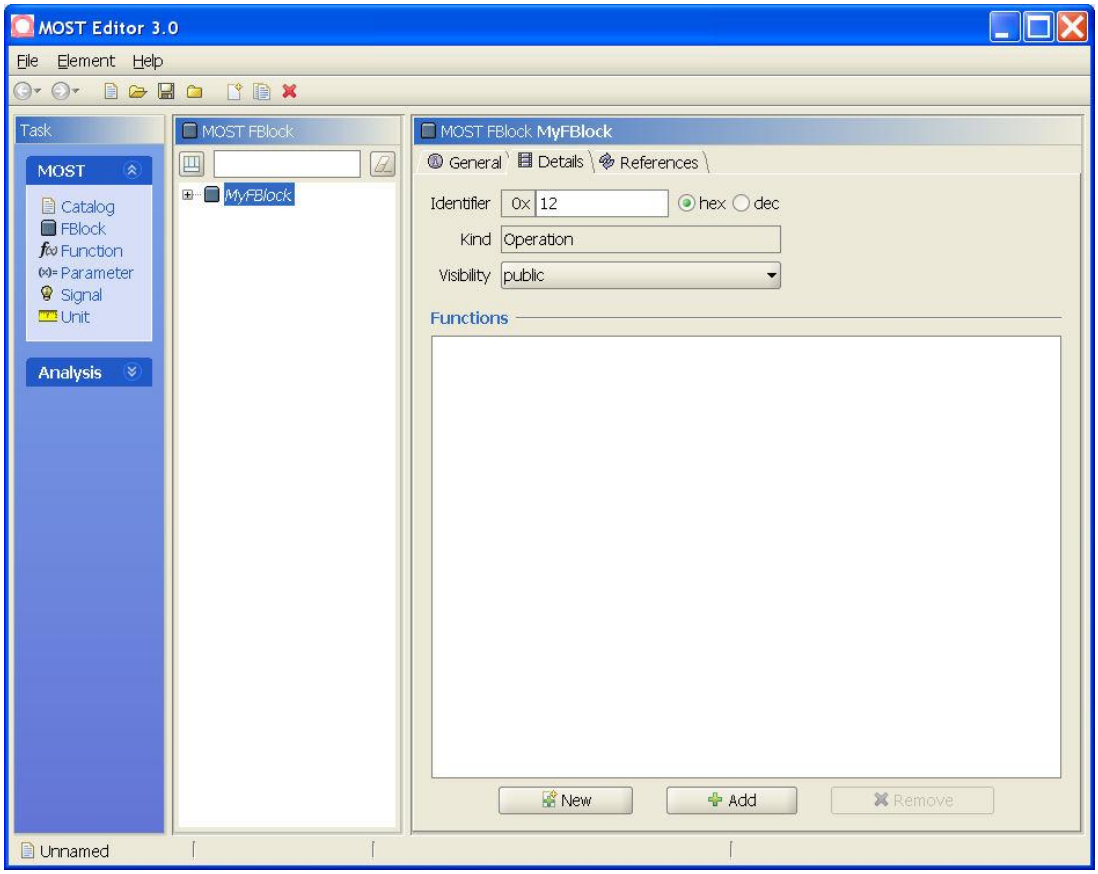


Enter the name of your FBlock and click “OK”. An FBlock is created and the appropriate detail editor opens.



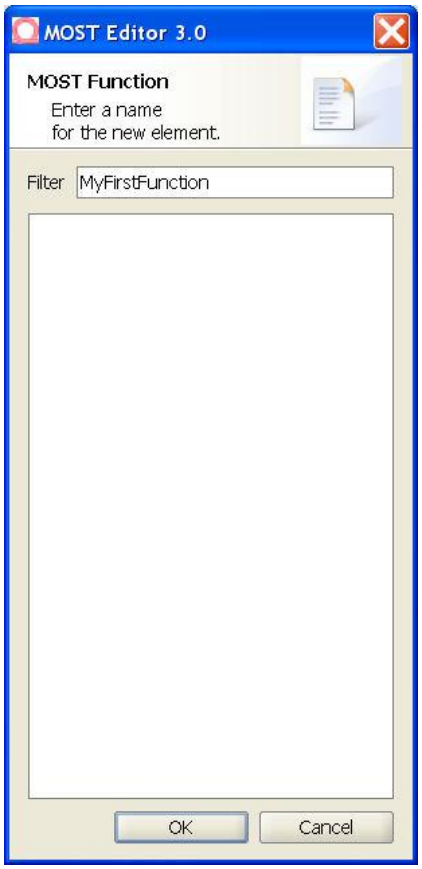
In the tab called “General” of the detail editor you can enter a description for your FBlock and add versioning information.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

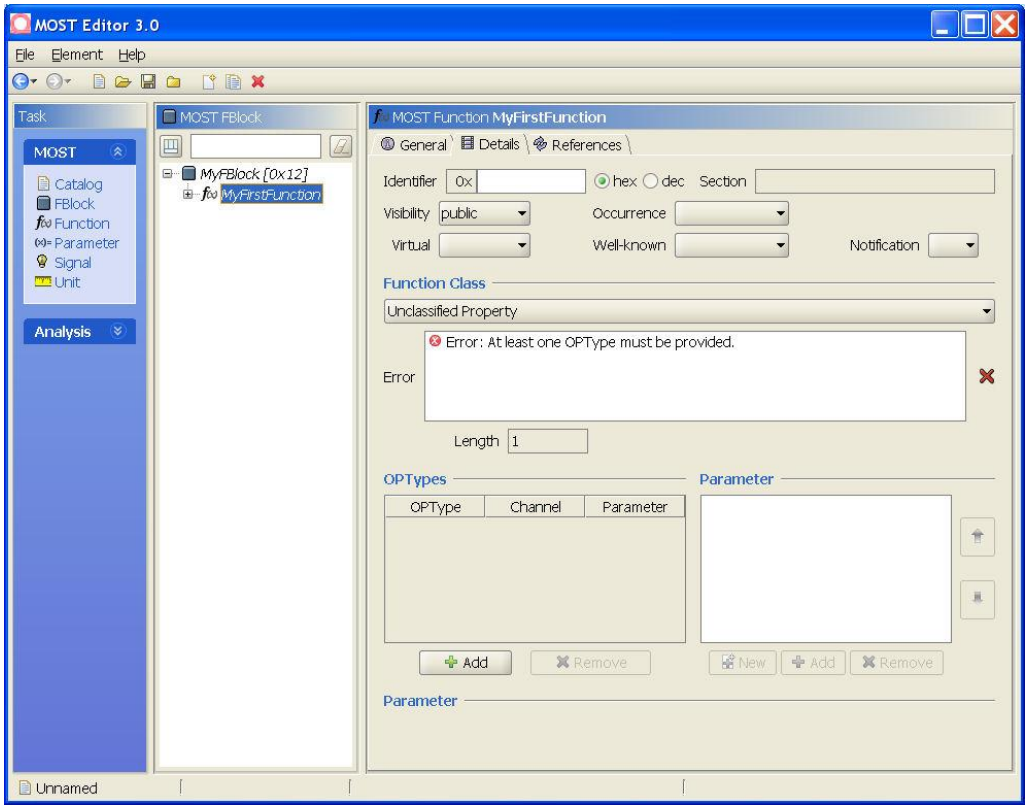


In the tab named “Detail” of the detail editor enter an identifier.
At the bottom at “Functions” choose “New” to create a new function. The following pop-up window appears:

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

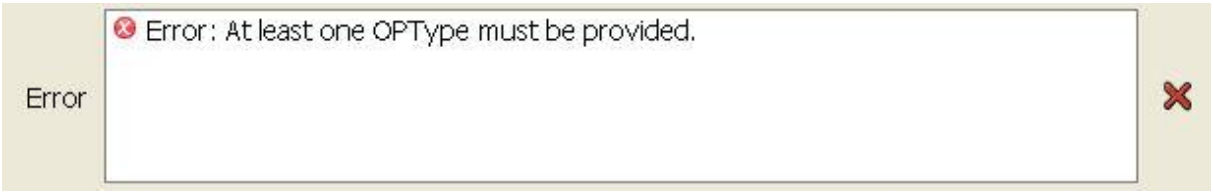


Enter a name for your function and click “OK”.

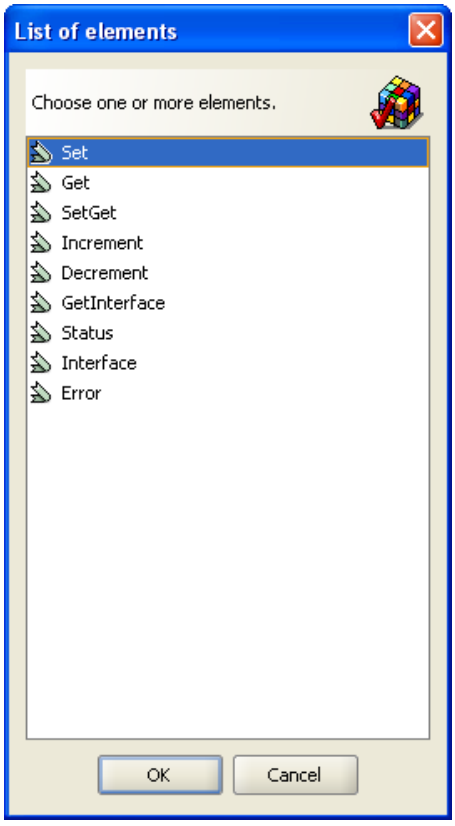


© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

The function is created and attached to the FBlock. The detail editor for the function opens. As you have not yet entered detail information about your function it is not in a consistent state. The “Error” section tells you what you have to do to make your function consistent according to the MOST Cooperation’s specification of the xml file that contains a function catalog. This is important as you can only save consistent elements in your function catalog.

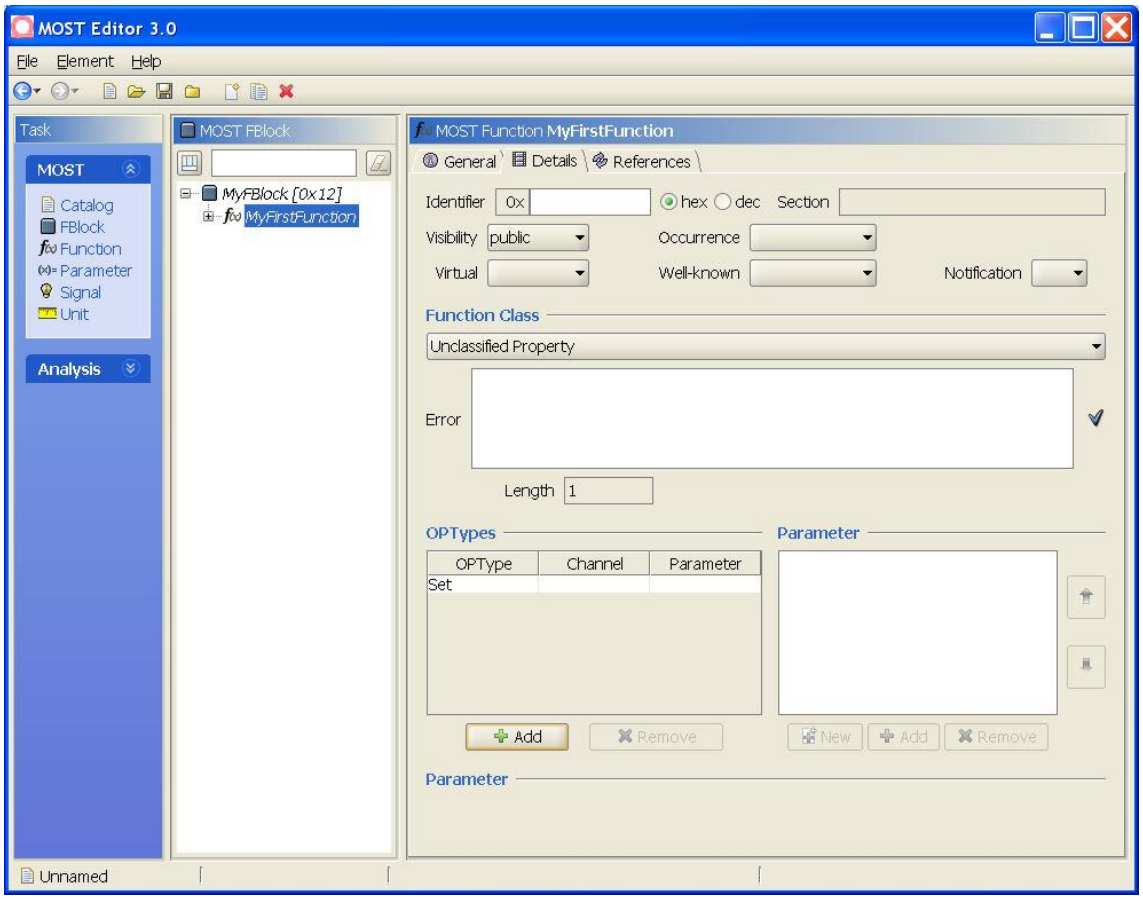


For your newly created function this means that you have to add at least one OP-Type. To do this click “Add” at the OPType-section in the “Details” tab of the detail editor of your function. A pop-up window opens where you can select an OPType for your function.

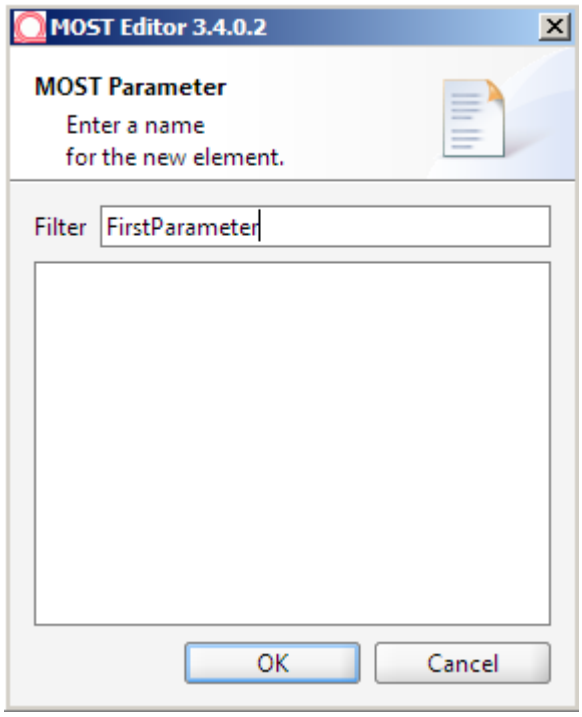


Choose for example “Set” and click “OK”. You can select multiple elements if you want to add more than one OPType.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

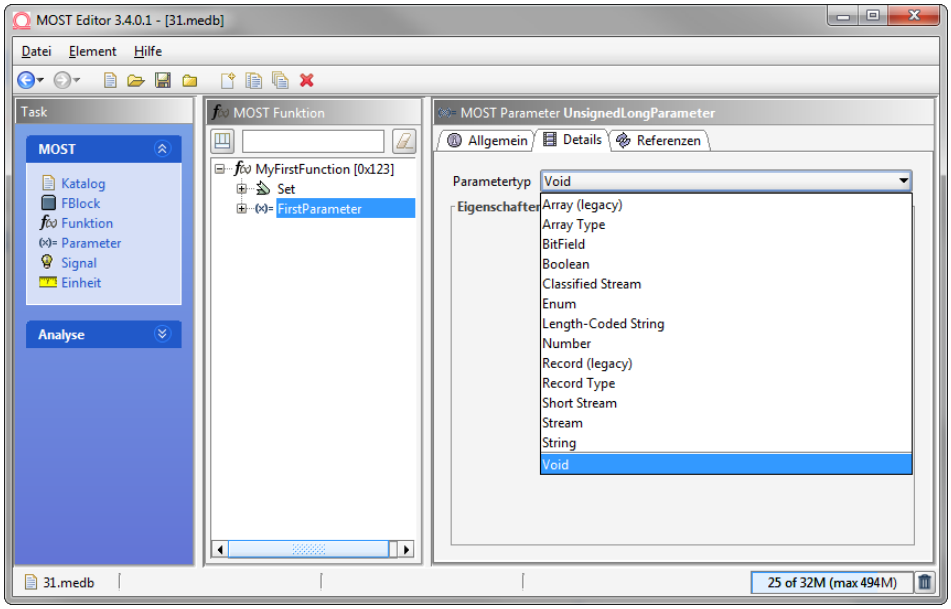


An OPType “Set” is created. Please note that the “Error” section is empty now. Click on the OPType “Set”. The “Parameter” area is activated. Click “New” to create a parameter.



Enter the name of your parameter and click “OK”.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.



The detail window for the newly created parameter opens. You can change the parameter type here. Depending on the parameter type you have to enter additional information about your parameter. Now select in the main menu “File” → “Save”. Enter a filename and click “OK”.

You have created your first function catalog using the MOST Editor.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

3 Working with the MOST Editor

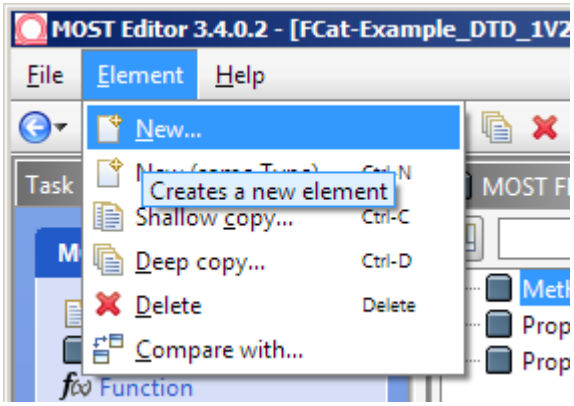
3.1 Using elements in the MOST Editor

FBlock, Function, Parameter, Signal, and Unit are different elements the MOST Editor works with. The basic mechanism for creation, editing, deletion, copy and comparison is the same for all these elements.

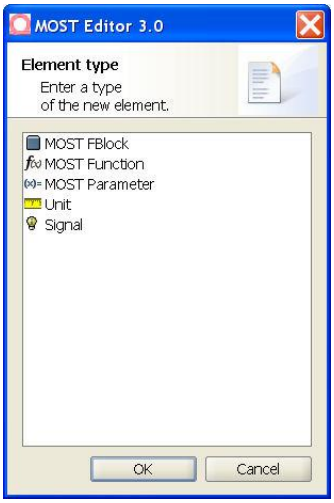
3.1.1 Create elements

There are several ways to create new elements. Which one you choose depends on your personal preferences.

1. New element
   Choose in the main menu “Element” → “New”

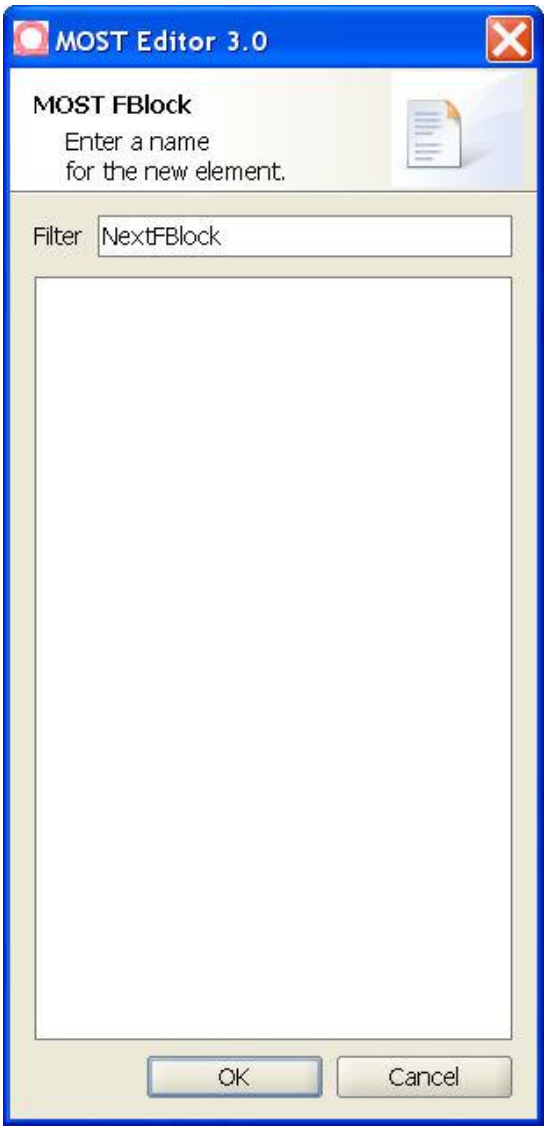


A new window opens where you can choose which type of the element you want to create.

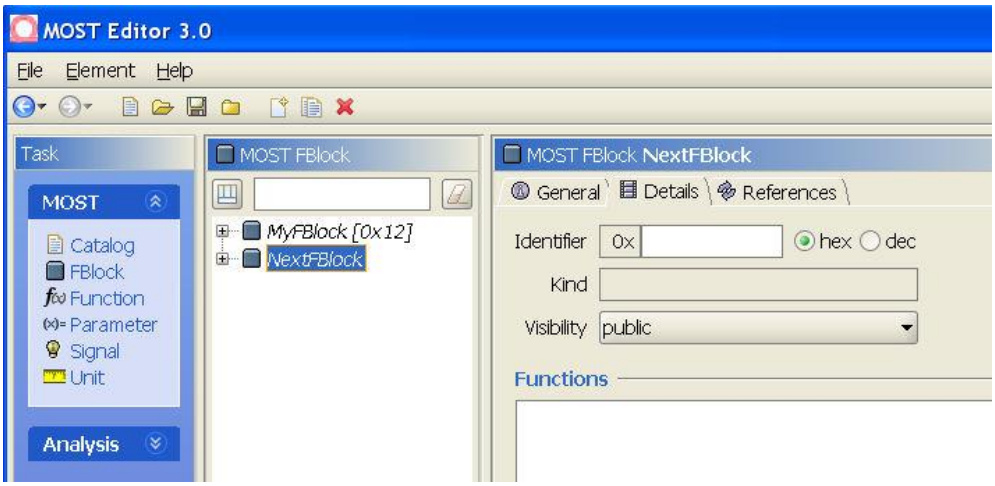


Choose for example “MOST FBlock” and click “OK”. Another pop-up window opens where you can enter the name of the element you want to create.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.



