

MOST

Media Oriented Systems Transport

Multimedia and Control
Networking Technology

Fibex4MOST Cookbook

Rev 1.0

01/2009

MOSTCO CONFIDENTIAL

See page 3 for the terms of disclosure



Legal Notice

COPYRIGHT

© Copyright 1999 - 2009 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
D-76185 Karlsruhe
Germany

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

Fax: (+49) (0) 721 966 50 01

E-mail: contact@mostcooperation.com

Web: www.mostcooperation.com



This Specification is Confidential Information of the MOST Cooperation. It may only be disclosed to member companies. Member companies wishing to discuss these Specifications with suppliers or other third parties must ensure that a commercially standard form of non-disclosure agreement has been previously executed by the party receiving such Specifications. Use of these Specifications may only be for purposes for which they are intended by the MOST Cooperation. Unauthorized use or disclosure is a violation of law.

© Copyright 1999 - 2009 MOST Cooperation.
All rights reserved

MOST is a registered trademark

Contents

1	REFERENCES	7
2	INTRODUCTION	8
3	GENERAL REMARKS	9
3.1	Fibex 3.0.....	9
3.2	Formatting.....	9
3.2.1	Text.....	9
3.2.2	Examples.....	9
3.2.3	Numbers.....	10
3.2.4	Dates.....	10
3.3	Optional Fibex Elements.....	11
3.4	Description of XML paths.....	11
3.5	References.....	11
3.6	Tables.....	12
4	FIBEX4MOST ELEMENTS	13
4.1	PROJECT – General Information.....	13
4.2	CLUSTER – MOST Cluster.....	16
4.3	CHANNEL – Communications Channel.....	17
4.4	ECU – Device.....	18
4.5	COMPOSITE.....	19
4.5.1	SW-PACKAGE – Software Module.....	19
4.5.2	INSTANCE – Instance.....	21
4.5.3	FBLOCK – FBlock.....	22
4.5.4	FUNCTION-GROUP (currently not used for MOST).....	23
4.6	FUNCTION – MOST Function.....	24
4.6.1	Function Classes of Category “Property”.....	26
4.6.1.1	Function Class Switch.....	26
4.6.1.2	Function Class Number.....	26
4.6.1.3	Function Class Text.....	26
4.6.1.4	Function Class Enumeration.....	26
4.6.1.5	Function Class BoolField.....	26
4.6.1.6	Function Class BitSet.....	26
4.6.1.7	Function Class Container.....	27
4.6.1.8	Function Class Sequence Property.....	27
4.6.1.9	Function Class Record.....	27
4.6.1.10	Function Class Array.....	27
4.6.1.11	Function Class DynamicArray.....	27
4.6.1.12	Function Class LongArray.....	27
4.6.1.13	Function Class ArrayWindow.....	28
4.6.1.14	Function Class Map.....	28
4.6.1.15	Function Class Unclassified Property.....	28
4.6.2	Function Classes of Category “Method”.....	28
4.6.2.1	Function Class Sequence Method.....	28
4.6.2.2	Function Class Trigger Method.....	28
4.6.2.3	Function Class Unclassified Method.....	29
4.7	PDU – OType.....	29
4.8	SIGNAL/CODING – MOST Parameter, Data Structures.....	31
4.8.1	MOST Data Types.....	33
4.8.1.1	Boolean.....	33
4.8.1.2	BitField.....	35
4.8.1.3	Enum.....	37
4.8.1.4	(Un)Signed Byte/Word/Long.....	40
4.8.1.5	String.....	47
4.8.1.6	Stream.....	48
4.8.1.7	Classified Stream.....	57
4.8.1.8	Short Stream.....	59

4.8.2	Data Structures.....	60
4.8.2.1	Data Structures Record and Record with Array	60
4.8.2.2	Data Structure Array and Array of Record	64
4.8.2.3	Data Structure DynamicArray	67
4.8.2.4	Data Structure LongArray	69
4.8.2.5	Data Structure Map.....	69
4.8.3	Data Structures with ACCESS-RESTRICTION-SIGNAL-TYPE	70
4.8.3.1	ACCESS-RESTRICTION-SIGNAL-TYPE.....	70
4.8.3.2	Data Structures Record and Record with Array	71
4.8.3.3	Data Structure Array and Array of Record	74
4.8.3.4	Data Structure DynamicArray	78
4.8.3.5	Data Structure LongArray	80
4.8.3.6	Data Structure Map.....	80
5	UNIT-SPEC - DECLARATION OF UNITS.....	81
6	APPENDIX A: INDEX OF FIGURES	84
7	APPENDIX B: INDEX OF TABLES.....	85

Document History

Change Ref.	Section	Changes
1V0_001	General	Initial Version.

1 References

All documents, which are referenced by this MOST document, are listed here along with their versions.

	Document	Revision
[1]	MOST Specification	2.5
[2]	ASAM FIBEX Specification	2.0.1
[3]	ASAM FIBEX Specification	3.0.0

Table 1-1: Document references

2 Introduction

The Fibex format (Field bus Exchange) is an exchange format in in-car networked system development. Originally developed for the FlexRay bus, Fibex now supports all common bus systems in the domain (CAN, FlexRay, LIN, and MOST).

In June 2002 BMW started an initiative for the development of this new exchange format. In June 2003 BMW requested the standardization of the format by ASAM e.V. After the agreement between MOST Cooperation and ASAM e. V., in the 2nd major release Fibex 2.0 in July 2006 through publication of Fibex4MOST the MOST protocol can be modeled in Fibex.