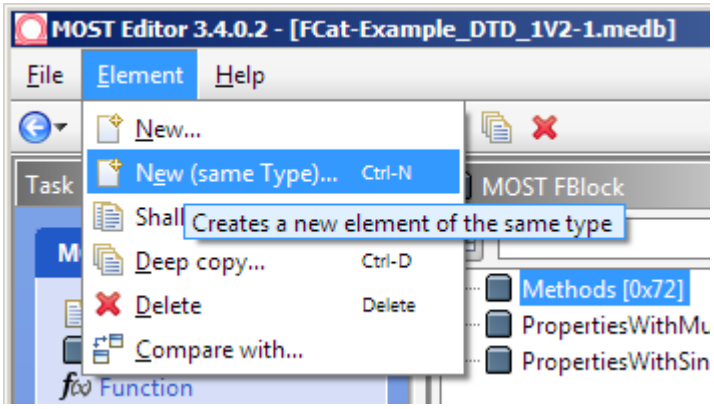
Enter a name and confirm by clicking “OK”.



The element is created and opened in the detail tree and in the detail editor for you to complete it.

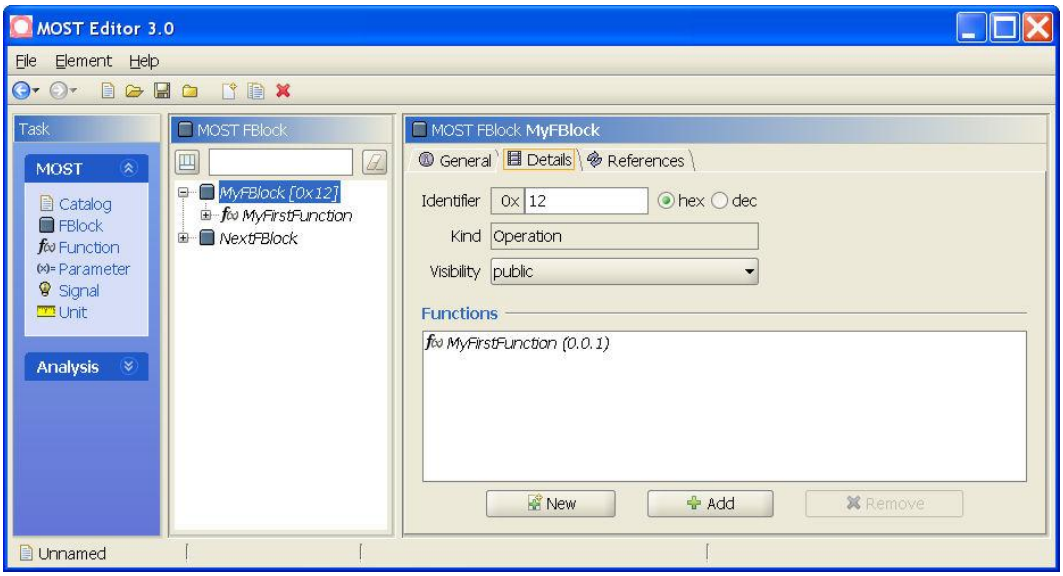
© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

2. New element (same)
   In the task menu mark the type of the element you want to create. For example select FBlock. Choose “Element” → “New (same Type)” from the main menu or press  in the toolbar.



   A pop-up window opens where you can enter the name of your element. The element gets created and the appropriate editor opens.

3. Create an element in the detail editor of a superior element.
   For example you can create a function in the detail window of an FBlock.



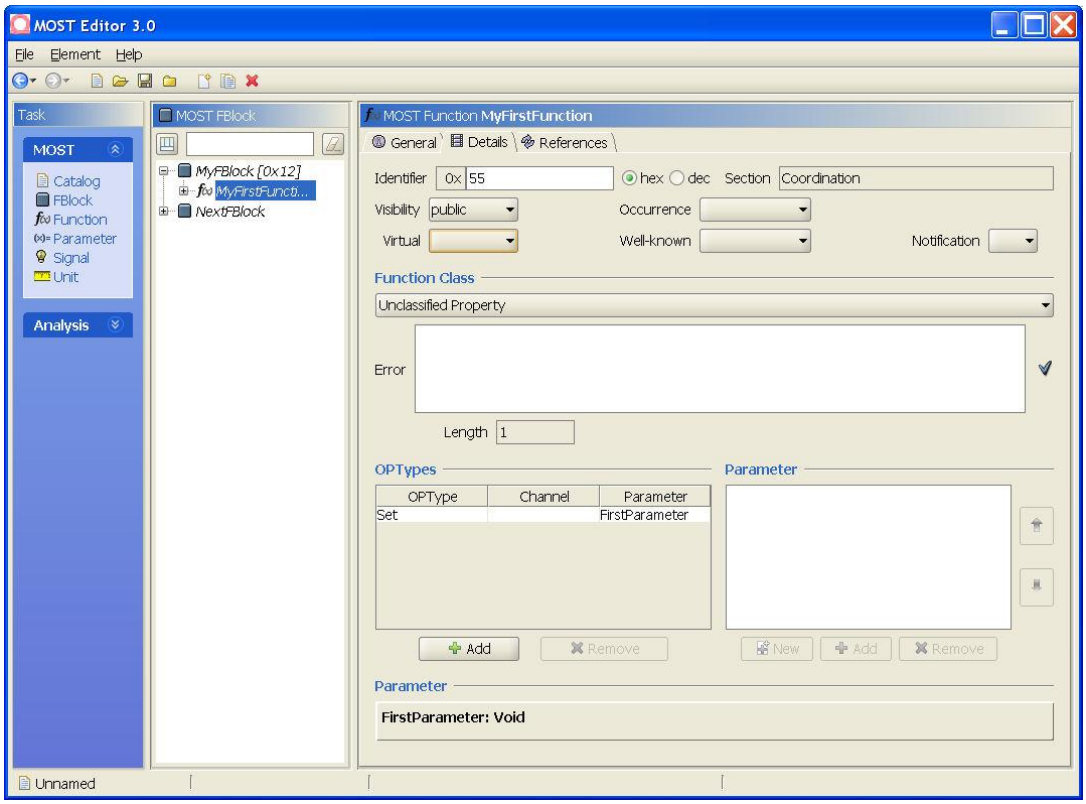
   Pressing “New” at the bottom of the “Details” tab of an FBlock opens the dialog to create a function.

After you created an element you have to enter the appropriate data in the detail editor.

3.1.2 Edit elements

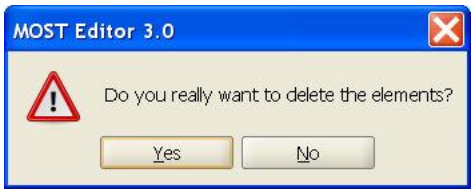
Select the element you want to edit in the detail tree. The appropriate editor opens in the detail editor window. The editor window contains different tabs that display information about the selected element and allows you to edit the data of the element.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

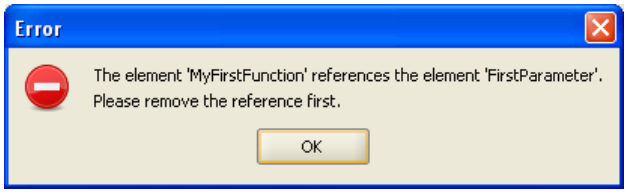


3.1.3 Delete elements

Select the element(s) you want to delete in the detail tree. Then choose “Element” → “Delete” from the main menu or  from the toolbar or right-click on one of the selected elements in the detail tree. The MOST Editor asks you to confirm the deletion of the selected element(s).



If the elements you marked for deletion are referenced by other elements you cannot delete them. You have to delete the references or the referencing elements first.

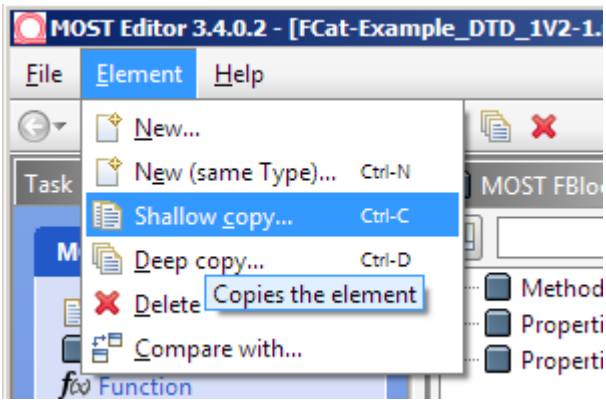


3.1.4 Single shallow copy of an element

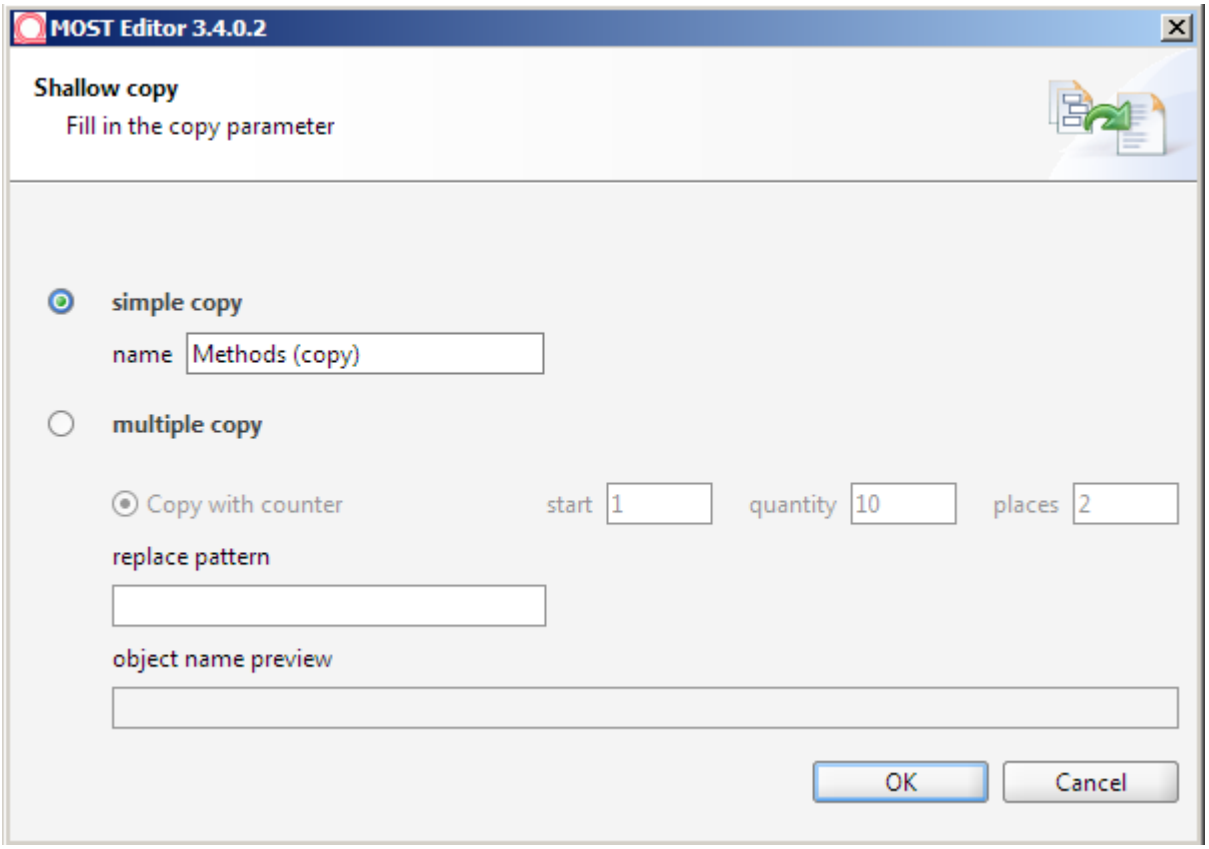
In the detail tree mark the element you want to copy. Please note that you can only copy one element at a time. If you mark more than one element only the element you marked last will be copied. Then choose “Element” → “Shallow copy…” from the

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

main menu or click  in the toolbar or right-click on the element you want to copy in the detail tree.



A window opens where you can customize your copy.



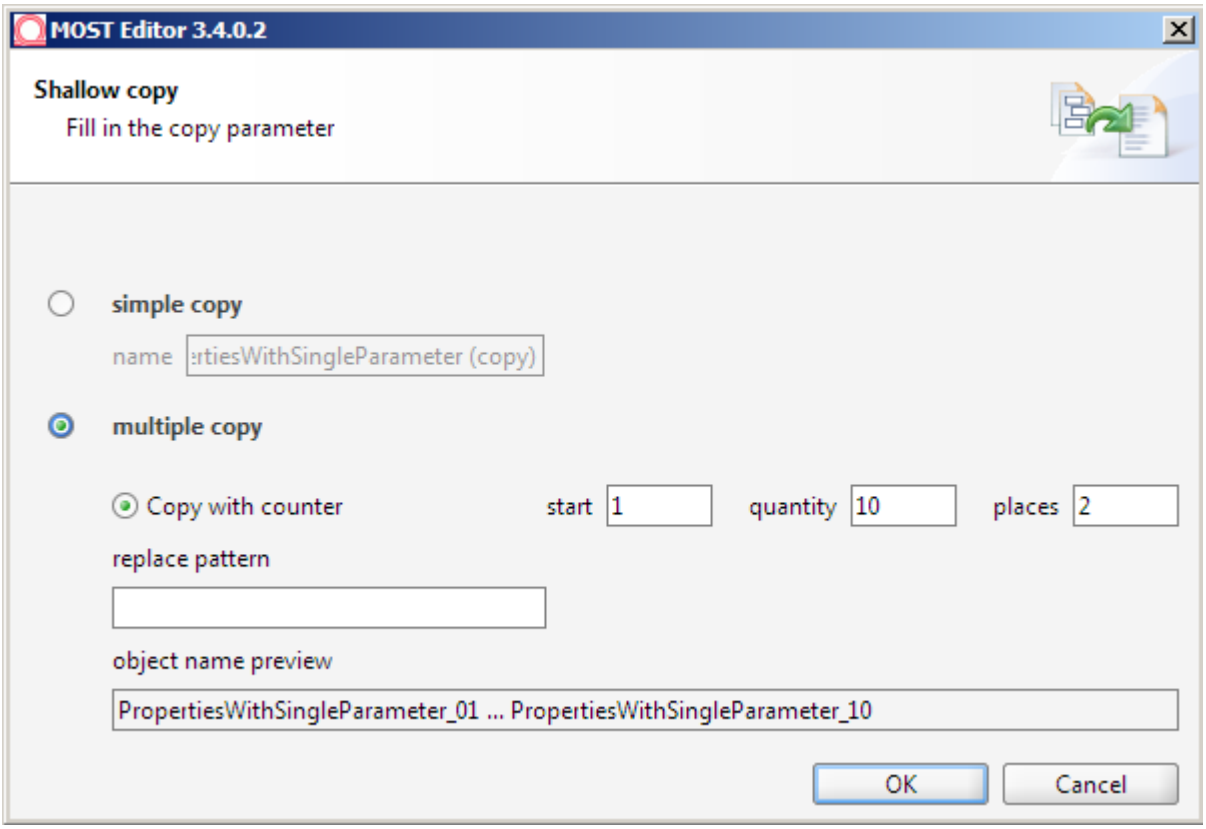
Enter a name for your copy and select “OK”. A copy of your element with the name you specified is created.

3.1.5 Multiple copies of an element

You can create multiple copies of an element.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

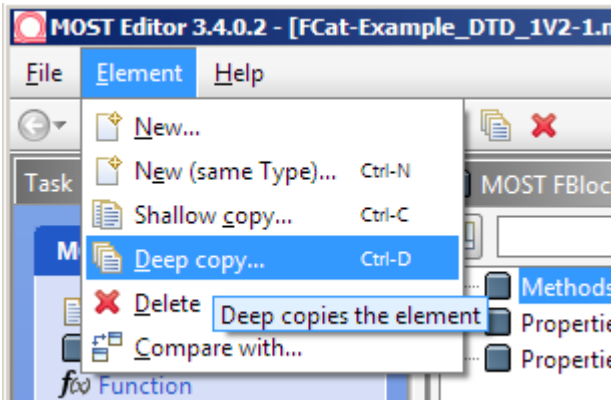
Choosing “File” → “Copy” in the main menu or  in the toolbar or right-click on the element you want to copy in the detail tree opens the following dialog:



You can customize the names of your copies by specifying a prefix and a suffix for the name of the new elements.

3.1.6 Deep copy of an element

To create a deep copy of an element including all references choose “Element” → “Deep copy…” from the main menu or right-click on the element you want to copy in the detail tree.



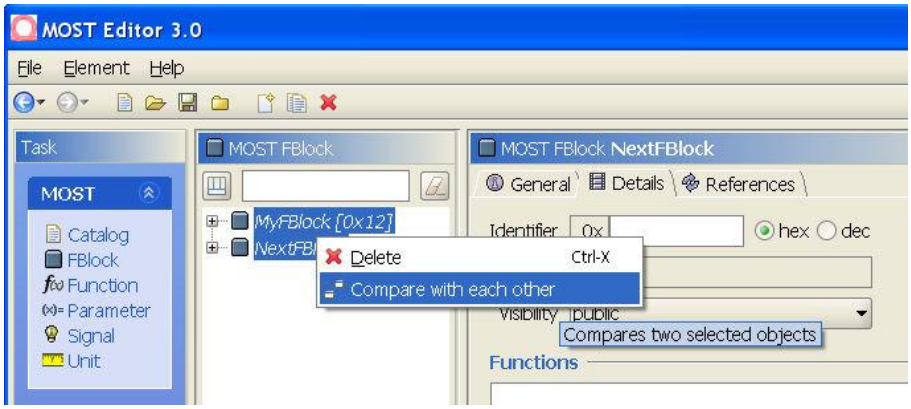
© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

The copy rules and the dialog to customize your copy is same as described in 3.1.4

3.1.7 Compare elements

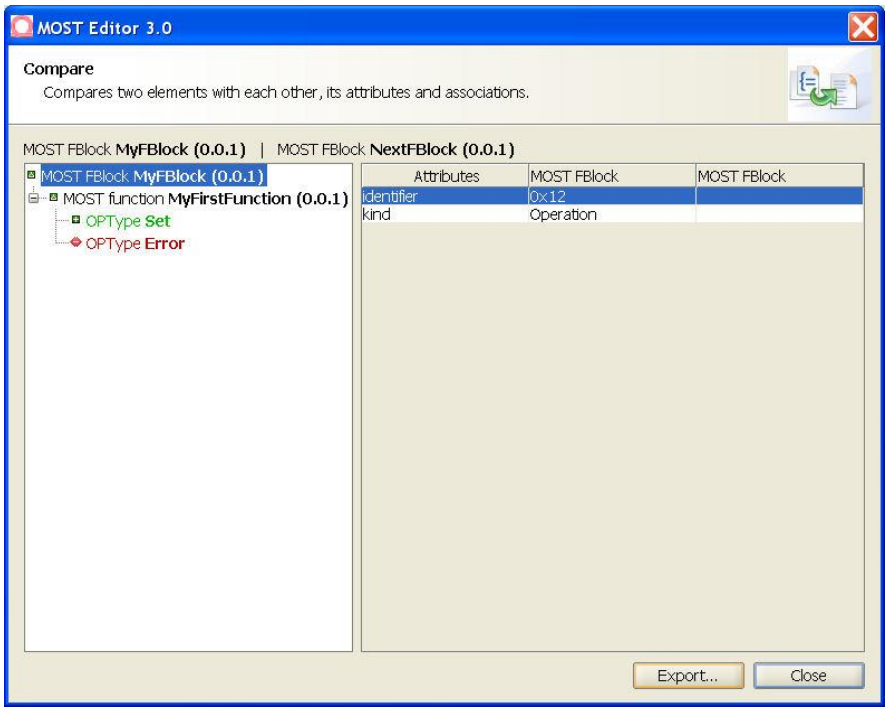
To compare two elements you can select both in the detail tree and then use “Compare with each other” from the context menu. Alternatively, you can select one element, then use “Element” → “Compare with” from the main menu. A pop-up window opens where you can select the second element for the comparison.

Please note that the compare function from the context menu and the compare function from main menu slightly differ: In the context menu you select two elements that will be compared with each other. If you use the compare functionality in the main menu you select one element in the detail tree. Then you use the dialog “Element” → “Compare with” to find the second element for the comparison. If you have selected two elements in the detail tree the comparison “Element” → “Compare with” from the main menu does not compare the two selected elements but compares the last selected element with the element you choose from the dialog opened by “Element” → “Compare with”.