3 General Remarks

Fibex4MOST is an extension of the existing Fibex format. The term Fibex4MOST is not exclusively used for the required extensions to Fibex but—in a more general manner—refers to the entire representation of the MOST protocol within the Fibex structure. To achieve this, elements of Fibex, Fibex4MOST and the Harmonized Data Objects are used.

This document neither replaces the MOST Specification [1] nor the Fibex Specification [2];[3]. The purpose of this document is to describe the concrete contents of an XML export of the Fibex4MOST format.

It is assumed that the reader is familiar with the aforementioned specifications.

This document is organized in a way that matches the top down topology of the described MOST elements.

3.1 Fibex 3.0

Fibex 3.0 (since Q1/2008) [3] and the corresponding Fibex4MOST portion are not identical to Fibex 2.0.x [2]. In this document, the differences between these versions will be pointed out.

The main difference is the use of the PDU object as a replacement of the FRAME object.

Thus, every occurrence of a FRAME object in Fibex4MOST version 2.0.x has to be replaced by a PDU in Fibex4MOST 3.0 and later version.

Furthermore, parameter representation was partially changed. This document relies on the Fibex 3.0 representation. In particular, this applies to parameters belonging to the function classes Switch, BitSet, and BoolField.

3.2 Formatting

3.2.1 Text

Text in this document is formatted based on the nature of the content. This shall make it easier to distinguish the meaning, especially in instructions for providing content, and improve readability.

General text: Arial

XML Code: Courier New

<XML-Code placeholder>: Arial in angle brackets

3.2.2 Examples

The XML examples in this document (boxes with gray background) show concrete and definite extracts from Fibex4MOST documents.

How to fill them is only shown in an exemplary manner. To improve readability, the content is not complete. In these cases—usually long descriptions or repeating structures—the omissions are marked by ellipses (“...”).

In actual Fibex4MOST documents, the corresponding objects have to be provided correctly and completely.

The following XML example contains the complete CODING object “Coding_Complex”. This CODING object contains `SHORT-NAME` as the only element, which contains `Complex_Parameter` as value.

Example:

```
<fx:CODING ID="Coding_Complex">  
  <ho:SHORT-NAME>Complex_Parameter</ho:SHORT-NAME>  
</fx:CODING>
```

The examples contain the corresponding namespaces. Fibex4MOST documents use the following namespaces:

Namespace	Description
most	Fibex4MOST Namespace
ho	Harmonized Data Objects Namespace (ASAM e. V.)
fx	Fibex Namespace
xsi	XML Schema Namespace
xml	XML Namespace
all	fibex4multiplatform, Gateway Scopes

3.2.3 Numbers

Numbers in Fibex4MOST exports, according to [2],[3], have to be provided in decimal representation. Meta-information may be stored in MANUFACTURER-EXTENSIONS (cf. section 4.8).

3.2.4 Dates

In Fibex4MOST documents, date and time information is always provided in the following format¹:

YYYY-MM-DD"T"HH:MM:SS

Example:

```
2007-08-23T22:50:12
```

If a date is not available, the default is "0000-00-00". If no time is available, the default is "00:00:00".

¹ <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#datetime>

3.3 Optional Fibex Elements

The elements described in this document have to be filled in in any case (assuming the corresponding data exists), independent of the declaration in Fibex as optional elements, unless otherwise stated in this document.

For example, the descriptive elements for `ho:DESC` are in many cases only available in one language. In example 1, a function has English names and descriptions but the German names and descriptions do not exist; the resulting export matches example 2.

Elements that are not optional have to be contained in the Fibex XML format in any case.

Example 1 XML Code:

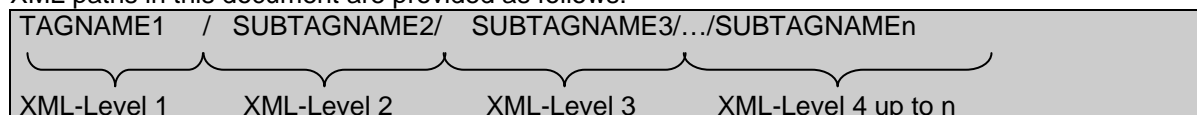
```
<fx:FUNCTION xsi:type="most:FUNCTION-TYPE" ID="_Function_37">
  <ho:SHORT-NAME>FunctionName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">FunctionName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard" />
  ...
</fx:FUNCTION>
```

Example 2 XML Code:

```
<fx:FUNCTION xsi:type="most:FUNCTION-TYPE" ID="_Function_37">
  <ho:SHORT-NAME>FunctionName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">FunctionName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  ...
</fx:FUNCTION>
```

3.4 Description of XML paths

XML paths in this document are provided as follows:



The XML tag TAGNAME1 contains an XML tag with the name SUBTAGNAME2, which in turn contains a tag element SUBTAGNAME3. The path is built depending on the depth n.

3.5 References

References within an exported file (one Fibex4MOST file) have to be unique. The references in multiple exports (multiple Fibex4MOST files) are independent of each other.

In this document, the references are consistent only within individual examples (one block of text with gray background) or within two examples that depend on each other.

In general, references are not consistent over multiple examples.