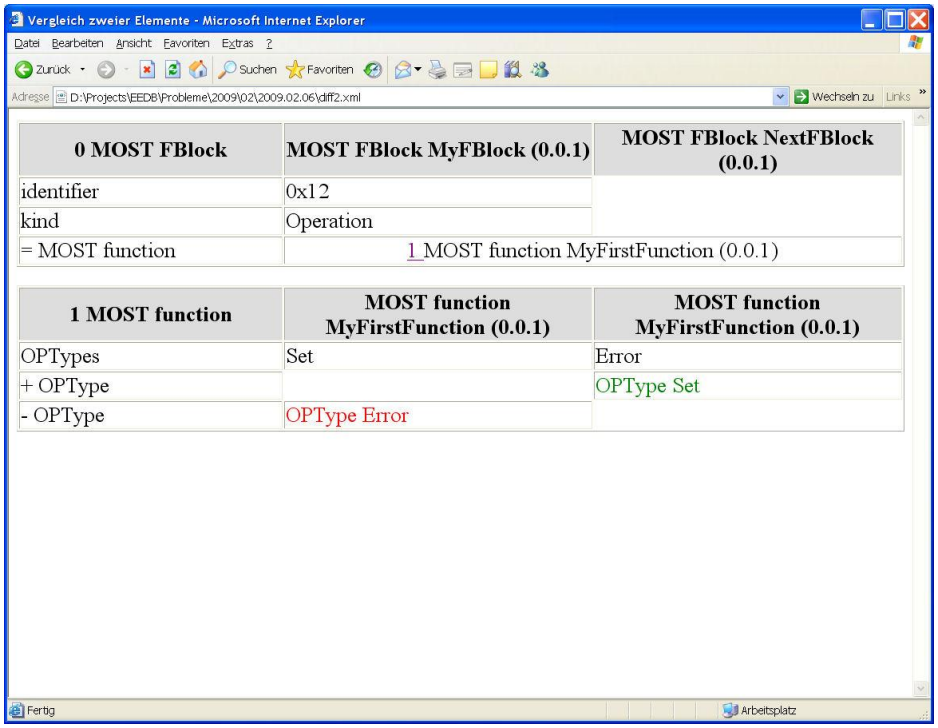


The result of the comparison is displayed in a new pop-up window. The two FBlocks differ in the OPType Error in the function MyFirstFunction:

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

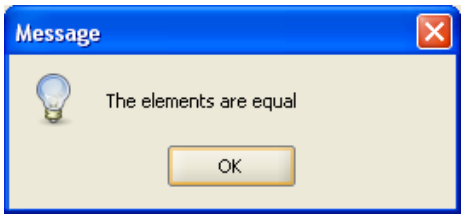


The differences between the two compared elements can be exported into an XML file that can be viewed in a Web browser.



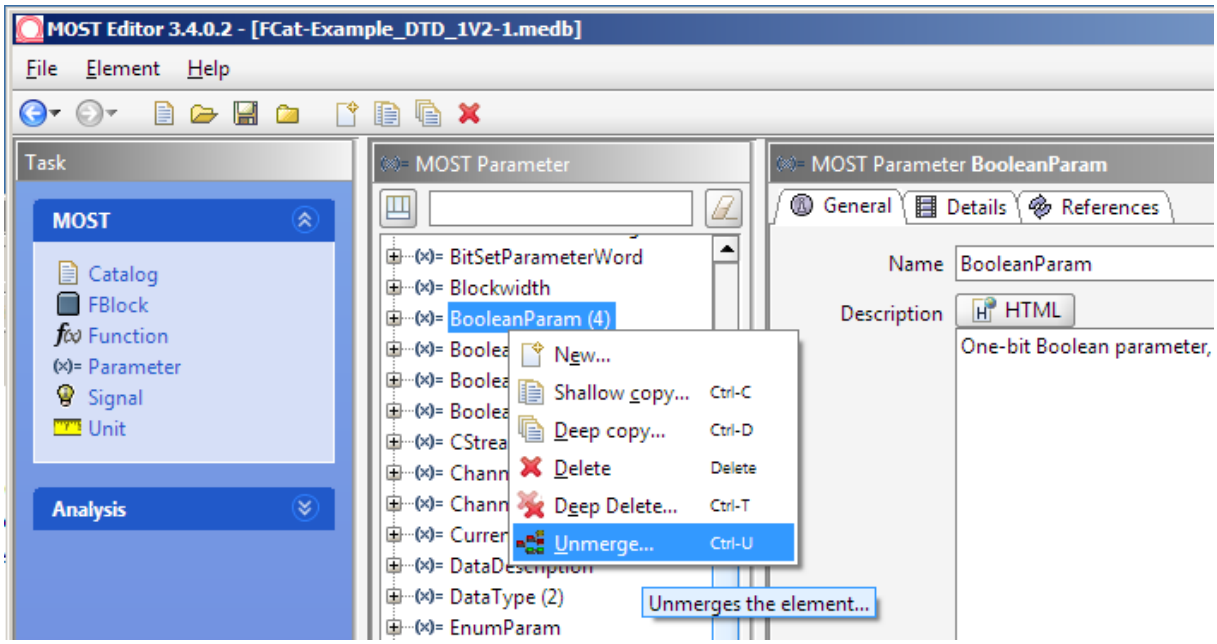
After removing the OPType Error the comparison results in the following message:

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.



3.1.8 Unmerging elements

MOST-Functions and Parameters referenced by multiple elements may be unmerged. The numer of references is displayed in perentheses next to the name for those elements. Let's assume "Function A" is used in two FBlocks "FBlock A" and "FBlock B". Unmerging this Function will create a new Function "Function A*" so that "FBlock A" uses "Function A" and "FBlock B" uses "Functions A*" afterwards. To unmerge an element select it in the tree and choose "Unmerge…" from the context menu:



.

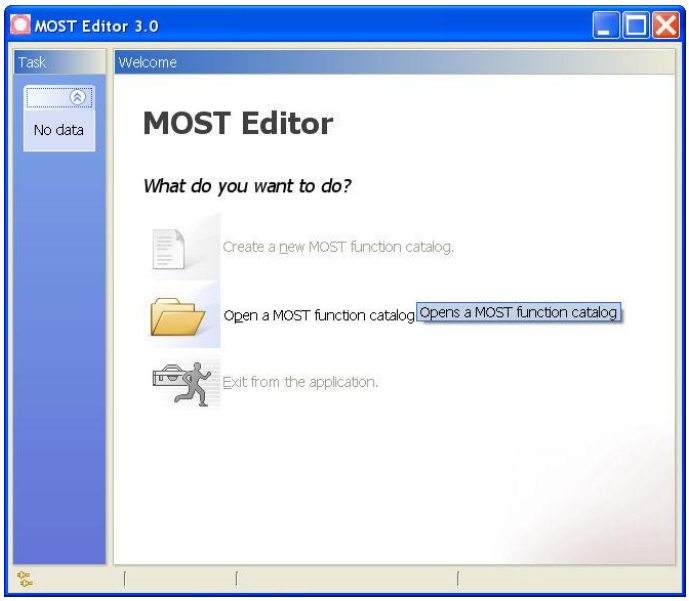
3.2 Editing a function catalog

A MOST function catalog is stored either in a file-based database (*.medb) or in an xml file that follows a structure defined by the MOST Cooperation. The MOST Editor allows you to comfortably work with a function catalog.

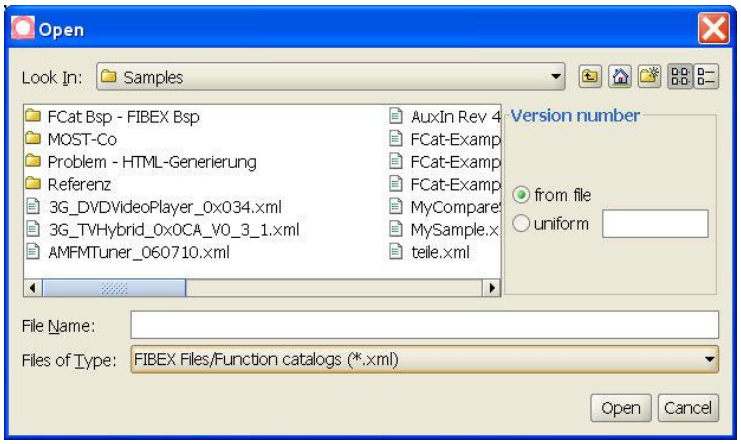
3.2.1 Open an existing function catalog

When you start your MOST Editor the welcome screen appears where you can choose to open an existing function catalog.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.



If you are already working with the MOST Editor and you want to open a new file you can return to the welcome screen by choosing “File” → “Close”. Alternatively, you can open a function catalog by choosing “File” → “Open” in the main menu, the open-function-catalog icon in the toolbar or by drag&drop from the Windows explorer. If you choose “Open” you can select a MOST xml file from your file system. In this case closing the current function catalog and opening the new function catalog will be done in one step.

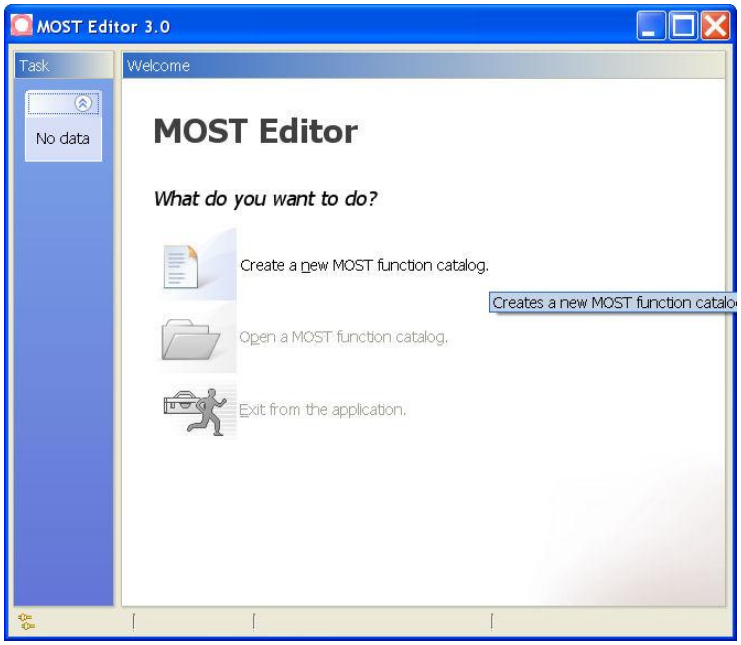


A MOST function catalog can be loaded from an MEDB file or from an XML file in either FCat DTD format or in FIBEX format. Depending on the contents of the file the MOST Editor can automatically decide in which format the file was saved and loads it accordingly. In the dialog one can provide a uniform version number that will be assigned to all imported elements. Otherwise, the version numbers read from the file will be assigned to the imported elements.

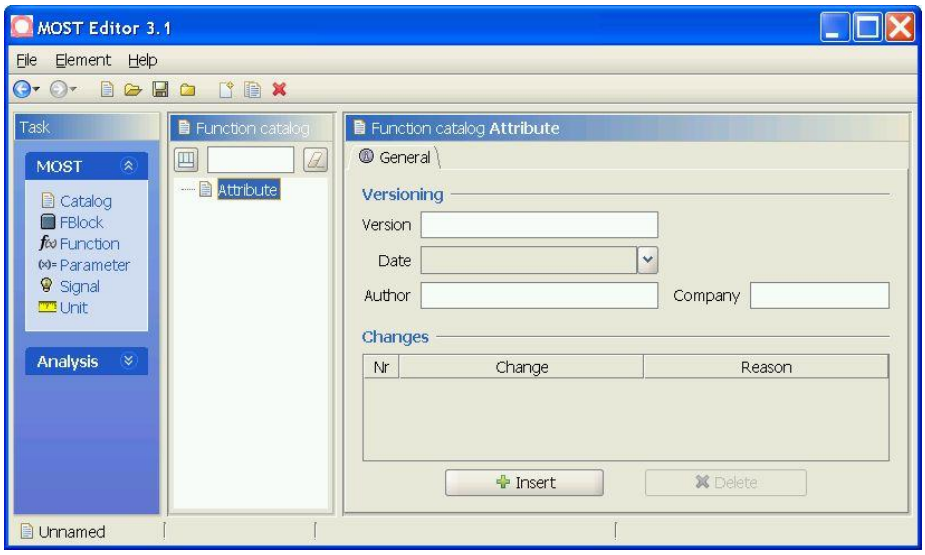
© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

3.2.2 Create a new function catalog

When you have just started your MOST Editor you can choose “Create a new MOST function catalog.” from the welcome screen.



The MOST Editor can handle only one function catalog at a time. If you have already opened a function catalog in your MOST Editor you have to close it in order to create a new function catalog.
In the main menu choose “File” → “New” to close your current function catalog and create a new one or use the create-function-catalog icon from the toolbar.

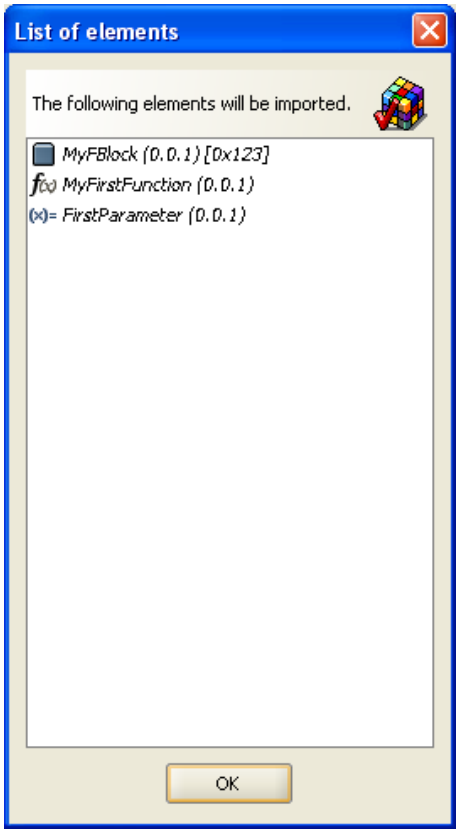


Selecting “Catalog” in the task menu and “Attribute” in the detail tree allows you to enter general information like author and version for the function catalog. This information can be used to distinguish different versions of a function catalog and to identify the person in charge of the maintenance of the function catalog.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

3.2.3 Importing from existing function catalogs

If you want to reuse elements from an existing function catalog in the function catalog you are working with you can import these elements. In the main menu select "File" → "Import" and choose the file that contains the function catalog you want to import.

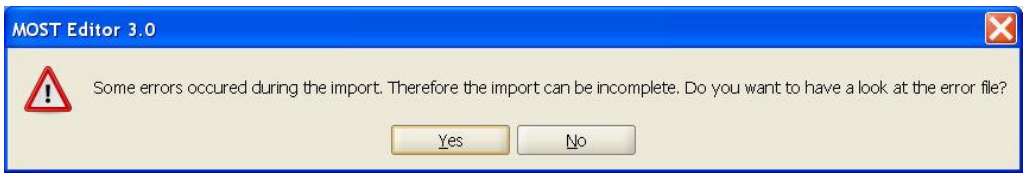


A pop-up window shows you which elements will be imported. Click "OK" to close the window.

3.2.4 FCat MOST Import Errors and Warnings

The file that is to be imported must be a valid XML file. It must satisfy the constraints imposed by both the DTD and the MOST specification.
An XML file can be completely imported only if it satisfies all the constraints. Otherwise the import may be incomplete/wrong.

If the file does not satisfy a constraint, the following message in a yes/no dialog will be shown:



© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

By choosing "yes" a list of error and warning messages is shown in a text editor. By choosing "no" the messages are not shown.

The list of error and warning messages is saved in the same directory from where the imported file was loaded and with the same name but with the extension ".txt". You can have a look at the messages at any time.

The constraint violations are separated by their severity into two groups: errors and warnings. A constraint violation is an error if it leads to an incomplete import of a MOST element (block, function or parameter) or even if the corresponding MOST element cannot be imported at all. All import problems that could only lead to a partial loss of information (for example, when a date attribute could not be parsed) are considered as warnings.

If an error occurred it could be the case that no MOST FBlock could be imported.

The error and warning messages have a special format containing the exact position in the XML file (the line and column number of the corresponding XML element), the MOST element position in the form

```
<Block>.<Function>.<Parameter1>.<Parameter2>…
```

and the message itself describing the constraint that was violated.

Let us consider some examples of possible warnings.

```
1) [23, 17] AMFMTuner.ATWaveband.: Obsolete function sections found
2) [2654, 40] AMFMTuner.TestData.TunerData.: No valid discriminator
     parameter found.
```

The first warning message should be read as follows:
At line 23 and column 17 in the XML file an obsolete function section was found in the function "ATWaveband" that is a part of the block "AMFMTuner".

The second warning message means the following:
At line 2654 and column 40 an XML element describing a discriminator parameter was expected at the parameter "TunerData" referred from the function "TestData" that is referred from the block "AMFMTuner".

3.2.4.1 Warnings

Nr	Text
1	*The date {0} is incorrect.* A date is expected to be in one of the following formats: a) "yyyy-MM-dd" (for example, "2007-12-21") b) "yyyyMMdd" (for example, "20071221") c) "dd.MM.yyyy" (for example, "21.12.2007") d) "dd MMMM yyyy", locale ENGLISH (for example, "21 December

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

	2007") e) "yyMMdd" (for example, "071221")
2	*The function classification 'Array of Record' is deprecated. The function classification 'Array' will be used instead.*
3	*The function classification 'Dynamic Array of Record' is deprecated. The function classification 'Dynamic Array' will be used instead.*
4	*The function classification 'Long Array of Record' is deprecated. The function classification 'Long Array' will be used instead.*
5	*Obsolete function sections found*
6	*The parameter must be of type Array.*
7	*The parameter must be either of type Record or of type Array.*
8	*There is an OPType with several parameters on the same position.*
9	*In a Record there are several parameters on the same position.*
10	*In a StreamCase there are several parameters on the same position.*
11	*The element does not have a valid class identifier.*
12	*The element does not have a valid subclass identifier.*
13	*The function has an unknown classification. A valid function classification is assigned.*
14	*The element does not have a valid identifier.*
15	*The flag "details" must be set to "false" when a dummy parameter is exported.*
16	*The parameter must be of type CStream.*
17	*A parameter of type Number must have a corresponding TypeRef.*
18	*The parameter must be of type ShortStream.*
19	*The attribute value is incorrect.*
20	*For a parameter of type Stream: TypeRef="type_stream".*
21	*The parameter must be of type String.*
22	*For a parameter of type Void: TypeRef="type_void".*
23	*No valid discriminator parameter found.*
24	*Multiple discriminator parameters found.*
25	*The ramified stream parameter does not contain a discriminator parameter.*

3.2.4.2 Errors

Nr	Text
1	*Unknown visibility* An unknown visibility value was found.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

2	*Unknown function classification.*
3	*Unknown OPType*.
4	*The parameter must be of type Boolean.*
5	*For an enumeration parameter: TypeRef="type_enum".*
6	*For a stream parameter: TypeRef="type_stream"*.
7	*The element has an invalid number format*.
8	*Unknown parameter type: {0}. A valid parameter type is assigned*.
9	*There are multiple definitions of an OPType in a function*.
10	*For a boolean parameter: TypeRef="type_boolean"*.
11	*The parameter must be of type Enumeration*.
12	*In an ordered collection only one object may be set to one position*.

3.2.5 FCat MOST Export Errors and Warnings

3.2.5.1 Errors

Nr	Text
1	*Both units with names {0} and {1} have the same 'ID'. Please assign a unique "ID".*
2	*The unit with the name {0} has an empty ID. Please assign the unit a unique "ID".*
3	*The unit with the name {0} has an invalid ID. Please assign the unit a valid "ID".*
4	*The XML export requires that at least one FBlock should be exported. No FBlock was selected for the export*. (DTD format only)
5	*Infinite recursion in parameters: {0}*
6	*The parameter {0} of type "ArrayOfRecord" must contain a parameter of type "Record"*.
7	*At least one OPType must be provided.*
8	*Multiple occurrences of the parameter {1} in OPType {0} are not allowed.*
9	*The OPType {0} is not allowed.*
10	*The parameter {0} of type "Record" may not contain parameters of types Void or Record.*
11	*The parameter {0} of type "Stream" must contain only parameters of the following types: Boolean, BitField, Number, Enumeration, String, Stream, CStream, ShortStream, Length-Coded String, Array Type, Record Type or Void.*
12	*The parameter {0} of type "Record" must contain at least one parameter*.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

13	*The parameter {0} of type "Array" may not contain a parameter of type "Void".*
14	*The parameter {0} of type "Array" must contain a subparameter.*
15	*The parameter {0} of type "Number" must contain a unit*.
16	*The parameter {0} of type "BitField" must contain at least one value range*.
17	*The parameter {0} of type "Boolean" must contain at least one value range*.
18	*The parameter {0} of type "Enumeration" must contain at least one value range*.
19	*The function {0} must contain the discriminator parameter {1}. Otherwise the discriminator parameter cannot be exported*.
20	*The parameter {0} of type "Stream" contains a discriminator parameter that cannot be exported due to missing Stream cases.*
21	*The complex stream parameter {0} does not have a discriminator parameter.*
22	*OPType {0} expects at least one parameter*
23	*The parameter {0} of type "Record" may contain a parameter of type 'Array' only on the last position of the record.*
24	*The parameter {0} of type Number must contain a unit.*
25	*The parameter {0} has an invalid value range that lies outside of the available data type values.*
26	*The parameter {0} has overlapping value ranges.*
27	*OPType {0} does not expect a Parameter*
28	*OPType {0} expects at least one parameter*
29	*OPType {0} expects {1} parameters*
30	*OPType {0} expects exactly one parameter*
31	*The parameter {0} of type Array contains the attribute "size" with the wrong value: {1}. Allowed values: [0-9]+|Nmax*
32	*OPType {0}: [Specific to FCat 0.6.05.3 format] Expecting {Boolean (legacy)(UnsignedWord), Boolean (legacy)(UnsignedByte), Boolean (legacy)(UnsignedLong)} instead of Number(UnsignedWord)*

3.2.5.2 Warnings

Nr	Text
1	*The following functions, parameters and signals are not referenced and are thus not saved: {0}*
2	*OPType {0} does not expect a Parameter.*
3	*The parameter {1} of type "Stream" in OPType {0} must be on the last position*.
4	*The parameter {0} of type "Record" may contain a parameter of type "Array"*

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

	only on the last position of the record.
5	*The parameter {0} of type "Record" may contain a parameter of type "Stream" only on the last position of the record.*
6	*The parameter {0} of type "Stream" contains a discriminator parameter that cannot be exported due to missing Stream cases*.
7	*The complex stream parameter {0} does not have a discriminator parameter*.
8	*The discriminator parameter {1} of the stream parameter {0} has no corresponding optype*.
9	*The parameter {0} of type "ArrayOfRecord" must contain a parameter of type "Record"*.
10	*Parameter "{0}" uses invalid stream discriminator: selector needs to precede the streams it controls.*

3.2.5.3 Function constraint errors

The constraints below describe the dependency of the types of function parameters on the function type.

Nr	Text
1	*A function of type "Number" may contain only direct parameters of the following types: Number, Void.*
2	*The funktion {0} contains a discriminator parameter {1} that cannot be exported (not BitEncoded).*
3	*A function of type "Text" may contain only direct parameters of the following types: String, Void*.
4	*A function of type "Enumeration" may contain only direct parameters of the following types: Enumeration, Number, Void.*
5	*A function of type "Container" may contain only direct parameters of the following types: CStream, Void, Stream and ShortStream.*
6	*A function of type "Record" may contain only direct parameters of the following types: Record, Number, Void*.
7	*A function of type "Array" may contain only direct parameters of the following types: Array, Number, Void*.
8	*A function of type "Sequence Property" may contain only direct parameters of the following types: Boolean, BitField, Number, Enumeration, String, Stream, CStream, ShortStream, Array Type, Record Type, Void*.
9	*A function of type "Unclassified Property" may contain only direct parameters of the following types: Boolean, String, Enumeration, BitField, Stream, CStream, ShortStream, Record, Array, Array Type, Record Type, Void*.
10	*A function of type "Trigger" may contain only direct parameters of the following types: Number, Void*.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

11	*A function of type "Switch", "Boolfield" or "Bitset" may contain only direct parameters of the following types: Boolean, BitField, Void*.
12	*A function of type "Sequence Method" may contain only direct parameters of the following types: Boolean, BitField, Number, Enumeration, String, Stream, CStream, ShortStream, Array Type, Record Type, Void*.
13	*A function of type "Unclassified Method" may contain only direct parameters of the following types: Boolean, String, Enumeration, BitField, Stream, CStream, ShortStream, Record, Array, Array Type, Record Type, Void.*

3.2.6 FIBEX Export Errors and Warnings

3.2.6.1 Errors

Nr	Text
1	*The function {0} does not have a valid identifier. Its value must be between 0 and 4095.*
2	*The function {0} having a component-by-component access has a Record parameter {1} with less than two fields.*
3	*The function {0} having a component-by-component access does not have a Pos parameter*.
4	*The parameter {0} of type "Record" must contain at least one record field.*
5	*The parameter {0} of type "Stream" is bit-encoded, but lacks a discriminator parameter.*
6	*The parameter {0} of type "Stream" has a discriminator parameter, but is not bit-encoded*.
7	*The parameter {0} of type "Stream" has a stream case without code*.
8	*The parameter {0} of type Stream containing several cases must have a discriminator parameter*.
9	*The parameter {0} of type "Stream" (or "ShortStream") has a stream case without parameters and signals*.
10	*The signal {0} has no valid value for length*.
11	*The block {0} does not have a valid identifier. Its value must be between 0 and 255*.
12	*The parameter {0} of type "Stream" with many stream cases must contain a discriminator parameter*.

3.2.6.2 Warnings

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

Nr	Text
1	*The function {0} has modifications that cannot be exported (their change or reason fields are empty).*
2	*The parameter {0} of type "Stream" with one stream case is not bitEncoded. Therefore its code cannot be exported and is redundant.*
3	*The parameter {1} of type {0} with one stream case has a description that cannot be exported*.
4	*The block {0} has modifications that cannot be exported (their change or reason fields are empty).*

3.2.7 FIBEX Import Errors and Warnings

3.2.7.1 Errors

Nr	Text
1	*Unknown OpType: {0}*

3.2.7.2 Warnings

Nr	Text
1	*In a Record there are several parameters on the same position.*
2	*In a StreamCase there are several parameters on the same position.*
3	*The coding of a boolean/bitfield parameter has a bad format: {0}. It must have the following format: (-)*[01](-)**.
4	*No parameter with Id {0} was found.*
5	*Unknown function classification: {0}.*
6	*Unknown parameter type: {0}*.

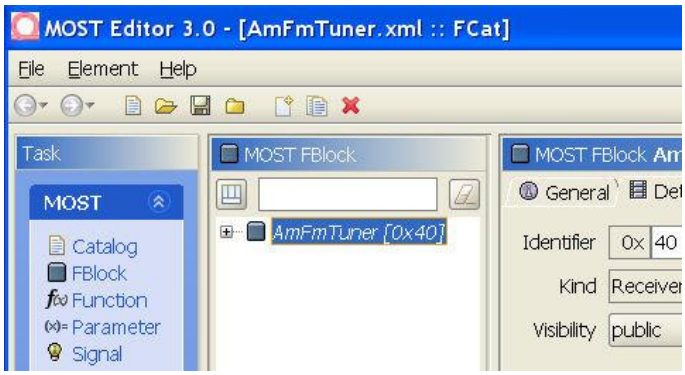
3.2.8 MOST Editor Window Title

Unless a function catalog was loaded (with “File” → “Open”) or saved for the first time in a file (with “File” → “Save”) the window title contains only the version number.

Otherwise it consists of three parts:

```
MOST Editor <Version Number> - [<File Name> :: <File Format>]
```

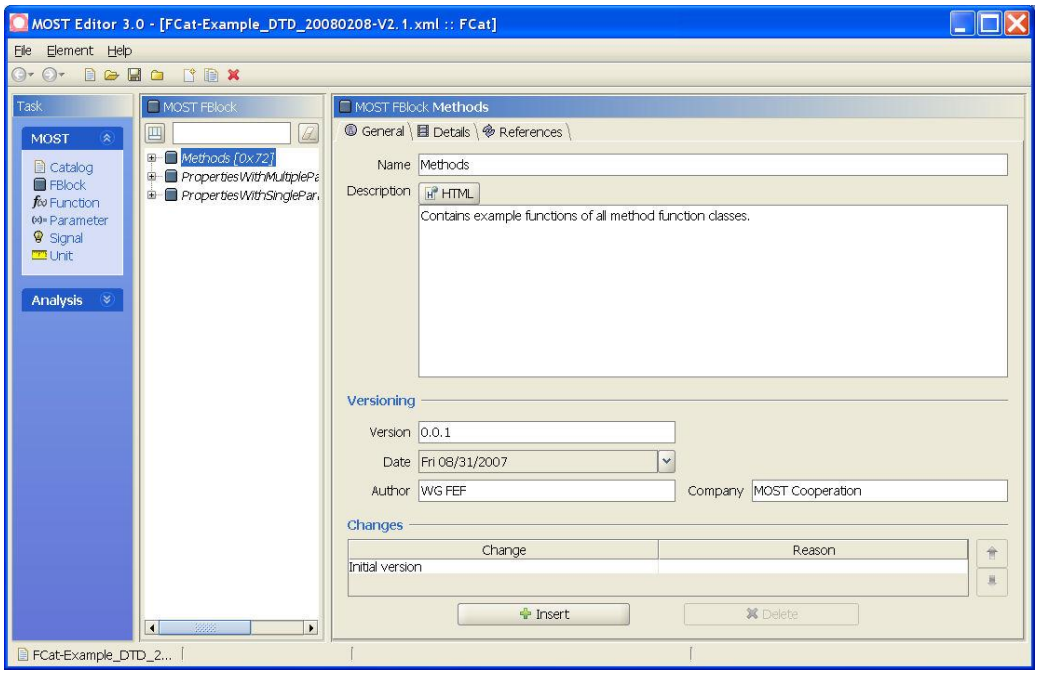
© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.



Two file formats are possible: FCat and FIBEX. In this particular example, the file "AmFmTuner.xml" was loaded in the format FCat by the MOST Editor of the version 3.0.

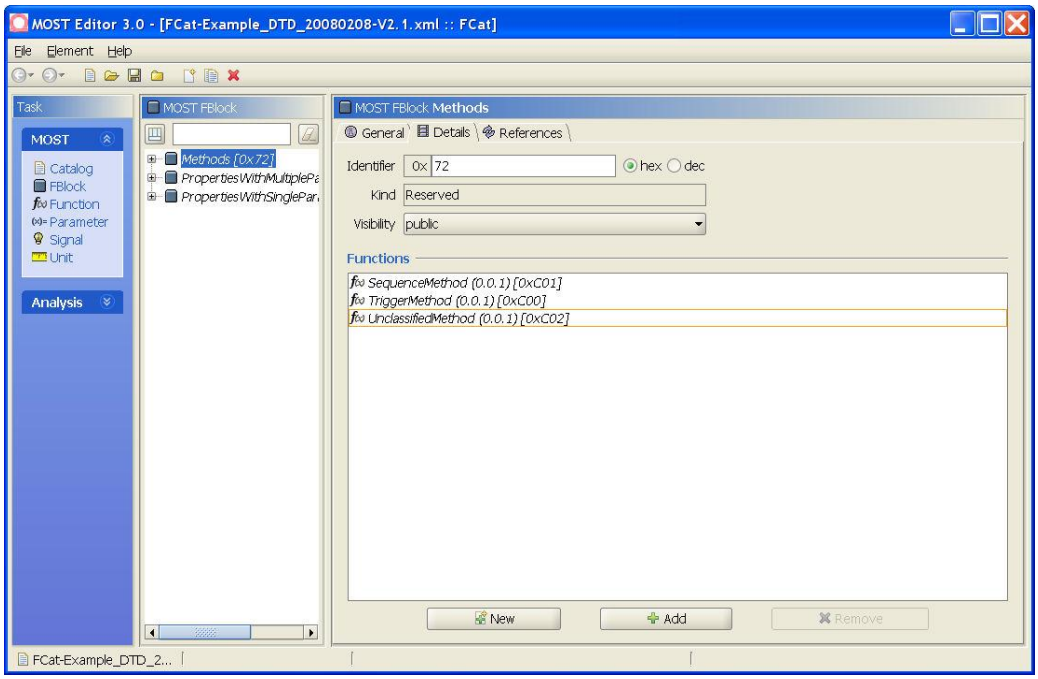
3.3 Working with MOST FBlocks

A function catalog needs at least one MOST FBlock. A MOST FBlock has three tree tabs in the detail editor. In the "General" tab you can edit the name and description of the FBlock. You can use the versioning and the changes section to maintain a change history of your FBlock.



In the "Details" tab you can set the identifier and the visibility of your FBlock and attach functions.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

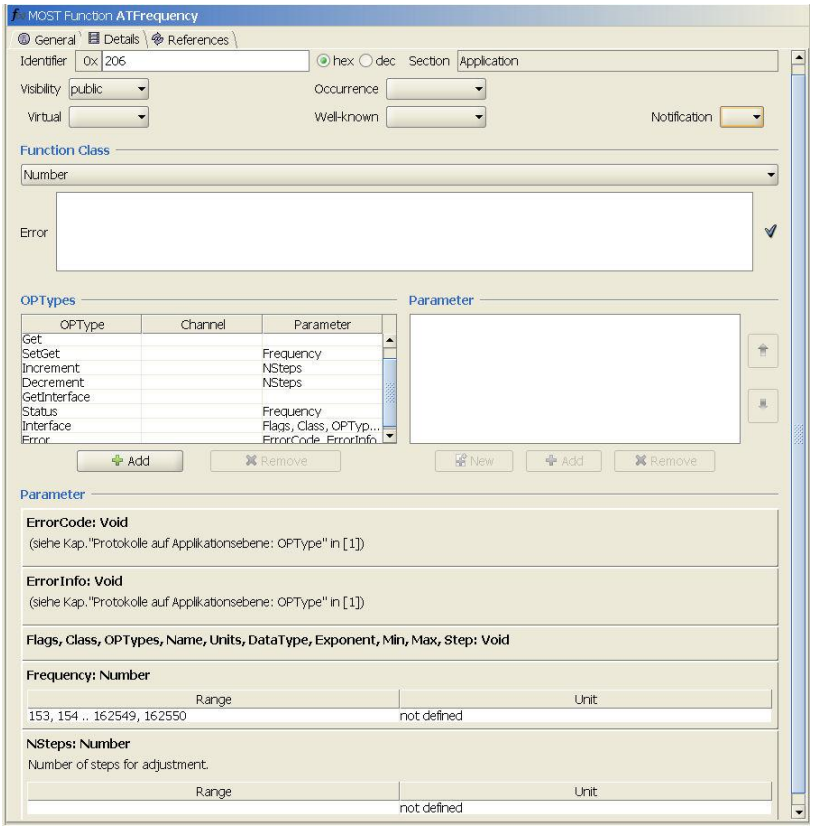


If you select “New” at the bottom of the “Details” tab a new function is created and attached to the FBlock. If you select “Add” an existing function is attached to the FBlock. To remove a function from the selected FBlock mark it and click on the “Remove” button. The removal from an FBlock does not delete the function. It still exists and can be reused in different FBlocks.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

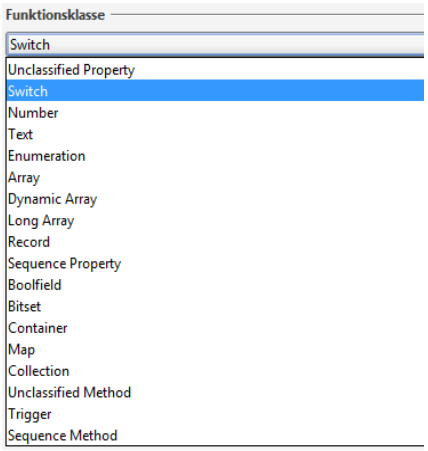
3.4 Working with functions

The key tab to edit functions is the “Details” tab in the detail editor.

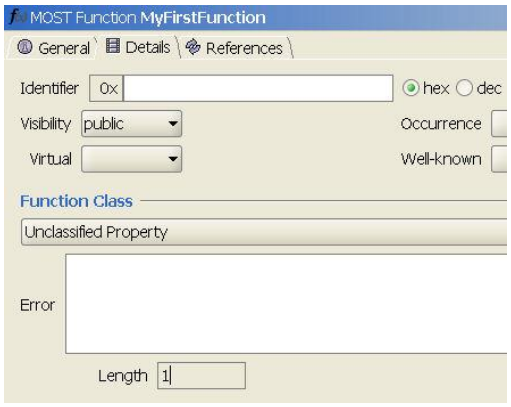


It has three Boolean-valued attributes: Virtual, Well-known and Notification. The attribute Occurrence can take three values: CONDITIONAL, MANDATORY and OPTIONAL. The four attributes can be left empty. Depending on the value of the attribute Identifier the calculated attribute Section can take its values.

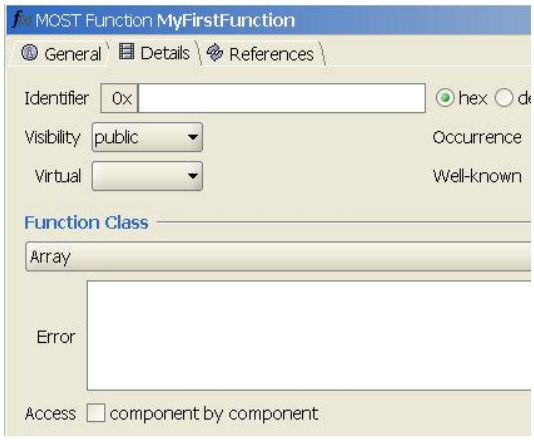
You can set the attributes of your function, add and remove OPTypes. When you select an entry in the OPTypes section by clicking on it you can add new (Button “New”) and existing (Button “Add”) parameter to your OPType. At the bottom part of the panel the function parameters with their short description are shown. The attribute “Function Class” can take the following values:



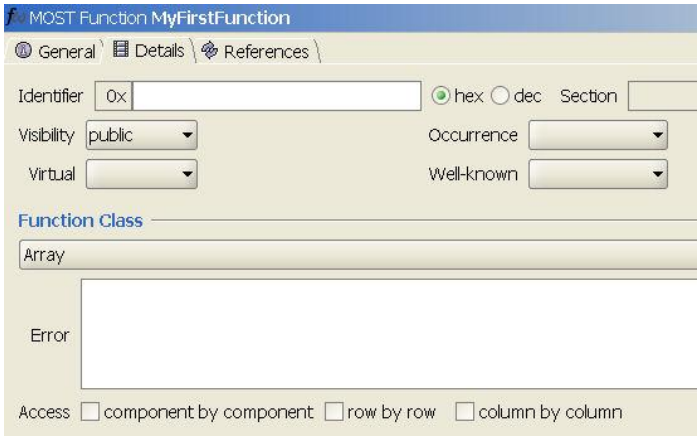
© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.



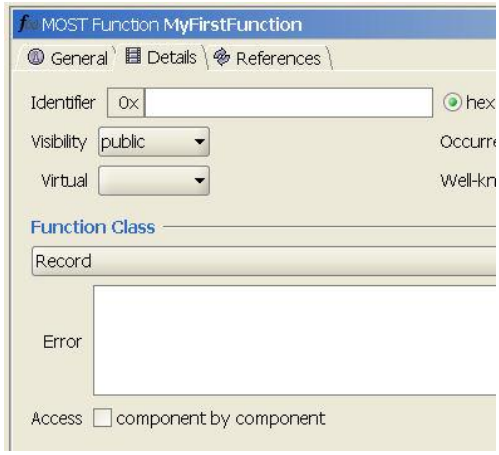
A MOST function with the function class „Unclassified Property“ contains an attribute Length. The attribute is not editable. It can get its value when the function is loaded from an FCat file. Its value will be exported to an FCat file when the function is saved.



A MOST function with the function classes “Array”, “Dynamic Array” or “Long Array” contains one or more boolean access attributes depending on the subparameter type of the Array parameter of the function.
It contains one boolean access attribute “component by component” if the subparameter type differs from Record.



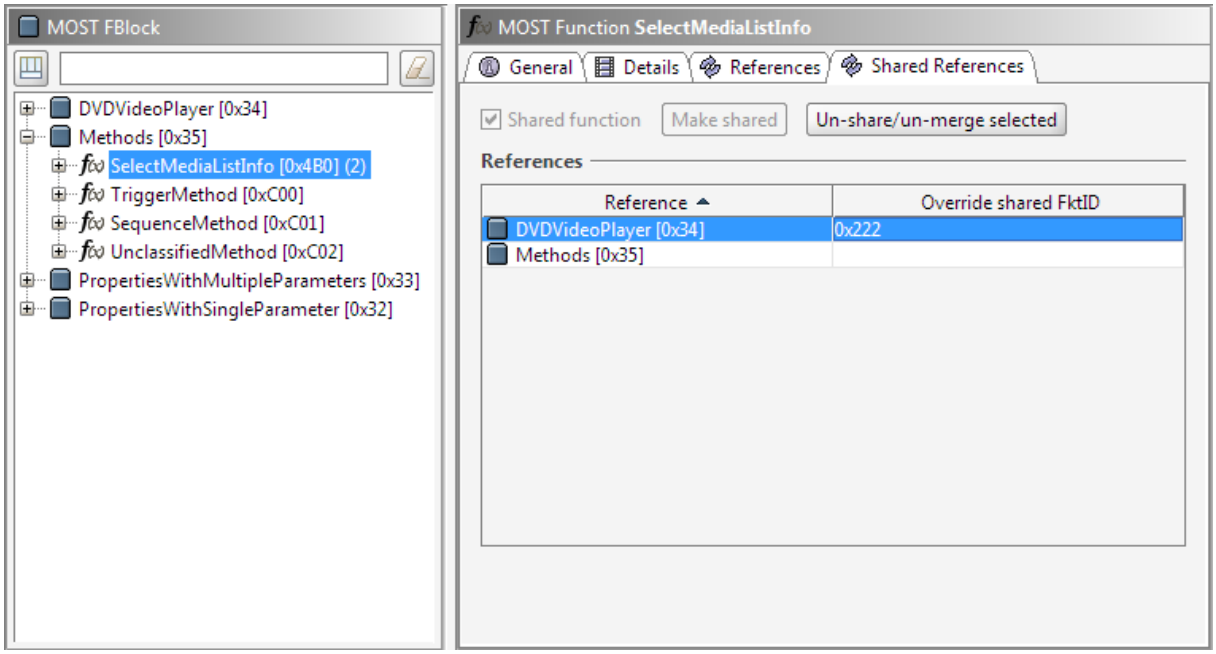
If the Array-parameter of the function has a Record-parameter as its subparameter then the function contains three boolean access attributes: “component by component”, “row by row” and “column by column”.



A MOST function with the function class Record contains one boolean access attribute “component by component”.