3.6 Tables

Tables that describe Fibex4MOST XML structures follow the following rules (shown in an example table):

Element	Instructions for Providing Content
FUNCTION	Attribute: type="FUNCTION-TYPE"
FUNCTION/FUNCTION-ID	<FktID>
FUNCTION/FCT-SECTION	Function range depending on the FktID
FUNCTION/FUNCTION-KIND	Property Function kind

The first column contains the XML tag, including the XML path if applicable but without the namespace. The second column contains the instructions on how to provide the content. The instructions provides existing attributes (without giving the namespace) and their values. The content of the element is described, explicitly or in form of a variable. Thus, the above table, results in the following XML lines.

One element `FUNCTION` contains one attribute `type` with the content `FUNCTION-TYPE`. `FUNCTION` contains a sub-element `FUNCTION-ID`. This element contains the value of the `FktID`. In this example, the MOST function has the function ID with the hexadecimal value `0x259` (corresponding to 601 in decimal representation). The content of the next sub-element `FCT-SECTION` is described in the table and filled with the value that corresponds to the range that the `FktID` is in. A `FktID` with the hexadecimal value `0x259` belongs to the `Extension` range. The third sub-element `FUNCTION-KIND` is filled with the value `Property`, which is the category this function belongs to.

This XML example does not show a complete `FUNCTION` object. It shall only help understand the structure of tables and the corresponding XML code.

```
<FUNCTION type="most:FUNCTION-TYPE" >
  ...
  <FUNCTION-ID>601</FUNCTION-ID>
  <FCT-SECTION>Extension</FCT-SECTION>
  <FUNCTION-KIND>Property</FUNCTION-KIND>
  ...
</FUNCTION>
```

4 Fibex4MOST Elements

This chapter describes the structure and concrete values of individual Fibex4MOST elements or Fibex elements whenever these are used.

The section headings consist of the name of the Fibex object in question and the corresponding MOST or Fibex4MOST term, separated by a hyphen. For example, the heading “PDU – OPType” means that the MOST element “OPType” is represented by the Fibex4MOST object “PDU”.

If no unambiguous correspondence exists, then following the hyphen a paraphrase of the meaning in MOST is provided, for example, “PROJECT – General Information”.

Every section starts with a short introduction of the content of the element. In a table, the elements and sub-elements along with instructions for providing the content are listed. The section ends with an example that illustrates the concrete representation. These examples are definite for the representation of MOST elements in a Fibex4MOST XML file.

The Fibex4MOST objects PROJECT, ECU, SW-PACKAGE, FBLOCK, FUNCTION and SIGNAL contain revision information. For all ELEMENT-REVISION the rule applies that only the most current (largest REVISION-LABEL) is contained in the Fibex XML. The TEAM-MEMBER-REF element determines the responsible person for that particular element. Revision information consists of the following XML sub-elements.

Element	Instructions for Providing Content
TEAM-MEMBER-REF	Attribute: ID-REF= "<Identifier of the responsible user> "
REVISION-LABEL	<Version> of the particular element
STATE	May contain the value <i>open</i> or <i>closed</i>
DATE	Contains the date of the last change
MODIFICATION	Only used under DOC-REVISION of the PROJECT object and described there

4.1 PROJECT – General Information

The Fibex element PROJECT contains general information on the Fibex file, for example, revision history, version information, and document generation information.

The PROJECT object contains the following elements:

Element	Instructions for Providing Content
PROJECT	Attribute: ID= "<Identifier of the PROJECT object> "
PROJECT/SHORT-NAME	MOST Characterizes the file as pure MOST function library
PROJECT/LONG-NAME	Attribute: lang= "en " Contains the same value as SHORT-NAME.
PROJECT/ADMIN-DATA/ DOC-REVISIONS/ DOC-REVISION/ TEAM-MEMBER-REF	Attribute: ID-REF= "<Identifier of the user> "
PROJECT/ADMIN-DATA/ DOC-REVISIONS/ DOC-REVISION/DATE	Contains the time stamp of the file (date and time of creation). comp. section 3.2.4
PROJECT/ADMIN-DATA/ DOC-REVISIONS/ DOC-REVISION/ MODIFICATIONS	Contains the document history.
PROJECT/ADMIN-DATA/ DOC-REVISIONS/	Information on document changes.

Element	Instructions for Providing Content
DOC-REVISION/ MODIFICATIONS/ MODIFICATION/CHANGE	
PROJECT/ADMIN-DATA/ DOC-REVISIONS/ DOC-REVISION/ MODIFICATIONS/ MODIFICATION/REASON	Information on reasons for changes.
PROJECT/COMPANIES-DATA/	Contains information on companies and employees involved in the project.
PROJECT/COMPANIES-DATA/ COMPANY-DATA	Attribute: ID= "<identifier of the COMPANY-DATA object> " Attribute: ROLE= "<role of the company> " Role may be: MANUFACTURER SUPPLIER
PROJECT/COMPANIES-DATA/ COMPANY-DATA/SHORT-NAME	<name> of the company
PROJECT/COMPANIES-DATA/ COMPANY-DATA/LONG-NAME	Attribute: lang= "en " Contains the same value as SHORT-NAME.
PROJECT/COMPANIES-DATA/ COMPANY-DATA/DESC	<company address>
PROJECT/COMPANIES-DATA/ COMPANY-DATA/ TEAM-MEMBERS	Contains information on the persons involved in the project.
PROJECT/COMPANIES-DATA/ COMPANY-DATA/ TEAM-MEMBERS/TEAM-MEMBER	Attribute: ID= "<identifier of this TEAM-MEMBER object> "
PROJECT/COMPANIES-DATA/ COMPANY-DATA/ TEAM-MEMBERS/ TEAM-MEMBER/SHORT-NAME	<name> of the employee, or <name1, name2, ...> comma-separated list of names for a group of employees
PROJECT/COMPANIES-DATA/ COMPANY-DATA/ TEAM-MEMBERS/ TEAM-MEMBER/LONG-NAME	Attribute: lang= "en " Contains the same value as SHORT-NAME.
PROJECT/COMPANIES-DATA/ COMPANY-DATA/ TEAM-MEMBERS/ TEAM-MEMBER/DEPARTMENT	<department> the employee belongs to, or <department1, department2, ...> comma-separated list corresponding to the SHORT-NAME (see above).