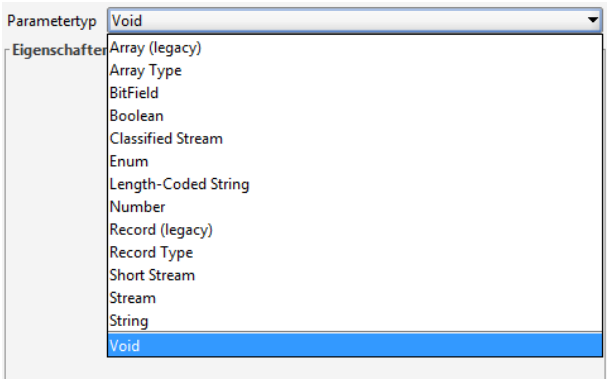
© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

For shared functions (i.e. functions used by different FBlocks) the MOSTEditor supports overwriting the shared FktID in the tab “Shared references”. In the table “References” all referencing FBlocks are listed. To overwrite the FktID edit the entry in the “Override shared FktID” column. In the following example FktID 0x4B0 of Function “SelectMediaListInfo” has been overwritten to 0x222 for “Methods”. The overwritten FktID is displayed italic in the tree view:



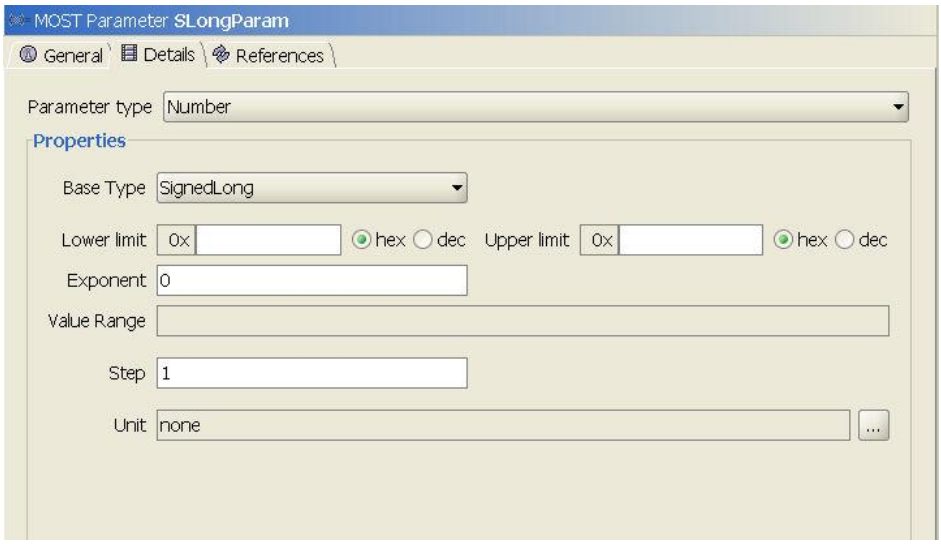
3.5 Working with parameters

The Details tab of a parameter is separated in two areas: From a drop-down menu you can select a parameter type. Depending on the parameter type the properties area lets you enter the details required for the selected parameter type.

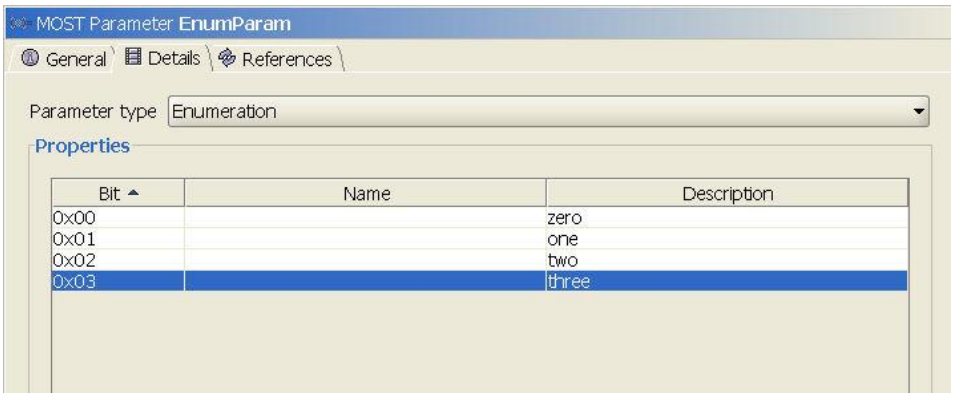


Parameter type void has no additional properties but number for example

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.



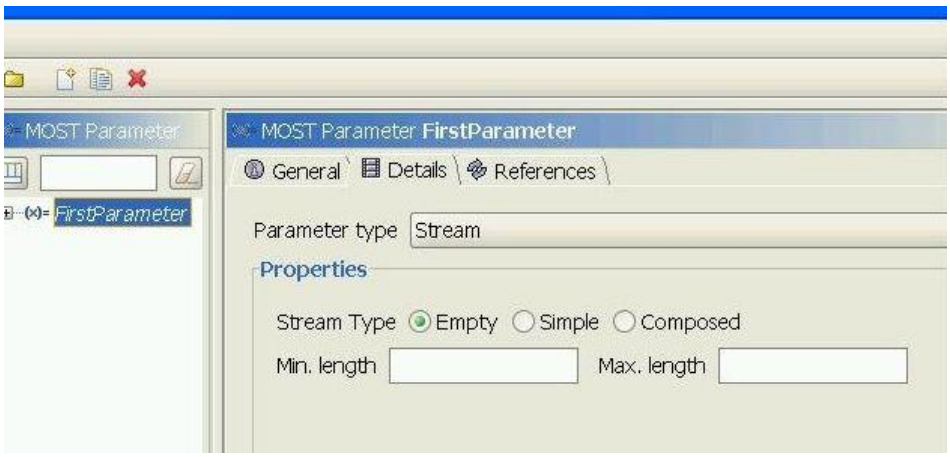
or enumeration



need additional information to be entered.

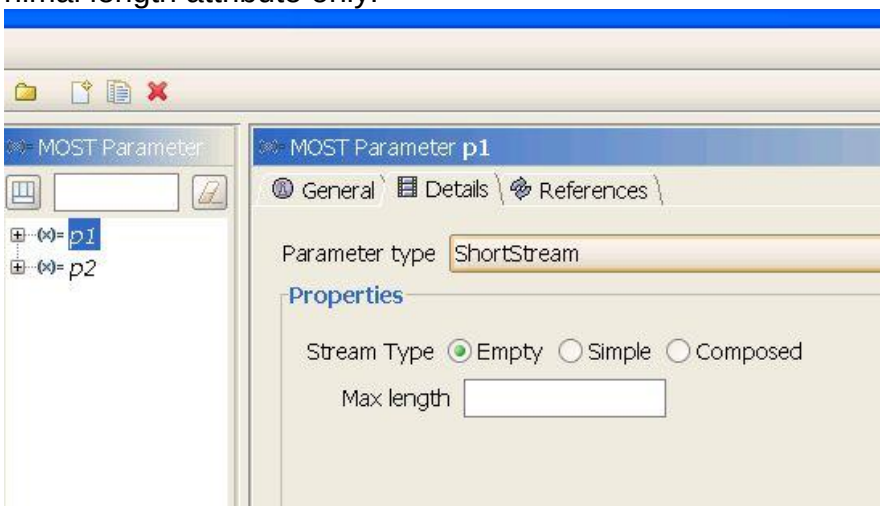
3.5.1 Stream parameter

There are three kinds of stream parameters: empty (without a stream case), simple (with exactly one stream case and without any discriminator parameter) and composed (with one or several stream cases and with a discriminator parameter).

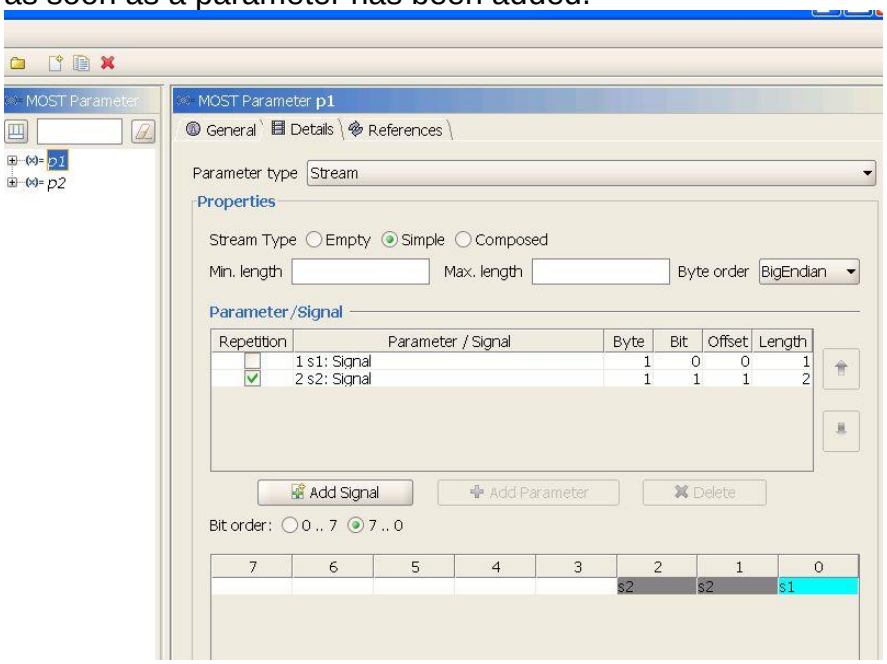


© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

A short stream parameter differs from a stream parameter in the absence of the minimal length attribute only.

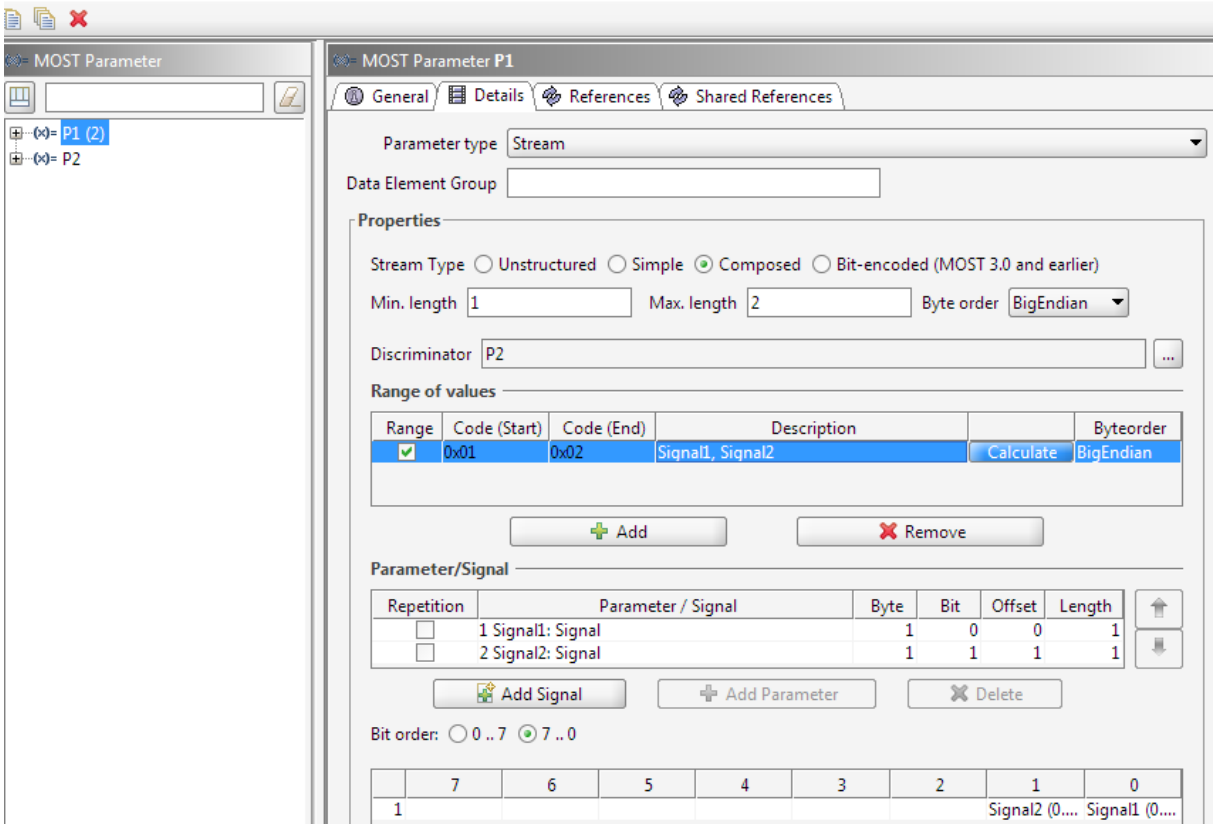


A simple stream is a stream with exactly one use case and without a discriminator parameter. A simple stream may consist of either signals or parameters. Signals and parameters cannot be mixed. A simple stream consisting of signals has a byte order attribute and the signal layout of the simple stream is displayed at the bottom. Its bit order can be reversed. The byte order attribute and the signal layout view make sense only for stream cases containing signals. The byte order attribute disappears as soon as a parameter has been added.



A composed stream is a stream containing a discriminator parameter. It may contain one or several stream cases. As soon as the composed stream type is selected, a discriminator parameter selection dialog is shown. If no parameter is selected, the simple stream type will be chosen instead.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

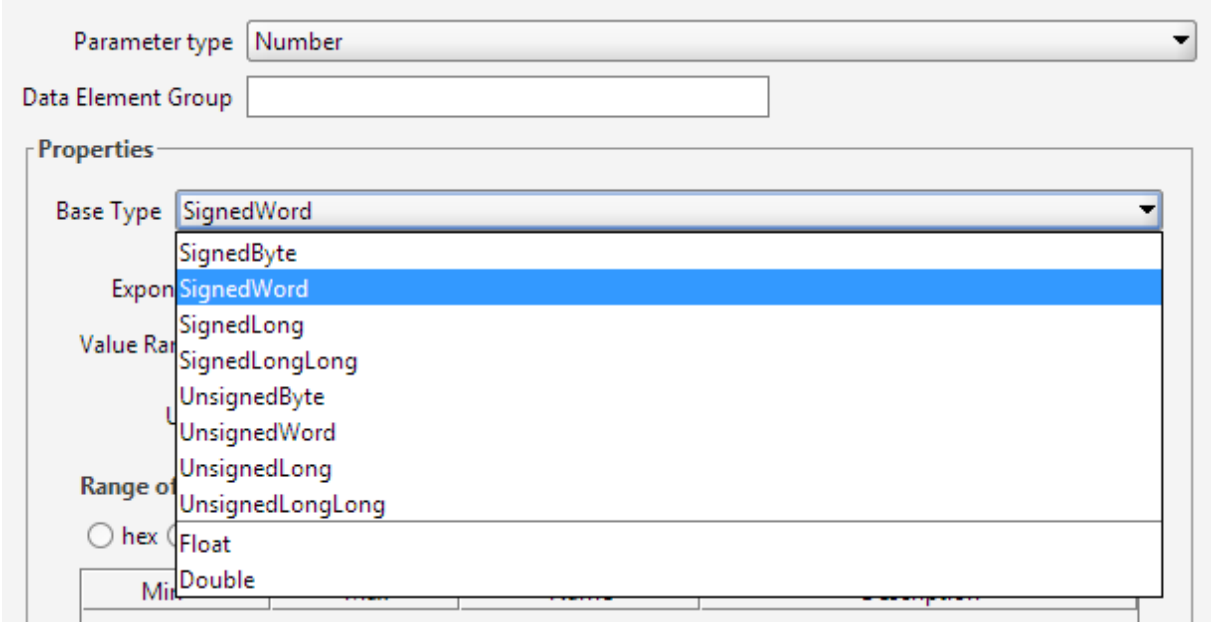


For every stream case a range with a start and end code can be defined. The end code can only be inserted if the checkbox “Range” in the first column has been selected. The inserted values will be checked and marked red if they are not compatible with the type of the underlying descriminator parameter. A tooltip is shown with further information. In a composed stream with several stream cases the description attribute is editable and the button “Calculate” is active. The button “Calculate” is used to calculate the stream case description on the basis of the selected parameters or signals.

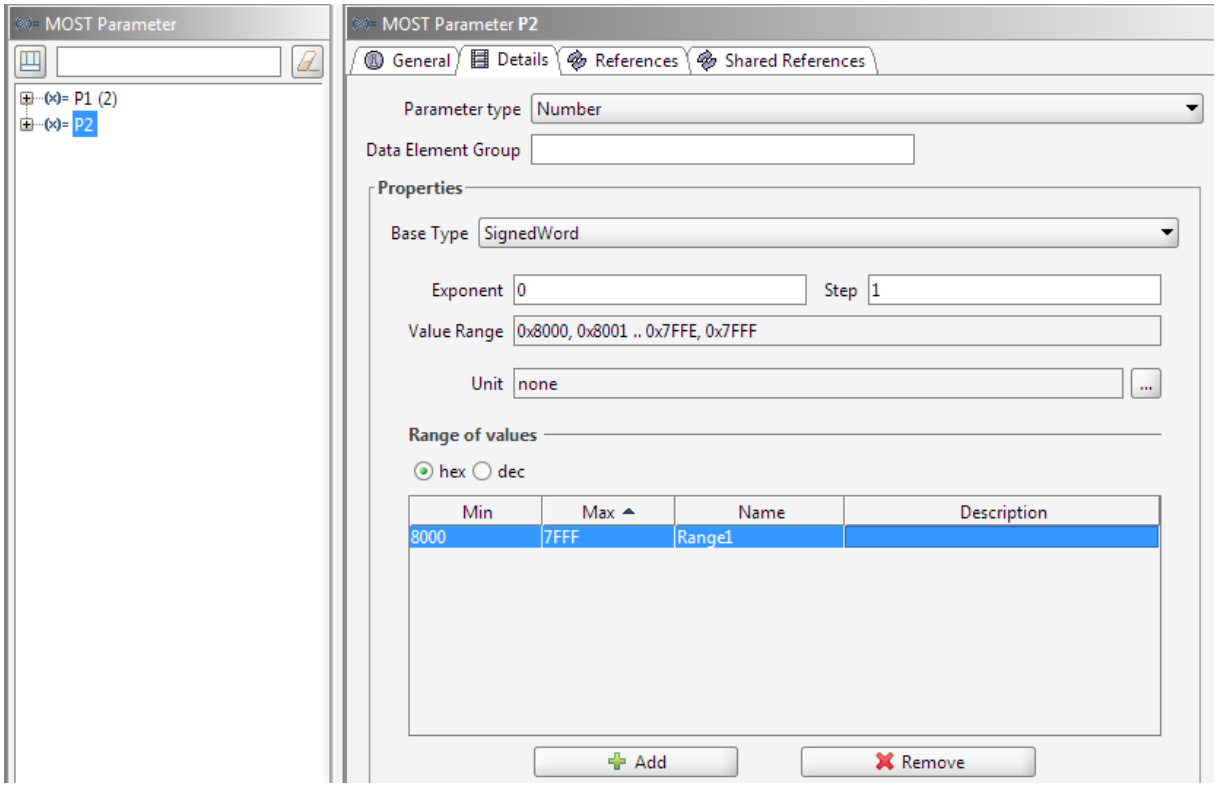
3.5.2 Number parameter

For number parameters various signed or unsigned base types are available:

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

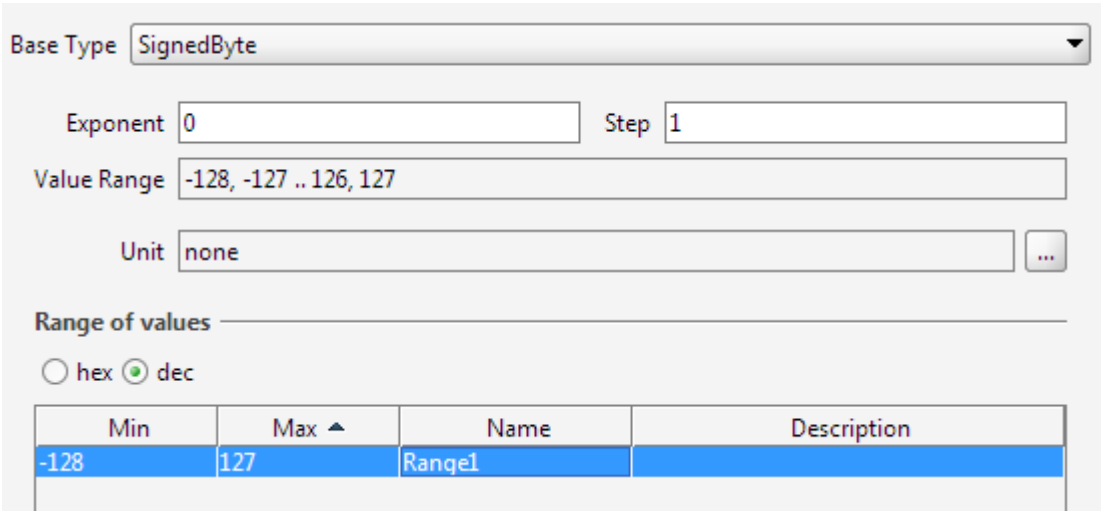


A range of values can be defined. Lower limit and upper limit values for every range can be entered in two formats: hexadecimal and decimal. Hexadecimal values must be positive. The value range field shows interpreted values (including negative values) according to the defined base type, exponent and step:

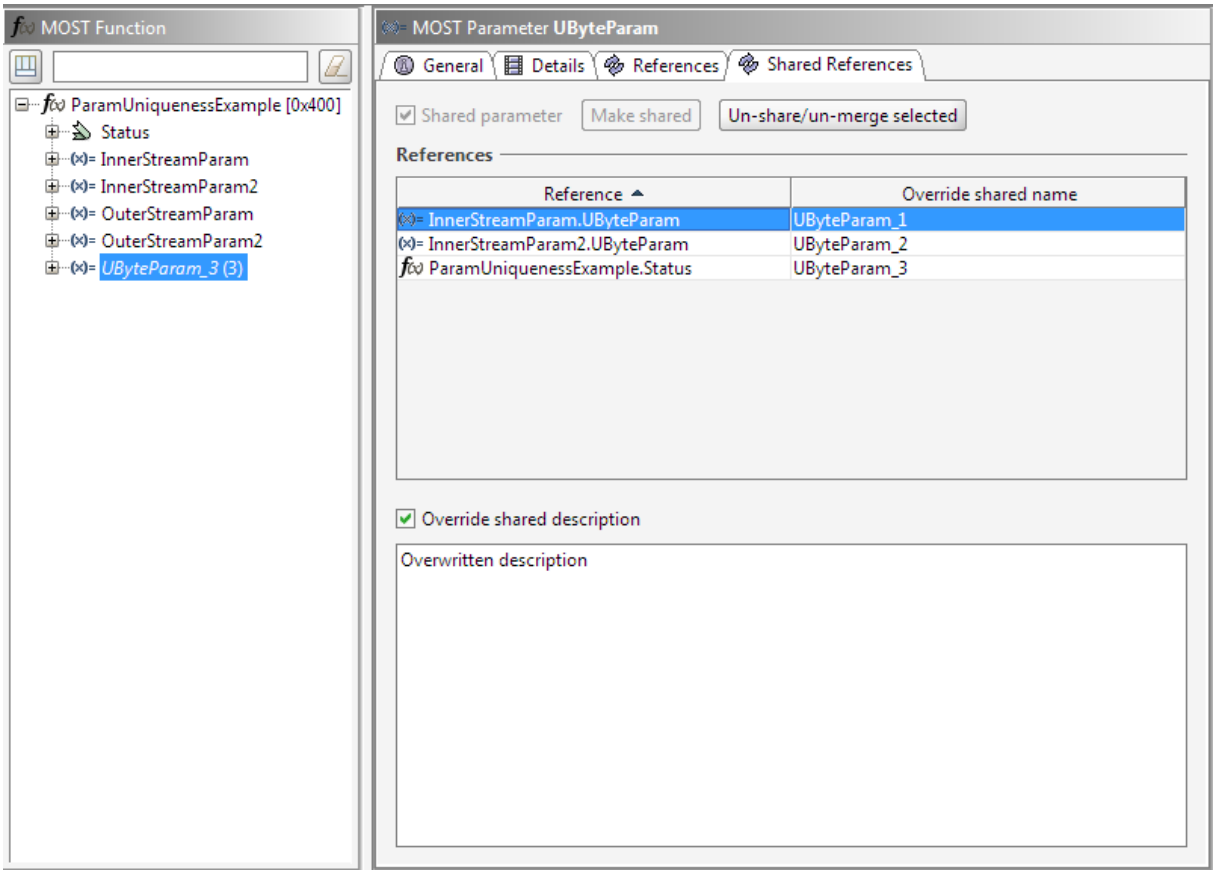


One can enter negative values in decimal format. The upper and lower limit values must be within the corresponding base type value range:

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

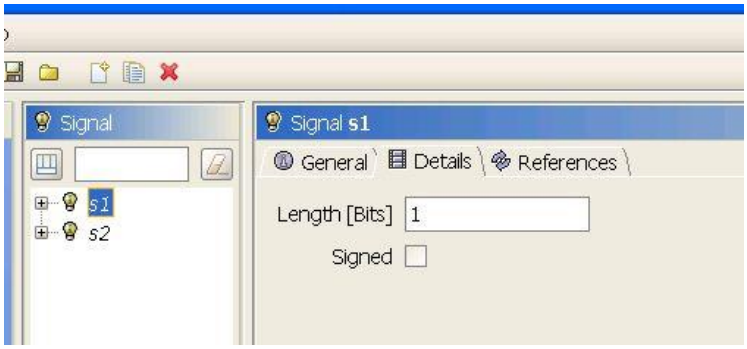


For shared parameters (i.e. parameters used by different functions and/or parameters) name and description can be substituted in the tab “Shared references”. In the following example Parameter “P1” is used in two functions. For “Function2” name and description has been ovewritten to “Renamed”. The overwritten name will be displayed italic in the tree view:



© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

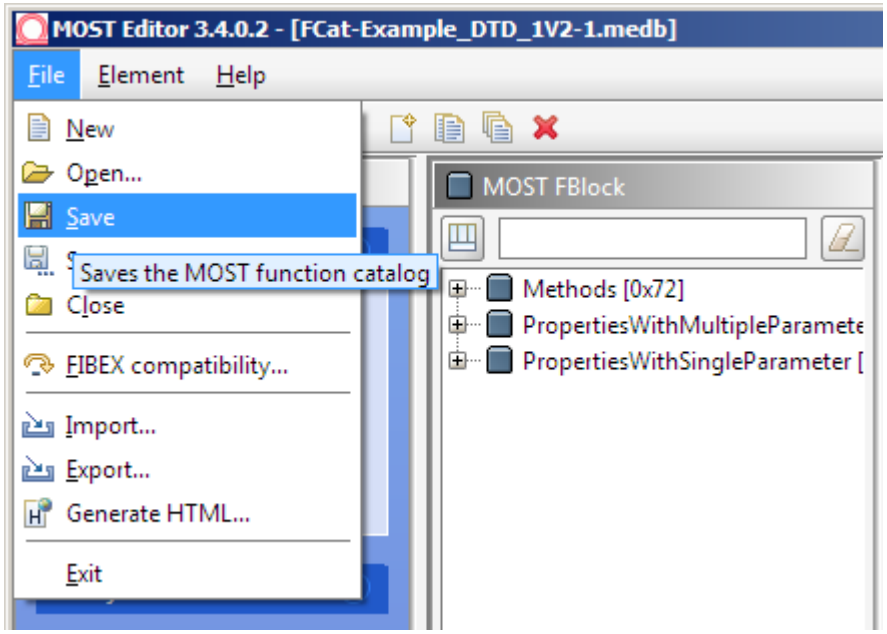
3.6 Working with signals



A signal can be signed or unsigned. It must have a length (in bits).

3.7 Saving a function catalog

You can use “File” → “Save” from the main menu or  from the toolbar to save a function catalog. Your function catalog is stored as MOST database file (*.medb).
If you opened an existing medb file containing a function catalog, worked on it with the MOST Editor and then clicked save your original xml file will be overwritten.

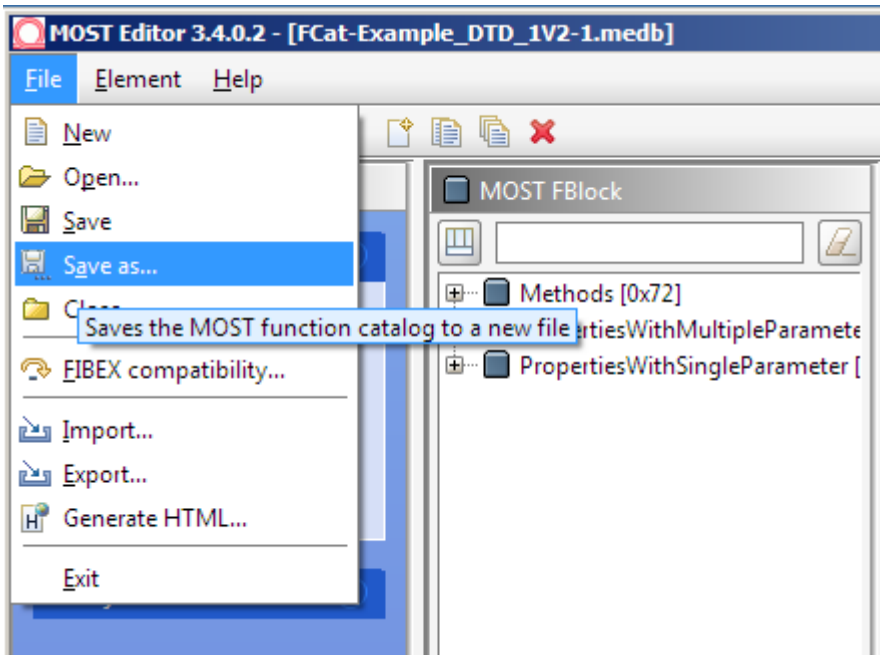


If you want to save your changes in a new file use “File” → “Save as” from the main menu.

3.8 Saving as new function cataloge

If you use “File” → “Save as” from the main menu you can choose parts of your current function catalog to be saved into a new file.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.



In the MOST Editor the original function catalog remains open. If you want to view the newly created function catalog you have to open it in your MOST Editor.

When you have made changes in your MOST Editor and you try to open a function catalog or create a new function catalog the MOST Editor asks you if you want to save your changes.

While editing your function catalog may be inconsistent but when you save it your function catalog must be consistent according to the specification of the MOST Co-operation.

The following conditions generate warnings when trying to save a function catalog:

- A function catalog that does not contain an FBlock.
- A function that is not attached to an FBlock.
- A parameter that is not attached to a function.
- A signal that is not attached to a parameter.

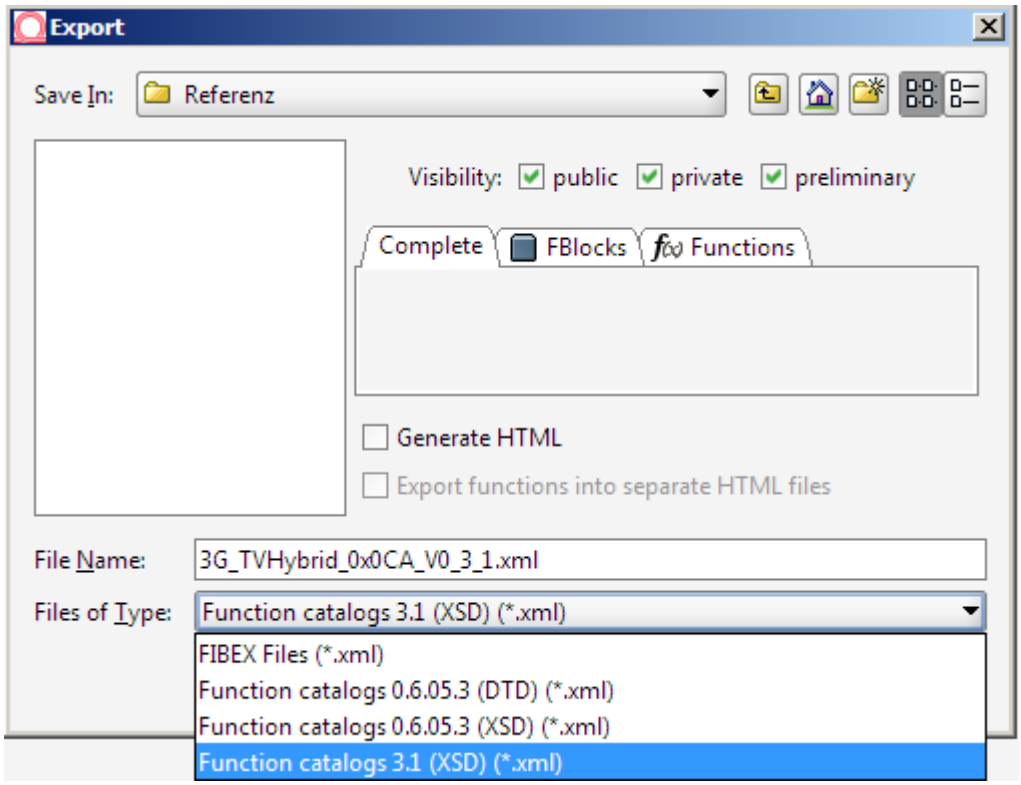
The function catalog is saved without the elements that are not correctly specified. The MOST Editor remains open so that you can correct your function catalog.

3.9 Exporting

If you use “File” → “Export” from the main menu you can choose parts of your current function catalog to be saved in a new xml file. Different export formats are supported.

The following pop-up allows you to define which parts of your function catalog you want to save:

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.



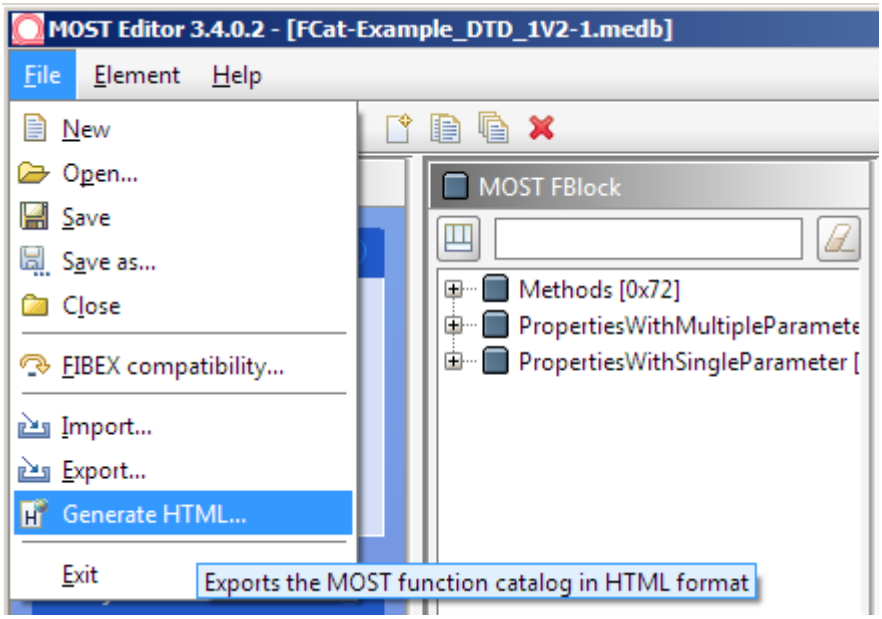
Only those blocks and functions will be exported that have chosen visibility values. A new function catalog containing only the selected elements is created on your disk.

One can export all blocks and functions or any part of them into one of two formats: FIBEX and FCat. The format can be chosen at the bottom part of the dialog.

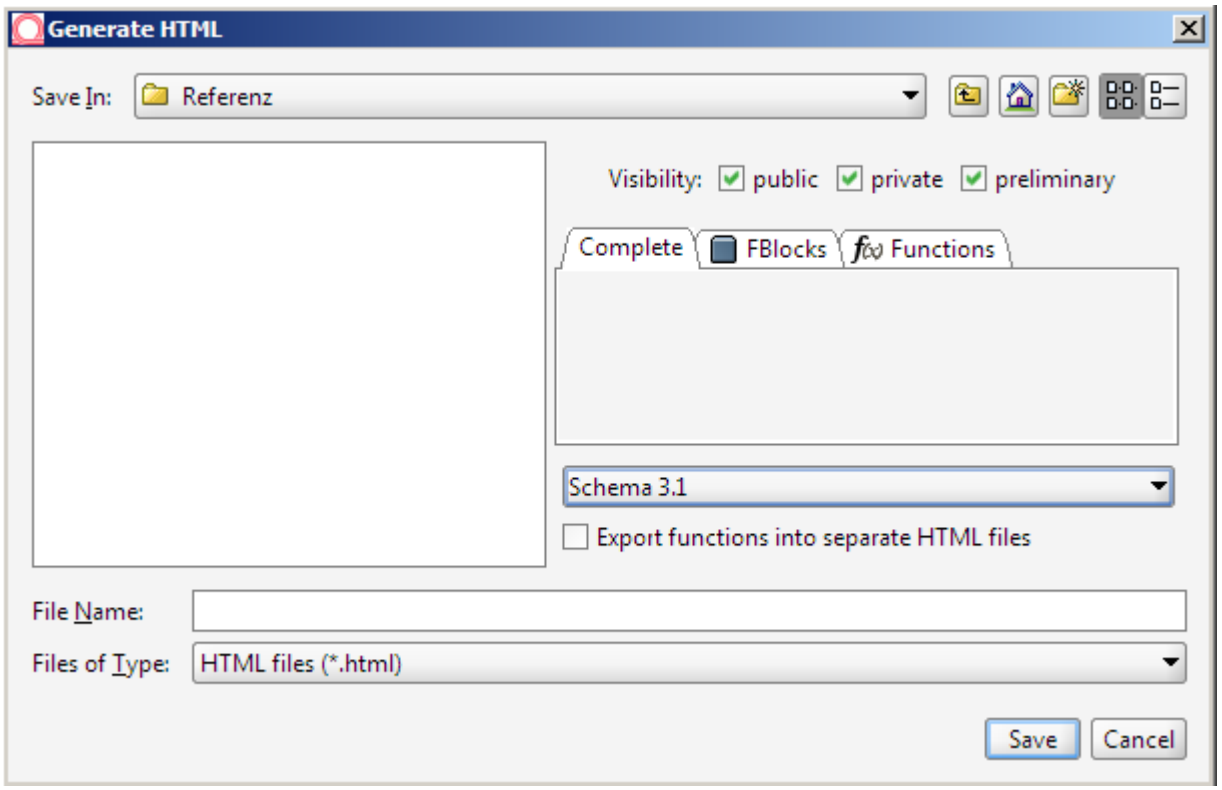
© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

3.10 Generating html output

For generating an html file that contains your function catalog select “File” → “Generate HTML” from the main menu.

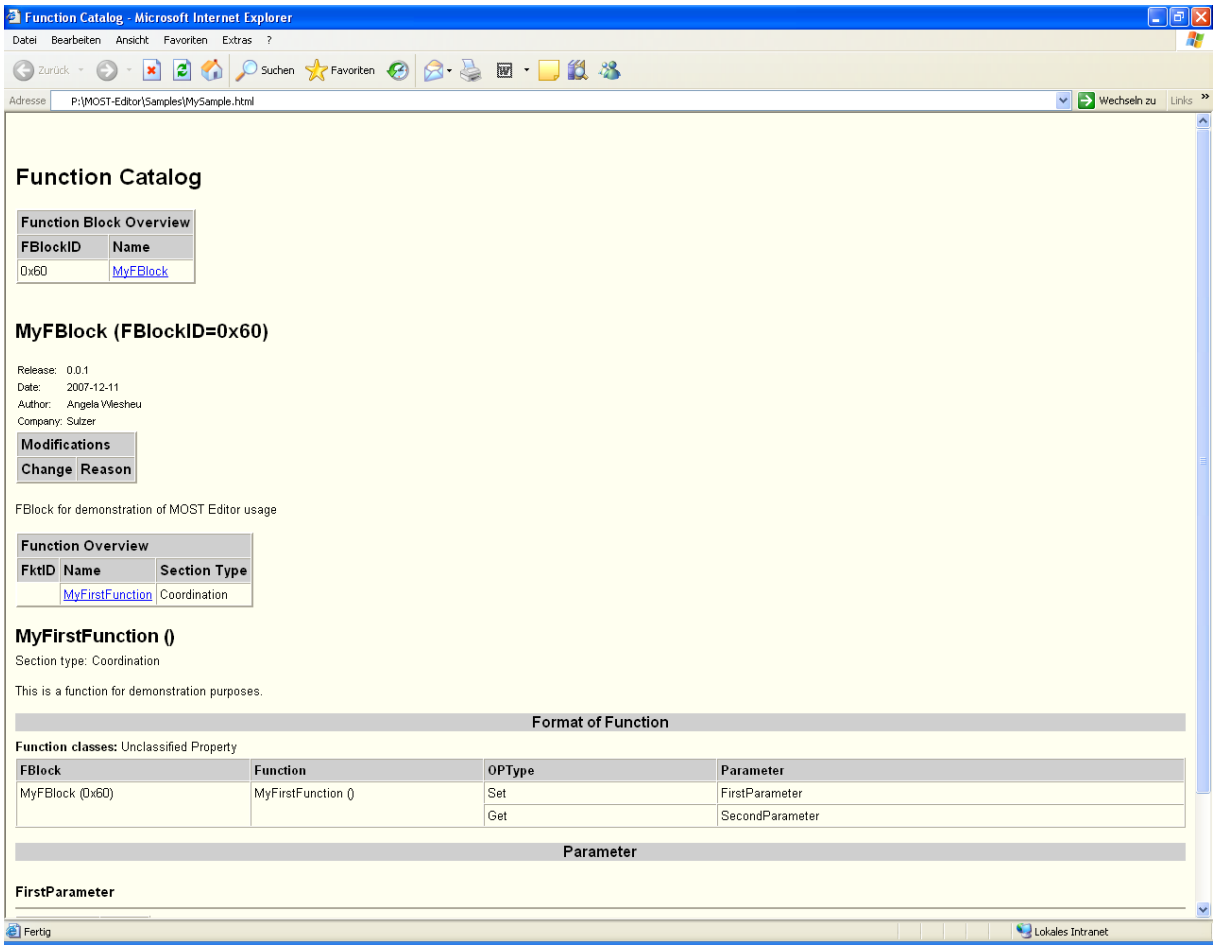


A pop-up window opens where you can customize the html export. You can save your whole function catalog or only parts of it as html document.



© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

You can select the stylesheet and an option to export functions into separate HTML files.
In addition to the html file a file called most.css is saved. This file prettifies the html output with background color and the like. Without this file you still can view your function catalog in a browser but it doesn't look as nice as it does with most.css in place.



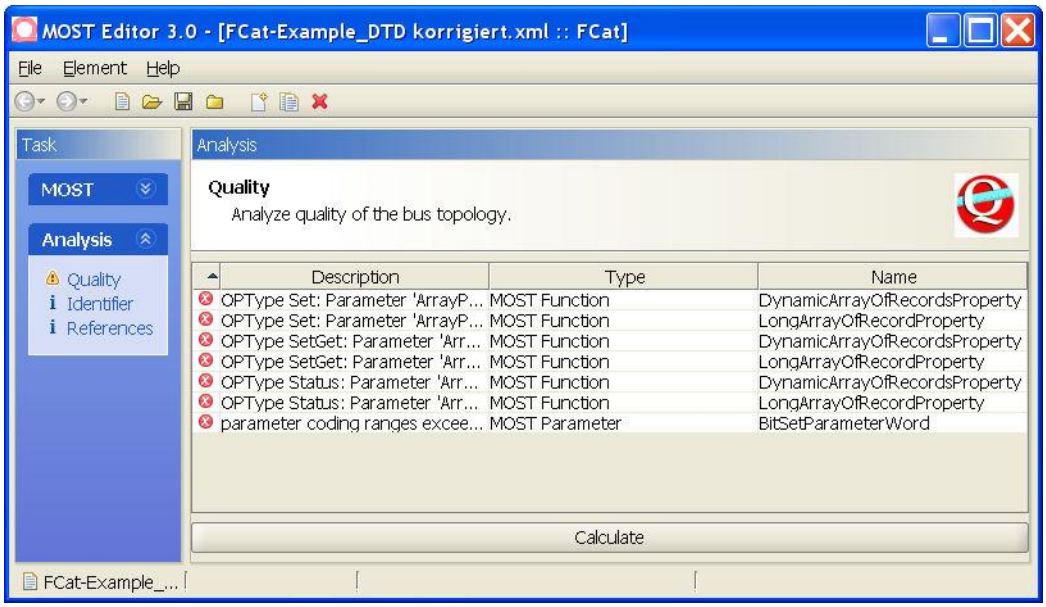
3.11 Analyzing a function catalog

The MOST Editor offers functionality to analyze a function catalog to get an overview of the elements it contains and to identify inconsistencies.