The following XML example shows a PROJECT object. Tool manufacturers, for example, may store additional information in the PROJECT object.

Example XML Code:

```
<fx:PROJECT ID="project1">
  <ho:SHORT-NAME>MOST</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">MOST</ho:LONG-NAME>
  <ho:ADMIN-DATA>
    <ho:DOC-REVISIONS>
      <ho:DOC-REVISION>
        <ho:TEAM-MEMBER-REF ID-REF="_user_1" />
        <ho:DATE>2007-02-01T22:50:12</ho:DATE>
        <ho:MODIFICATIONS>
          <ho:MODIFICATION>
            <ho:CHANGE>Created on 2007-02-01 at 22:50:12</ho:CHANGE>
            <ho:REASON>initial version</ho:REASON>
          </ho:MODIFICATION>
        </ho:MODIFICATIONS>
      </ho:DOC-REVISION>
    </ho:DOC-REVISIONS>
  </ho:ADMIN-DATA>
  <ho:COMPANIES-DATA>
    <ho:COMPANY-DATA ID="idCompany" ROLE="MANUFACTURER">
      <ho:SHORT-NAME>Company</ho:SHORT-NAME>
      <ho:LONG-NAME xml:lang="en">Company</ho:LONG-NAME>
      <ho:DESC>Company-Adresse</ho:DESC>
      <ho:TEAM-MEMBERS>
        <ho:TEAM-MEMBER ID="_user_1">
          <ho:SHORT-NAME>User1</ho:SHORT-NAME>
          <ho:LONG-NAME xml:lang="en">User1</ho:LONG-NAME>
          <ho:DEPARTMENT>XY</ho:DEPARTMENT>
        </ho:TEAM-MEMBER>
      </ho:TEAM-MEMBERS>
    </ho:COMPANY-DATA>
  </ho:COMPANIES-DATA>
</fx:PROJECT>
```

4.2 CLUSTER – MOST Cluster

The Fibex4MOST CLUSTER object references the channels that are used for MOST.

Element	Instructions for Providing Content
CLUSTER	Attribute: ID=" <Identifier of this cluster> "
CLUSTER/SHORT-NAME	MOSTCluster
CLUSTER/LONG-NAME	Attribute: lang=" en " Contains the same value as SHORT-NAME.
CLUSTER/SPEED	Speed of the MOST cluster 25000000
CLUSTER/ IS-HIGH-LOW-BIT-ORDER	True
CLUSTER/ BIT-COUNTING-POLICY	MONOTONE
CLUSTER/PROTOCOL	MOST
CLUSTER/CHANNEL-REFS/ CHANNEL-REF	Attribute: ID=" <Reference of the channel> " Reference to the channels that are used for MOST. This element is created for every channel.

Example XML Code:

```
<fx:CLUSTER ID='_Cluster_1'>
  <ho:SHORT-NAME>MOSTCluster</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">MOSTCluster</ho:LONG-NAME>
  <fx:SPEED>25000000</fx:SPEED>
  <fx:IS-HIGH-LOW-BIT-ORDER>true</fx:IS-HIGH-LOW-BIT-ORDER>
  <fx:BIT-COUNTING-POLICY>MONOTONE</fx:BIT-COUNTING-POLICY>
  <fx:PROTOCOL>MOST</fx:PROTOCOL>
  <fx:CHANNEL-REFS>
    <fx:CHANNEL-REF ID-REF='_Channel_CC' />
    <fx:CHANNEL-REF ID-REF='_Channel_MOSTHIGH' />
  </fx:CHANNEL-REFS>
</fx:CLUSTER>
```


4.3 CHANNEL – Communications Channel

Caution: This section only applies for Fibex 3.0 and later.

In Fibex4MOST the CHANNEL object is used to specify the communications channel used by individual OPTypes. In particular, CHANNEL refers to the “Control Channel” or the “MOST High Protocol”.

If for an OPTYpe no channel is provided, then the corresponding PDU object will not have a PDU-TRIGGERING.

Element	Instructions for Providing Content
CHANNEL	Attribute: ID= " <identifier of this CHANNEL object> "
CHANNEL/SHORT-NAME	Contains the name of the channel, corresponding to CHANNEL-TYPE (see below)
CHANNEL/LONG-NAME	Attribute: lang= "en" Contains the same value as SHORT-NAME.
CHANNEL/ PDU-TRIGGERINGS/ PDU-TRIGGERING	Attribute: ID= " <identifier of this PDU-TRIGGERING object> " Is created for every OPTYpe that is transported on this channel.
CHANNEL/ PDU-TRIGGERINGS/ PDU-TRIGGERING/ PDU-REF	Attribute: ID-REF= " <identifier of the OPTYpe> "
CHANNEL/CHANNEL-TYPE	Determines the MOST channel. Possible values are: CONTROL MOST_HIGH ETHERNET OTHER

The following XML example contains the CHANNEL object for the “Control Channel”. The references for two OPTypes are explicitly provided. The repetition in the structure is replaced with ellipses (“...”).

Example XML Code:

```
<fx:CHANNEL xsi:type="most:CHANNEL-TYPE" ID="_Channel_1">
  <ho:SHORT-NAME>Control</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Control</ho:LONG-NAME>
  <fx:PDU-TRIGGERINGS>
    <fx:PDU-TRIGGERING ID="_Pdu_Triggering_1">
      <fx:PDU-REF ID-REF="_OP_1" />
    </fx:PDU-TRIGGERING>
    <fx:PDU-TRIGGERING ID="_Pdu_Triggering_2">
      <fx:PDU-REF ID-REF="_OP_2" />
    </fx:PDU-TRIGGERING>
    ...
  </fx:PDU-TRIGGERINGS>
  <most:CHANNEL-TYPE>CONTROL</most:CHANNEL-TYPE>
</fx:CHANNEL>
```

4.4 ECU – Device

The ECU object (Electronic Control Unit) contains all the information that relates to one ECU (device) in the MOST bus. For example, all existing bus connections to the device are specified in this Fibex object. The representation provided here only applies to the pure MOST parts of a Fibex XML file. In the case of a file that covers more busses; additional requirements have to be taken into consideration. Every ECU object contains the following elements:

Element	Instructions for Providing Content
ECU	Attribute: ID=" <identifier of this ECU object > "
ECU/SHORT-NAME	Contains the name of the device
ECU/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
ECU/DESC	Attribute: lang="en" Attribute: TYPE="Standard" English description of the device
ECU/DESC	Attribute: lang="de" Attribute: TYPE="Standard" German description of the device
ECU/ ELEMENT-REVISIONS/ ELEMENT-REVISION	The sub-elements TEAM-MEMBER-REF, REVISION-LABEL, STATE and DATE are used (see chapter 4)
ECU/CONNECTORS	.
ECU/CONNECTORS/CONNECTOR	Attribute: ID" <identifier of this CONNECTOR object> " Created for every CHANNEL object that is used by this device.
ECU/CONNECTORS/ CONNECTOR/CHANNEL-REF	Attribute: ID-REF=" <identifier of a CHANNEL object> "

The following XML example shows an ECU object that references exactly one CHANNEL object.

Example XML Code:

```
<fx:ECU ID="_Device_1">
  <ho:SHORT-NAME>DeviceName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">DeviceName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:TEAM-MEMBER-REF ID-REF="_user_1" />
      <ho:REVISION-LABEL>16</ho:REVISION-LABEL>
      <ho:STATE>open</ho:STATE>
      <ho:DATE>2003-05-08T12:00:00</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CONNECTORS>
    <fx:CONNECTOR ID="_Con_Device_1">
      <fx:CHANNEL-REF ID-REF="_Channel_1" />
    </fx:CONNECTOR>
  </fx:CONNECTORS>
</fx:ECU>
```

4.5 COMPOSITE

The Fibex object COMPOSITE is contained in Fibex4MOST in four types: SW-PACKAGE, INSTANCE, FBLOCK and FUNCTION-GROUP. These are described in the following.

4.5.1 SW-PACKAGE – Software Module

The COMPOSITE object in der type SW-PACKAGE (SW-PACKAGE-TYPE) represents a bundle of FBLOCKS and implemented MOST functions.

SW-PACKAGE objectes reference one or more INSTANCE objects (see 4.5.2), which reference FBLOCKS. In addition, those MOST functions are referenced that are actually implemented.

Every COMPOSITE object of type SW-PACKAGE-TYPE contains the following elements:

Element	Instructions for Providing Content
COMPOSITE	Attribute: type= "SW-PACKAGE-TYPE " Attribute: ID= " <identifier of this SW-PACKAGE object> "
COMPOSITE/SHORT-NAME	Contains the name of the SW-PACKAGE
COMPOSITE/LONG-NAME	Attribute: lang= "en " Contains the same value as SHORT-NAME.
COMPOSITE/DESC	Attribute: lang= "en " Attribute: TYPE= "Standard " English description of the SW-PACKAGE
COMPOSITE/DESC	Attribute: lang= "de " Attribute: TYPE= "Standard " German description of the SW-PACKAGE
COMPOSITE/ ELEMENT-REVISIONS/ ELEMENT-REVISION	The sub-elements TEAM-MEMBER-REF, REVISION-LABEL, STATE and DATE are used (see chapter 4)
COMPOSITE/ECU-REFS/ ECU-REF	Attribute: ID-REF= " <identifier of one device> " Reference to the containing device; created for every containing device.
COMPOSITE/SW-PACKAGE/ IMPLEMENTOR-SERVICE- PORTS/ IMPLEMENTOR-SERVICE- PORT/COMPOSITE-REF	Attribute: ID-REF= " <identifier of an instance> " The SW-PACKAGE references the contained FBlock instances. For every instance one IMPLEMENTOR-SERVICE-PORT is created
COMPOSITE/SW-PACKAGE/ IMPLEMENTOR-SERVICE- PORTS/ IMPLEMENTOR-SERVICE- PORT/FUNCTION-REFS/ FUNCTION-REF	Attribute: ID-REF= " <identifier of a MOST function> " Created for every implemented MOST function of an instance.

The following XML example shows a SW-PACKAGE that references two FBlock instances, which implement two and three MOST functions respectively.