3.11.1 Quality

When you select "Quality" in the task tree and press the "Calculate" button at the bottom of the detail editor all items that seem to be incorrect are listed.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

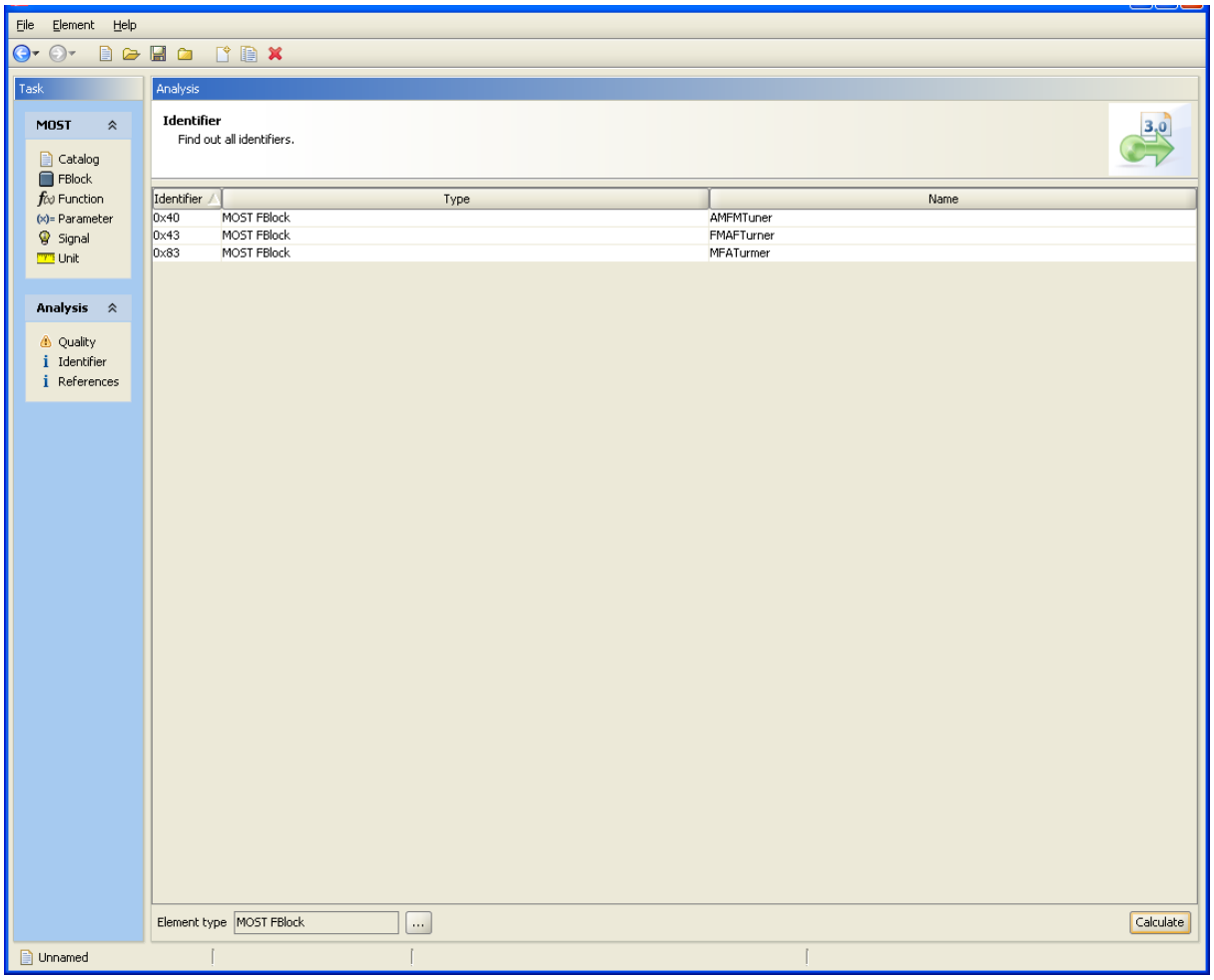


A double click on a quality analysis entry opens the respective editor window where you can correct the error.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

3.11.2 Identifier

By choosing Identifier from the task tree and specifying the element type at the bottom of the detail editor you can list all elements of the selected element type ordered by their identifier.

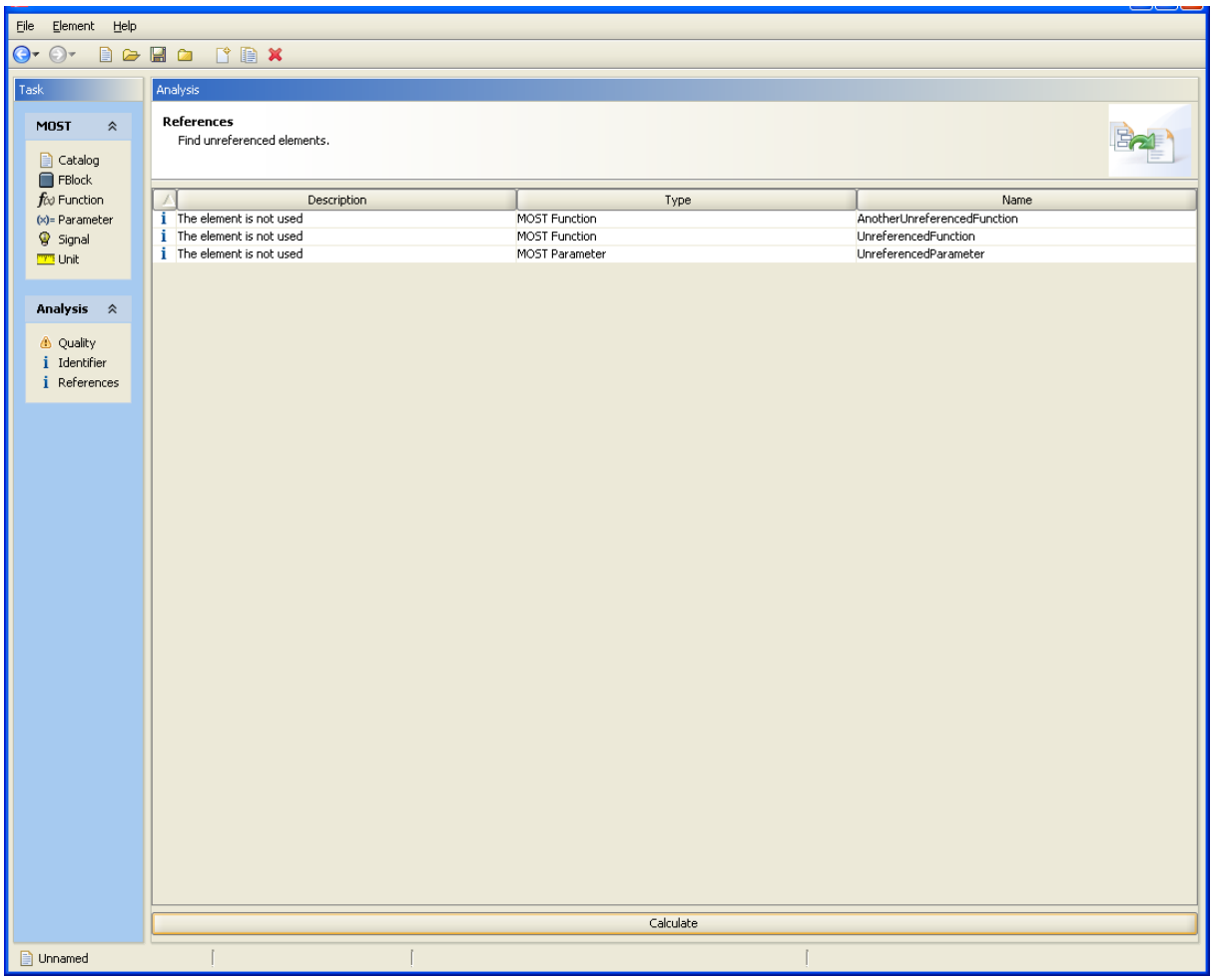


A double click on a line opens the respective editor window.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

3.11.3 References

In the references section you can find all unreferenced elements. This is especially important as you cannot store unreferenced elements.

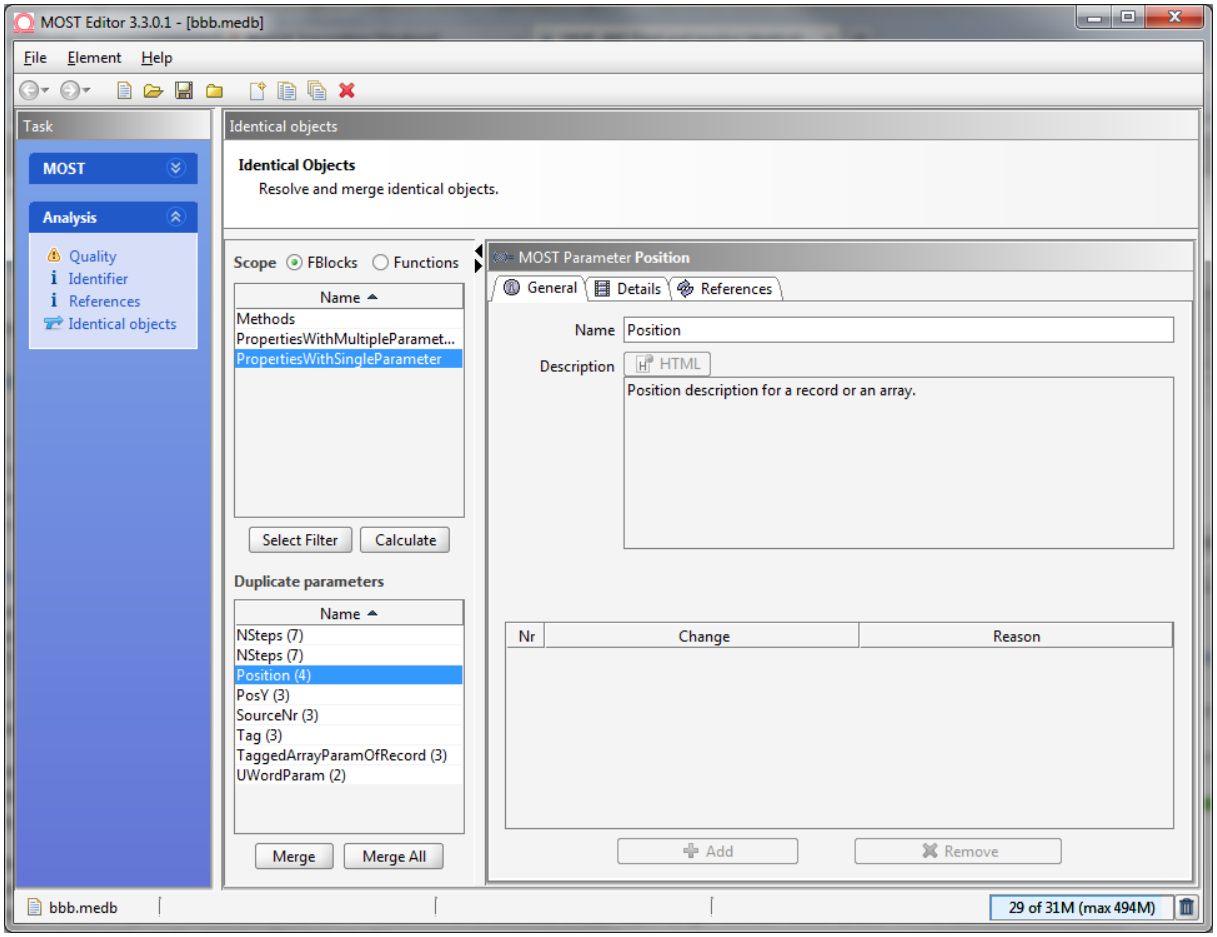


A double-click on a line opens the respective editor window.

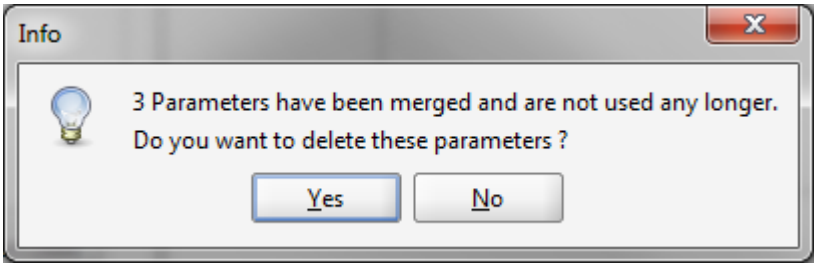
© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

3.11.4 Identical objects

In the identical objects section you can identify all identical parameters for a given search scope:



To define a search scope use the “Select Filter” button and select one or more of either FBlocks or Functions from the provided list. Press “Calculate” to start searching for identical parameters within that scope. All duplicate parameters will be shown in the list of duplicate parameters below. You can select one or more parameters from that list and pressing the “Merge” or “Merge All” button will remove all duplicate parameters within the referenced elements. If the merged parameters are not used anymore outside the search scope a message dialog pops up:

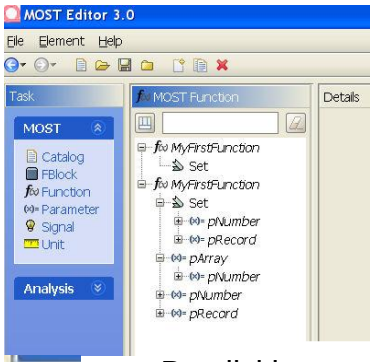


Confirming with “Yes” will delete these unused parameters permanently. If you select a parameter from the duplicate parameters list the related details of that parameter will be show on the right side.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

3.12 Toggle between tree/table element viewing modes

MOST elements can be viewed in two modes: tree and table. In the tree mode one can clearly see the hierarchical structure of the elements.

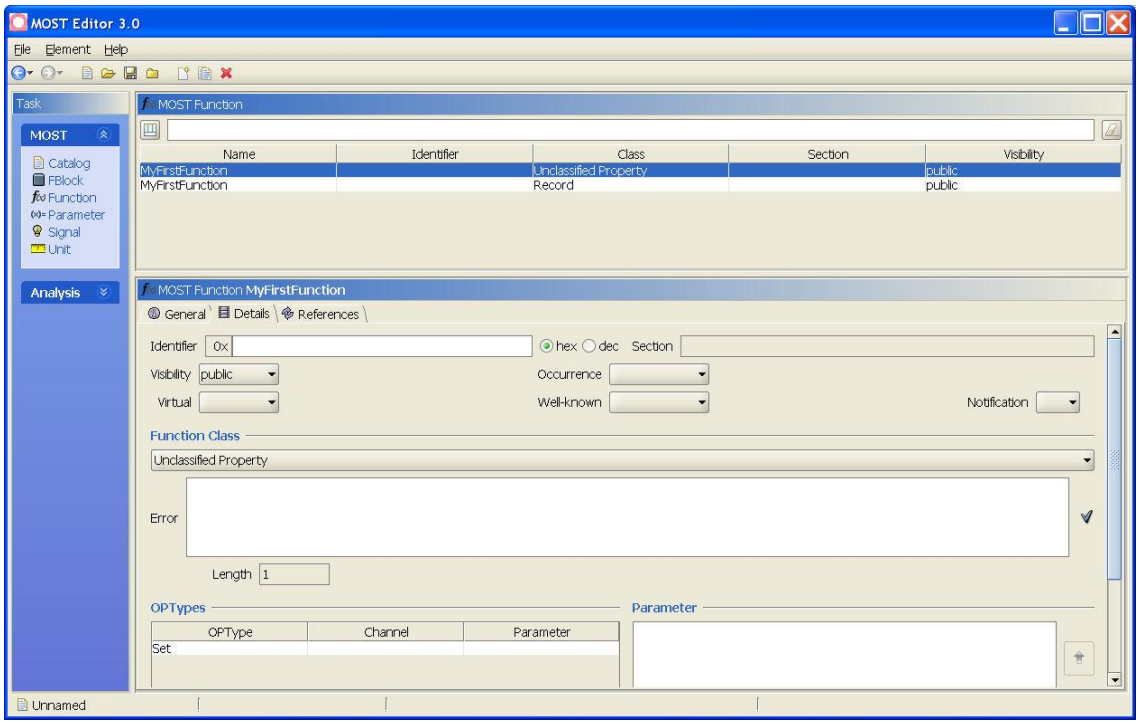


In the tree mode one can clearly see the hierarchical structure of the elements.



By clicking on the icon one can switch back and forth between tree and table modes.

In the table mode one can see the values of the most important attributes of the elements.



3.13 Comparing two FBlocks that are stored in different function catalogs

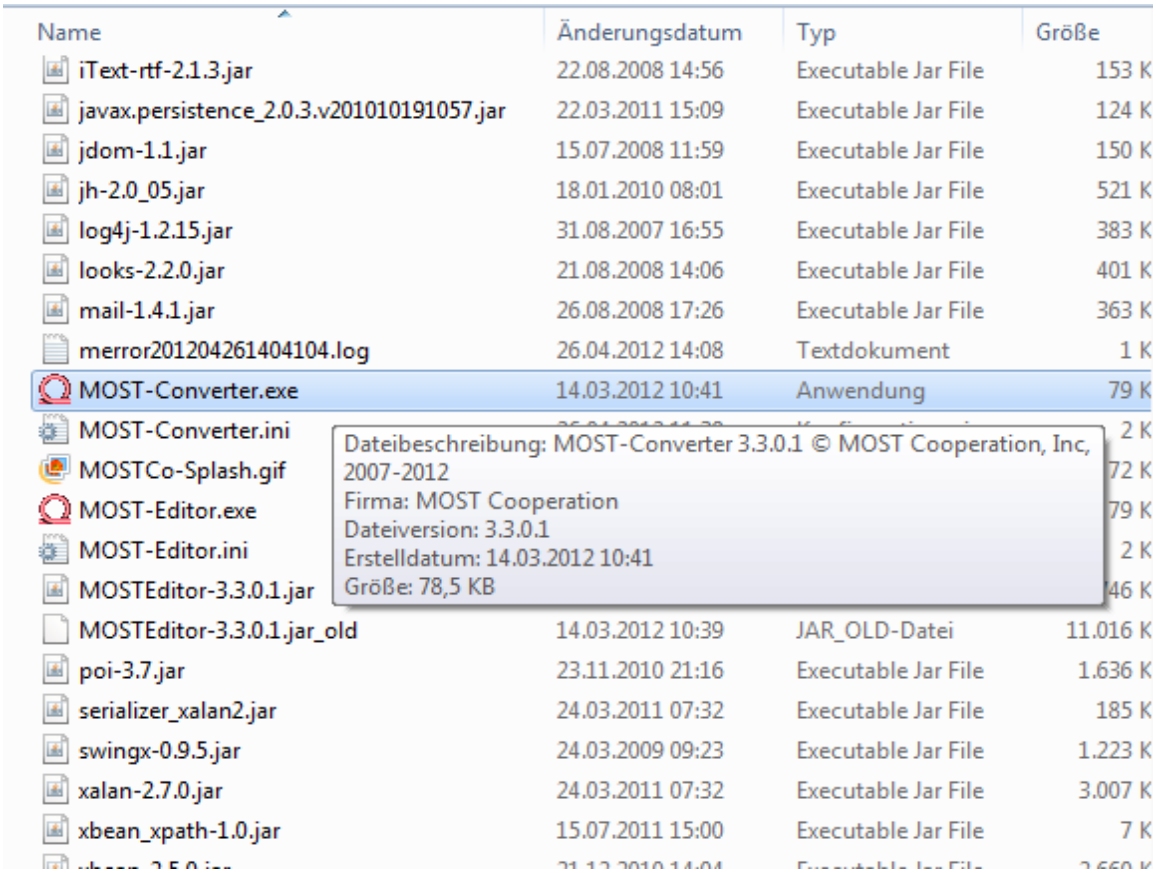
You can compare two FBlocks that are stored in different function catalogs by importing the FBlock to compare in your current FBlock.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

1. Open the first function catalog in your MOST Editor.
2. Import the FBlock from the other function catalog (see section 3.2.3 Importing from existing function catalogs). The original and the imported FBlock are displayed in the detail tree.
3. Now, you can mark both FBlocks and choose “Compare with each other” from the context menu (press right mouse button). See section 3.1.7 Compare elements.