Example XML Code:

```
<fx:COMPOSITE xsi:type="most:SW-PACKAGE-TYPE" ID="_SWPackage_1">
  <ho:SHORT-NAME>SWPackageName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">SWPackageName_long</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:TEAM-MEMBER-REF ID-REF="_user_1" />
      <ho:REVISION-LABEL>10</ho:REVISION-LABEL>
      <ho:STATE>offen</ho:STATE>
      <ho:DATE>2007-02-01T22:50:12</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:ECU-REFS>
    <fx:ECU-REF ID-REF="_Device_1" />
  </fx:ECU-REFS>
  <most:SW-PACKAGE>
    <most:IMPLEMENTOR-SERVICE-PORTS>
      <most:IMPLEMENTOR-SERVICE-PORT>
        <fx:COMPOSITE-REF ID-REF="_Instance_1" />
        <most:FUNCTION-REFS>
          <fx:FUNCTION-REF ID-REF="_Function_1" />
          <fx:FUNCTION-REF ID-REF="_Function_2" />
        </most:FUNCTION-REFS>
      </most:IMPLEMENTOR-SERVICE-PORT>
      <most:IMPLEMENTOR-SERVICE-PORT>
        <fx:COMPOSITE-REF ID-REF="_Instance_2" />
        <most:FUNCTION-REFS>
          <fx:FUNCTION-REF ID-REF="_Function_3" />
          <fx:FUNCTION-REF ID-REF="_Function_4" />
          <fx:FUNCTION-REF ID-REF="_Function_5" />
        </most:FUNCTION-REFS>
      </most:IMPLEMENTOR-SERVICE-PORT>
    </most:IMPLEMENTOR-SERVICE-PORTS>
  </most:SW-PACKAGE>
</fx:COMPOSITE>
```

4.5.2 INSTANCE – Instance

The COMPOSITE object of type INSTANCE (INSTANCE-TYPE) represents the link between an SW-PACKAGE and an FBlock instance (see 4.5.3)

An INSTANCE object contains the instance information and the reference to the FBLOCK object.

An INSTANCE-ID with value“-1” means that the instance ID is not assigned in a static fashion but generated dynamically in the vehicle during run-time.

Every COMPOSITE of type INSTANCE-TYPE contains the following elements:

Element	Instructions for Providing Content
COMPOSITE	Attribute: type=" INSTANCE-TYPE " Attribute: ID=" <identifier of this INSTANCE object> "
COMPOSITE/SHORT-NAME	Instance
COMPOSITE/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
COMPOSITE/ COMPOSITE-REFS/ COMPOSITE-REF	Attribute: ID-REF=" <identifier of an FBlock> " Reference to an FBlock in this instance. Created for every FBlock in this instance.
COMPOSITE/ INSTANCE/ INSTANCE-ID	<InstID>

The following XML example shows an INSTANCE object.

Example XML Code:

```
<fx:COMPOSITE xsi:type="most:INSTANCE-TYPE" ID="_Instance_1">
  <ho:SHORT-NAME>Instance</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Instance</ho:LONG-NAME>
  <fx:COMPOSITE-REFS>
    <fx:COMPOSITE-REF ID-REF="_FBlock_1" />
  </fx:COMPOSITE-REFS>
  <most:INSTANCE>
    <most:INSTANCE-ID>1</most:INSTANCE-ID>
  </most:INSTANCE>
</fx:COMPOSITE>
```

4.5.3 FBLOCK – FBlock

A COMPOSITE object of type FBLOCK (FBLOCK-TYPE) contains all MOST-FBlock-specific information.

The FBLOCK contains references to FUNCTION objects, that is, MOST functions (see 4.6).

Every COMPOSITE object of type FBLOCK-TYPE contains the following elements:

Element	Instructions for Providing Content
COMPOSITE	Attribute: type="FBLOCK-TYPE" Attribute: ID=" <identifier of this FBlock object> "
COMPOSITE/SHORT-NAME	German name of the FBlock
COMPOSITE/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
COMPOSITE/DESC	Attribute: lang="en" Attribute: TYPE="Standard" English description of the FBlock
COMPOSITE/DESC	Attribute: lang="de" Attribute: TYPE="Standard" German description of the FBlock
COMPOSITE/ ELEMENT-REVISIONS/ ELEMENT-REVISION	The sub-elements TEAM-MEMBER-REF, REVISION-LABEL, STATE and DATE are used (see chapter 4)
COMPOSITE/FUNCTION-REFS	References to the functions that are contained in the FBlock.
COMPOSITE/FUNCTION-REFS/ FUNCTION-REF	Attribute: ID-REF=" <identifier of a function> " One entry for every contained function.
COMPOSITE/FBLOCK/ FBLOCK-ID	<FBlockID>

The XML example shows an FBLOCK object. Only the first and the last reference are provided. The recurring structure is replaced with ellipses ("..."). In an actual XML file this would have to be provided completely.

Example XML Code:

```
<fx:COMPOSITE xsi:type="most:FBLOCK-TYPE" ID="_FBlock_1">
  <ho:SHORT-NAME>FBlockName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">FBlockNameAuxIn</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:TEAM-MEMBER-REF ID-REF="_user_1" />
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2006-02-23T12:00:00</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:FUNCTION-REFS>
    <fx:FUNCTION-REF ID-REF="_Function_1" />
    ...
    <fx:FUNCTION-REF ID-REF="_Function_34" />
  </fx:FUNCTION-REFS>
  <most:FBLOCK>
    <most:FBLOCK-ID>36</most:FBLOCK-ID>
  </most:FBLOCK>
</fx:COMPOSITE>
```

4.5.4 FUNCTION-GROUP (currently not used for MOST)

4.6 FUNCTION – MOST Function

MOST functions are assigned to a function class (FUNCTION-CLASS) depending on the type of their parameters (see section 4.8). Function classes are grouped into the categories "Property" and "Method".

MOST functions reference their parameters through OPTypes; these are represented as FRAME objects in Fibex4MOST version 2.0.x and as PDU objects in Fibex 3.0 and later (see section 4.7).

Every FUNCTION object contains the following elements:

Element	Instructions for Providing Content
FUNCTION	Attribute: type="most:FUNCTION-TYPE" Attribute: ID=" <identifier of this FUNCTION object> "
FUNCTION/SHORT-NAME	German name of the function
FUNCTION/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
FUNCTION/DESC	Attribute: lang="en" Attribute: TYPE="Standard" English description of the function
FUCNTION/DESC	Attribute: lang="de" Attribute: TYPE="Standard" German description of the function
FUNCTION/ ELEMENT-REVISIONS/ ELEMENT-REVISION	The sub-elements TEAM-MEMBER-REF, REVISION-LABEL, STATE and DATE are used (see chapter 4)
FUNCTION/FUNCTION-ID	<FktID>
FUNCTION/FCT-SECTION	Depending on the value of FktID, Coordination, Mandatory, Extension, Unique or Proprietary is used
FUNCTION/FUNCTION-KIND	Contains Property or Method, depending on the category of the function class.
FUNCTION/FUNCTION-CLASS	<function class>
FUNCTION/WELLKNOWN	May contain true or false
FUNCTION/NOTIFICATION	May contain true or false
FUNCTION/OCCURRENCE	May contain Mandatory, Conditional or Optional
FUNCTION/VIRTUAL	May contain true or false
FUNCTION/FRAME-REFS	The FRAME-REF sub-elements contain the references to the FRAME objects (OPTypes) (applies to Fibex 2.0)
FUNCTION/FRAME-REFS/ FRAME-REF	Attribute: ID-REF=" <identifier of the referenced FRAME object> "
FUNCTION/PDU-REFS	The PDU-REF sub-elements contain the references to the PDU objects (OPTypes) (Fibex 3.0 and later; see example)
FUNCTION/PDU-REFS/ PDU-REF	Attribute: ID-REF=" <identifier of the referenced PDU object> "

The descriptions of the function classes and their sub-elements follow in sections 4.6.1 and 4.6.2.

The following XML example shows how to fill the FUNCTION object of a MOST function of function class "BoolField", which is of the category "Property" (see below). All FUNCTION objects, independent of category and function class, are constructed correspondingly. The distinction is made in the content of the FUNCTION-KIND and FUNCTION-CLASS elements. Function classes and the specific content of the FUNCTION-KIND and FUNCTION-CLASS elements are described in the following sections (see sections 4.6.1 and 4.6.2). The example follows the rules of Fibex 3.0.

Example XML Code:

```
<fx:FUNCTION xsi:type="most:FUNCTION-TYPE" ID="_Function_12">
  <ho:SHORT-NAME>FunctionName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">FunctionName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:TEAM-MEMBER-REF ID-REF="_user_1" />
      <ho:REVISION-LABEL>2</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <most:FUNCTION-ID>3112</most:FUNCTION-ID>
  <most:FCT-SECTION>Proprietary</most:FCT-SECTION>
  <most:FUNCTION-KIND>Property</most:FUNCTION-KIND>
  <most:FUNCTION-CLASS>BoolField</most:FUNCTION-CLASS>
  <most:WELLKNOWN>false</most:WELLKNOWN>
  <most:NOTIFICATION>true</most:NOTIFICATION>
  <most:OCCURRENCE>Mandatory</most:OCCURRENCE>
  <most:VIRTUAL>false</most:VIRTUAL>
  <most:PDU-REFS>
    <fx:PDU-REF ID-REF="_OP_43" />
    <fx:PDU-REF ID-REF="_OP_44" />
    <fx:PDU-REF ID-REF="_OP_45" />
    <fx:PDU-REF ID-REF="_OP_46" />
    <fx:PDU-REF ID-REF="_OP_3" />
  </most:PDU-REFS>
</fx:FUNCTION>
```

4.6.1 Function Classes of Category “Property”

The meaning and use of function classes is described in [1]. MOST functions of category "Property" are represented in the FUNCTION object as follows.

Element	Instructions for Providing Content
FUNCTION/FUNCTION-KIND	Property

4.6.1.1 Function Class Switch

The function class "Switch" is used for MOST functions which are characterized by exactly one parameter of data type “Boolean” (see 4.8.1.1) for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Switch

4.6.1.2 Function Class Number

The function class "Number" is used for MOST functions which are characterized by exactly one parameter of data type "(Un)Signed Byte/Word/Long" (see section 4.8.1.4) for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Number

4.6.1.3 Function Class Text

The function class "Text" is used for MOST functions which are characterized by exactly one parameter of data type "String" (see section 4.8.1.5) for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Text

4.6.1.4 Function Class Enumeration

The function class "Enumeration" is used for MOST functions which are characterized by exactly one SIGNAL object of data type "Enum" (see section 4.8.1.3) for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Enumeration