3.14 Command line converter

The MOST-Editor includes a small command line converter which allows performing conversions between export formats or running the quality checker. The command line converter is distributed as “MOST-Converter.exe” within the standard installation directory of the MOST-Editor:



Usage:

most-converter.exe –convert --targetDTD6V5_3 --log <log_file1> <input_file1> <output_file1> <input_file2> <output_file2>

most- converter.exe --check --silent --log <log_file1> <input_file1>

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

For using the converter the command line arguments needs to specify a command as first attribute. The following commands are supported:

--convert (convert between input and output file(s) using a format)
--check (quality check of the input file(s))

"--convert" is used for conversion between formats and "--check" is used for running the quality checker.
When specifying "--convert" one of the following target formats must be specified:

--targetDTD6V5_3 (FCat DTD)
--targetFIBEX3V1 (FCat Fibex)
--targetSchema0V6_5 (FCat XSD V065)
--targetSchema3V1 (FCat XSD Rev. 3.1)

Optionally the following arguments are allowed:

--log (specify the name of the log file)
--silent (disables information output on the screen)

Examples:

most-converter.exe --convert -- targetSchema3V1 --log log.txt Input1.xml Output1.xml Input2.xml Output2.xml

Convert the input files (*Input1.xml Input2.xml)* to the output files *(Output1.xml Output2.xml*) using FCat Rev. 3.1 XSD format and create a logfile named *„log.txt"*

most-converter.exe --check –silent --log log.txt Input1.xml

Do a quality check of the input file (*Input1.txt)* in silent mode and create a log file named *„log.txt".* The result of the quality check will be written to a file named „*Input1.xls"*

For using blanks in file or path names you need to embrace the input and output parameters with quotation marks.

© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

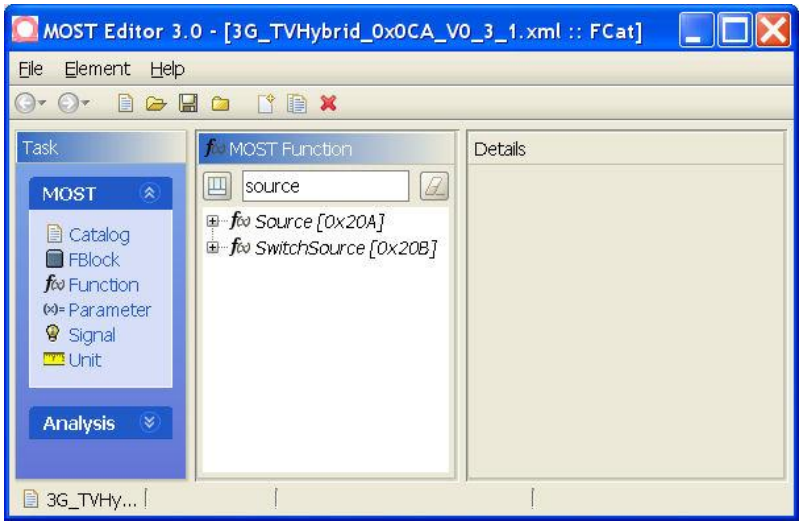
4 Principles and concepts guiding the MOST Editor

4.1 Top-down approach

The MOST Editor is guided by a top-down approach. You can start creating your high level elements and add the details later on. You start for example by creating an FBlock, then you create functions that you attach to your FBlock and parameters for your functions. Then you add the details to your FBlocks, functions, parameters and signals. So you can start with the big picture and then work down to the details.
Of course you can also work bottom-up with the MOST Editor by defining the parameters that you will need first, then all the functions and finally your FBlocks but the more convenient way to create a function catalog usually is the top-down approach.

4.2 Infix-Filter

The infix-filter in the detail tree is a very powerful tool for searching elements.

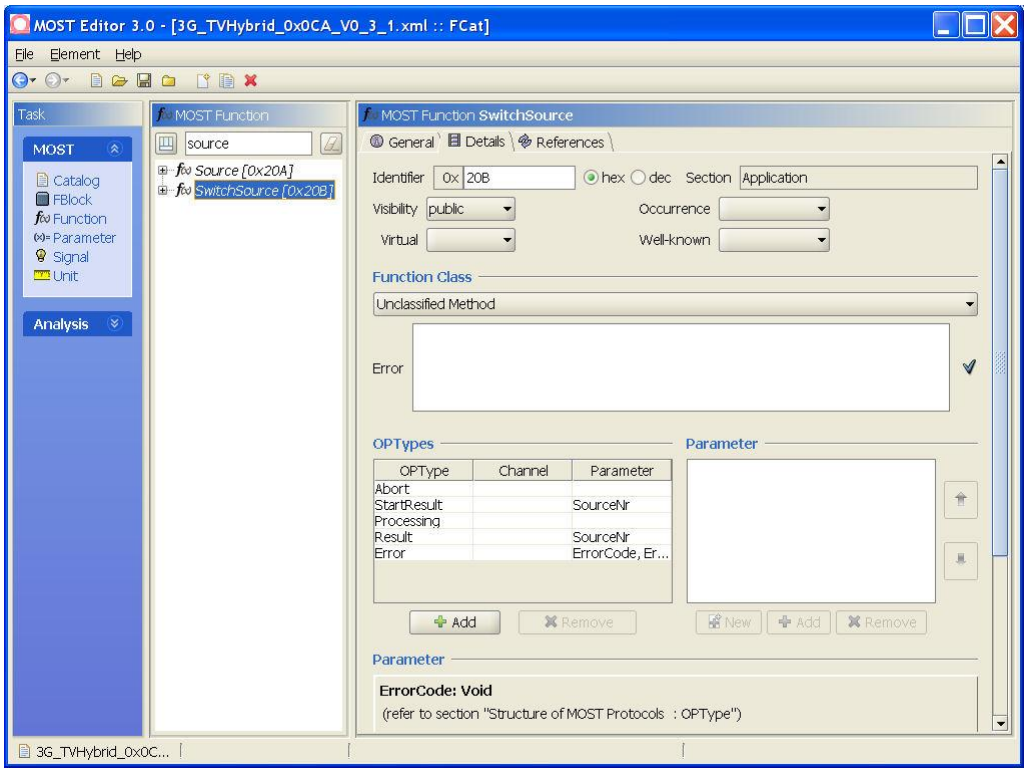


In the task tree you select an element type for example “Function”. In the detail tree all functions are displayed. When you enter a string in the infix-filter only those elements containing this string are shown. The infix-filter is not case-sensitive and it searches for the string entered in the whole name of the element not only at the beginning.

4.3 Reuse of existing elements of a function catalog

You can reuse elements within your function catalog. When you attach an inferior element to your element you often can choose between “Add” and “New”. “Add” usually opens a dialog where you can select an existing element whereas “New” opens a dialog to create a new element of the required type.
For example adding a parameter to a function

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

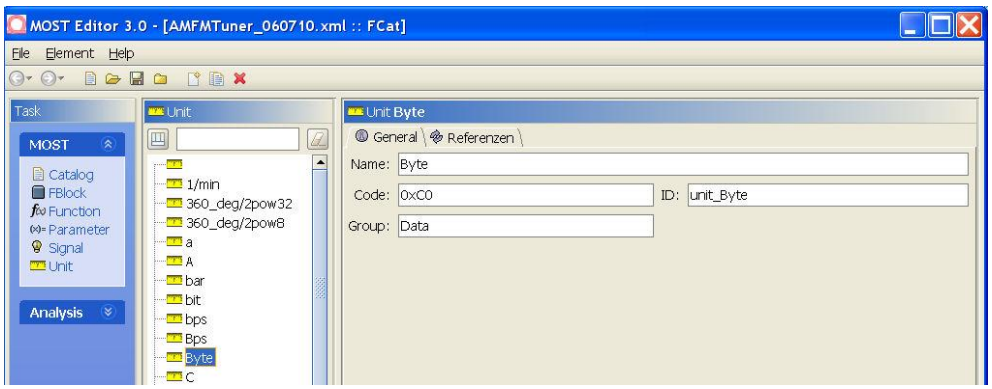


By pressing “New” you attach a newly created parameter to your function. By pressing “Add” you can select one of your already created parameters. Clicking “Remove” only removes the references to the element but not the element itself even if the element has no other references.

4.4 Predefined elements

4.4.1 Units

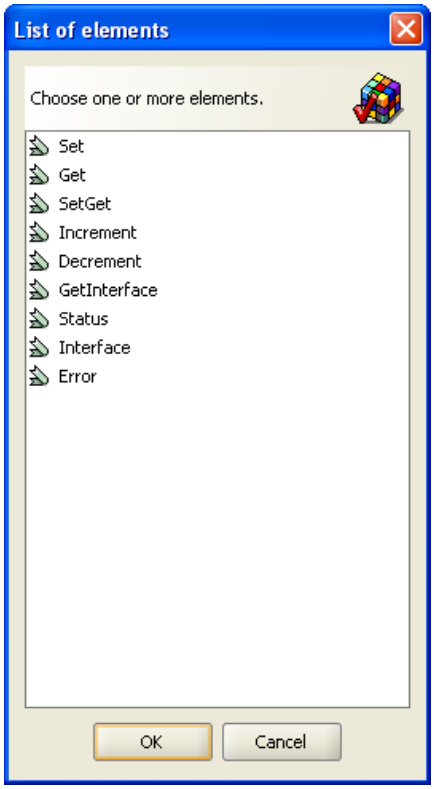
The MOST Editor has a set of predefined units that are available on startup. Nevertheless, you can add your own units.



4.4.2 OPTypes

OPTypes are predefined by the MOST specification. You can choose from the following OPTypes:

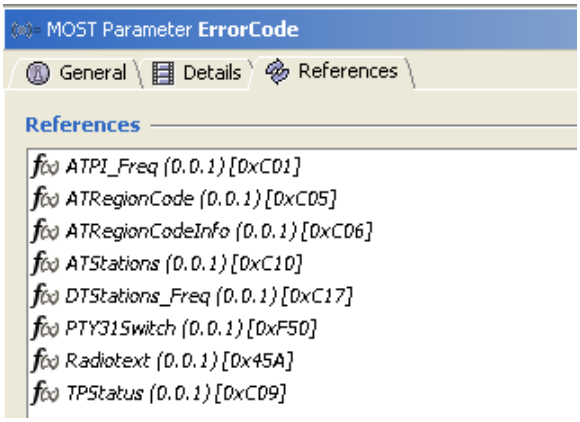
© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.



Note that you can choose several OPTypes at a time. If you want to select more than one element you can use the popular mechanism of pressing “Ctrl” while clicking on the elements you need. If you want to select consecutive elements you can mark the first element you need then press “Shift” then click on the last element you need. All elements between the one marked first and the one marked last are selected.

4.5 References

The “References” tab in the detail editor shows which elements reference the selected element to be able to judge the consequences of changes to this element.

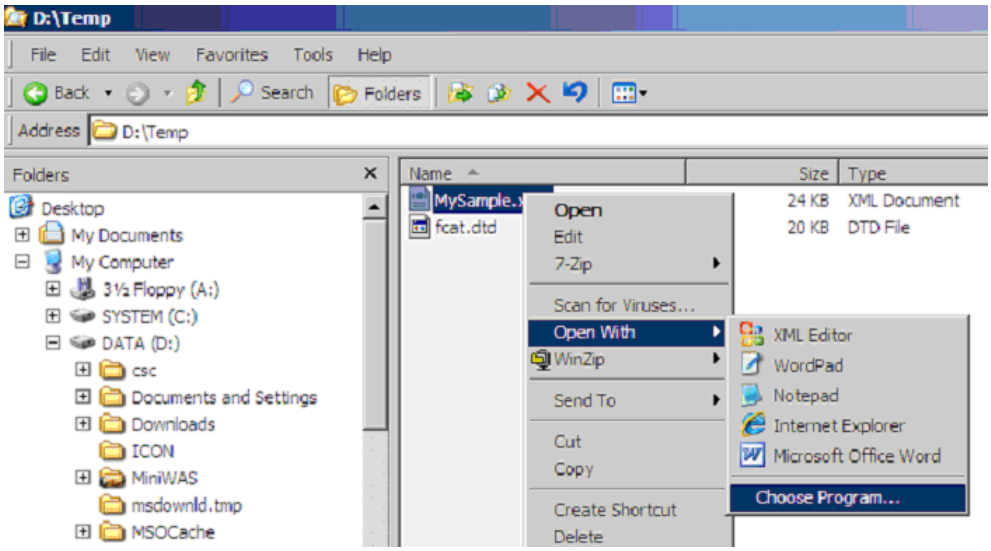


© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

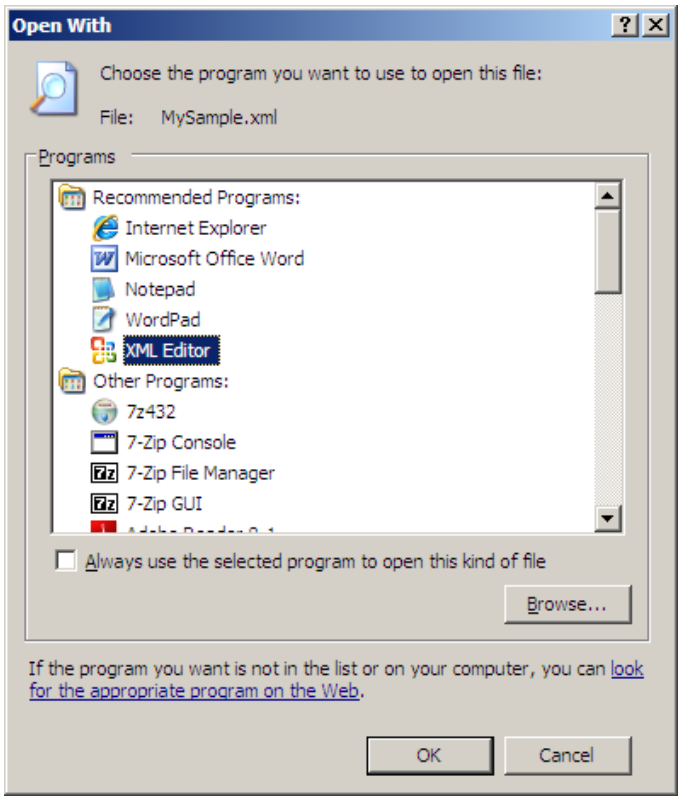
5 How-To / Tips and tricks / FAQ

5.1 How can I incorporate the MOST Editor in the context menu of my explorer?

Open your windows explorer and move to a directory that contains MOST xml files.

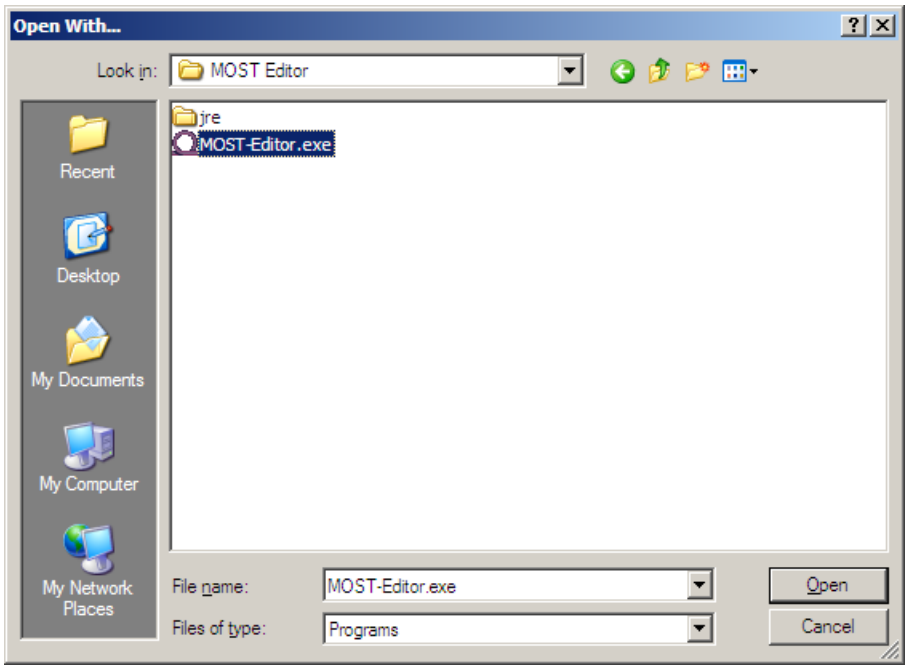


Right-click on your MOST xml file. If the MOST Editor is not available in your context menu select “Choose Program”.

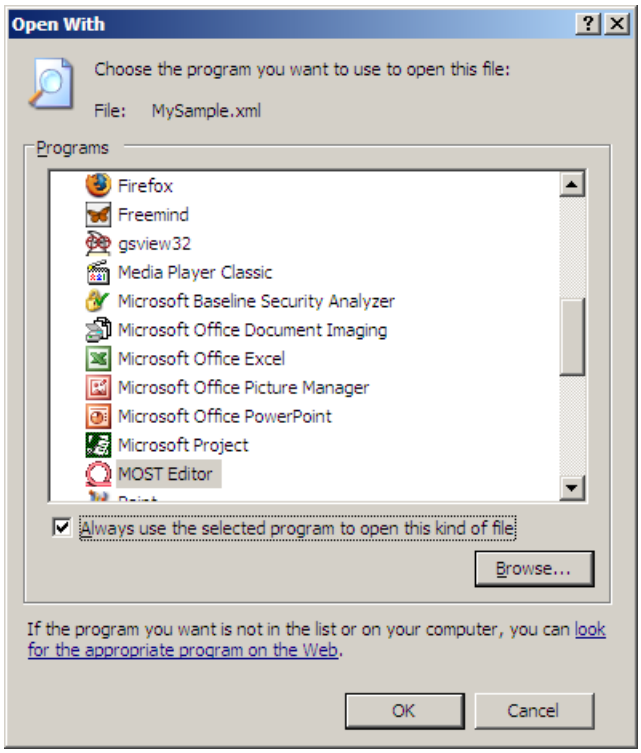


© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

If the MOST Editor is not in your program list you have to search for it in your file system. The standard location for the MOST Editor is C:\Program Files\MOST Cooperation\MOST Editor\MOST-Editor.exe.



Select “MOST-Editor.exe” and click “Open”. The MOST Editor is now selectable in your programs list.

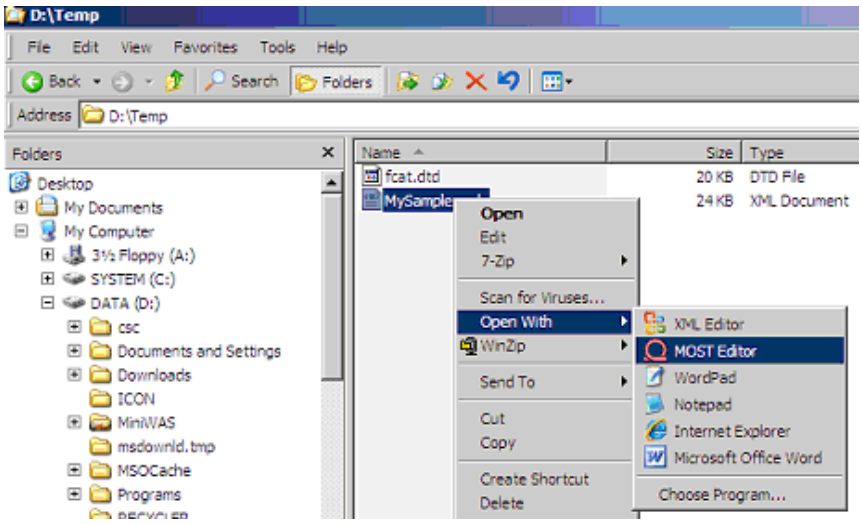


Select the MOST Editor and click “OK”. Now the MOST Editor can be used from the context menu item open within the Window Explorer.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

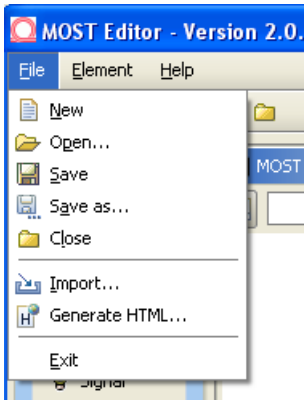
Note: If you check the box “Always use the selected program to open this kind of file” the MOST Editor will automatically start when you double-click an xml file. If you don’t want this behavior uncheck this box.

Now you can use the MOST Editor from your context menu:



5.2 Can I use shortcuts?

You can use shortcuts in the main menu. If you press Alt + the underlined character in the main menu this menu item is opened.



When the menu item is open you only need to press the underlined character.
For example by pressing Alt + f and then pressing n you create a new function catalog. Furthermore, you can navigate in the submenu with your up and down cursor. For example you choose Alt + e which opens the “Element” menu item in the main menu then you use the cursor down to select “Delete” and press “Enter”.
For several elements there are special shotcuts:

Ctrl + N	Creates a new element of the same type
Ctrl + C	Copies an element
Ctrl + X	Deletes an element
Ctrl + Z	Undo (use on description fields only)
Ctrl + Y	Redo (use on description fields only)

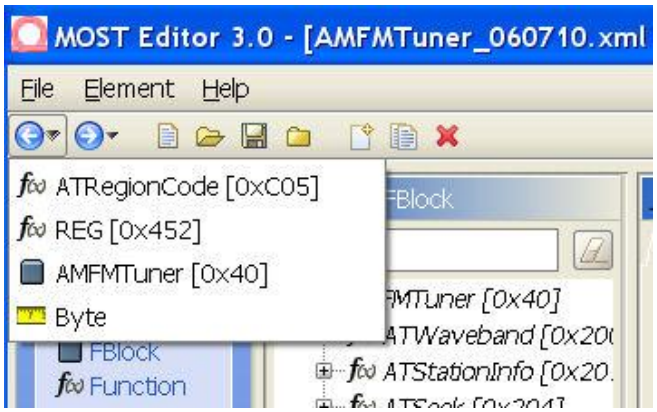
© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.

In the task tree no shortcuts are available. You have to use your mouse there.

© Copyright 1999 – 2018 MOST Cooperation. All rights reserved.

5.3 How can I quickly access previously edited elements?

The navigation elements in the toolbar allow you to easily navigate through the elements you have already edited.



© Copyright 1999 - 2018 MOST Cooperation. All rights reserved.