4.6.1.5 Function Class BoolField

The function class "BoolField" is used for MOST functions which are characterized by exactly one parameter of data type "bit-wise coded Unsigned Byte/Word/Long" for the OPTypes "Set", "SetGet", and "Status". The data type is described in section 4.8.1.4 "(Un)Signed Byte/Word/Long".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	BoolField

4.6.1.6 Function Class BitSet

Die Funktionsklasse "BitSet" is used for MOST functions which are characterized by exactly one parameter of data type "BitField" (see section 4.8.1.2) for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	BitSet

4.6.1.7 Function Class Container

The function class "Container" is used for MOST functions which are characterized by exactly one parameter of data type "Stream", "Classified Stream" or "ShortStream" (see section 4.8.1.6 et seq.) for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Container

4.6.1.8 Function Class Sequence Property

The function class "Sequence" is used for MOST functions which are characterized by multiple parameters of (potentially) different data type for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Sequence

4.6.1.9 Function Class Record

The function class "Record" is used for MOST functions which are characterized by the parameter "Pos" and exactly one parameter conforming to the "Record" data structure (see section 4.8.3.2) for the OPTypes "Set", "SetGet", and "Status".

Note:

The 2-byte parameter Pos has this structure:

1. Byte: PosX
2. Byte: PosY

This structure is not explicitly modeled in Fibex4MOST.

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Record

4.6.1.10 Function Class Array

The function class "Array" is used for MOST functions which are characterized by one parameter "Pos" and exactly one parameter conforming to the "Array" data structure (see section 4.8.3.3) for the OPTypes "Set", "SetGet", and "Status".

Note:

The 2-byte parameter Pos has this structure:

1. Byte: PosX
2. Byte: PosY

This structure is not explicitly modeled in Fibex4MOST.

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Array

4.6.1.11 Function Class DynamicArray

The function class "DynamicArray" used for MOST functions which are characterized by the parameters "Tag" and "PosY" and exactly one parameter conforming to the "DynamicArray" data structure (see section 4.8.3.4) for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	DynamicArray

4.6.1.12 Function Class LongArray

The function class "LongArray" is described as follows; it uses the parameter conforming to the "LongArray" data structure (see section 4.8.3.5).

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	LongArray

4.6.1.13 Function Class ArrayWindow

Note: Definition of a function class ArrayWindow in Fibex 2.0.1 is currently not possible! This function class becomes available with MOST Specification Rev. 3.0 and Fibex 3.0.

MOST functions of function class "ArrayWindow" are created dynamically during the run-time of the system. One ArrayWindow corresponds to exactly one MOST function of function class "LongArray" and supports in its OPTypes ("Status", optionally "Set" and "SetGet") the parameters "Tag", "PosY", "CurrentSize", "AbsolutePosition" and exactly one parameter conforming to the "LongArray" data structure (see Kap. 4.8.3.5).

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	ArrayWindow

4.6.1.14 Function Class Map

Note: Definition of a function class Map in Fibex 2.0.1 is currently not possible!

The function class "Map" is used for MOST functions which are characterized by the parameters "Tag", "PosY", and exactly one parameter conforming to the "Map" data structure (see section 4.8.3.6) for the OPTypes "Set", "SetGet", and "Status".

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Map

4.6.1.15 Function Class Unclassified Property

The function class "UnclassifiedProperty" is used for MOST functions of category "Property" that cannot be modelled using another function class of this category.

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Unclassified

4.6.2 Function Classes of Category "Method"

The meaning and use of function classes is described in [1]. MOST functions of category "Method" are represented in the FUNCTION object as follows.

Element	Instructions for Providing Content
FUNCTION/FUNCTION-KIND	Method

4.6.2.1 Function Class Sequence Method

The function class "SequenceMethod" is used for MOST functions which are characterized by multiple parameters of arbitrary data type for the OPTypes "Result", "ResultAck", "ErrorAck", etc.

An additional parameter "SenderHandle" is required for "Ack" OPTypes.

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Sequence

4.6.2.2 Function Class Trigger Method

The function class "TriggerMethod" is used for MOST functions which have no parameters for the OPTypes "Result", "Error", etc.

A parameter "SenderHandle" is required for "Ack" OPTypes.

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Trigger

4.6.2.3 Function Class Unclassified Method

The function class "UnclassifiedMethod" is used for MOST functions of category "Method" that cannot be modelled using another function class of this category.

The parameters for the OPTypes of a function that belongs to function class "UnclassifiedMethod" may be based on any data type.

Element	Instructions for Providing Content
FUNCTION/FUNCTION-CLASS	Unclassified

4.7 PDU – OPType

Caution: The following sections are valid only for Fibex version 3.0 and later. In older Fibex versions the FRAME object was used to represent the OPType. The structure of a FRAME object matches that of a PDU object. The keyword FRAME was merely replaced with the keyword PDU. Since Fibex 3.0 the use of the FRAME object is no longer possible.

The order of the parameters within on OPType is defined in the PDU object.

A PDU object for Fibex4MOST consists of the following elements:

Element	Instructions for Providing Content
PDU	Attribute: type="OPTYPE-TYPE" Attribute: ID=" <identifier of this PDU object> "
PDU/SHORT-NAME	Contains the name of the OPTypes, e.g., Get
PDU/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
PDU/BYTE-LENGTH	Contains the value 0 because an OPType does not have a fixed determinable length
PDU/PDU-TYPE	Contains the value OTHER because an OPType cannot be matched to any of the existing PDU types in Fibex.
PDU/ SIGNAL-INSTANCES	Contains the references to the parameters in use. Is only used (including the sub-elements) if the OPType actually references parameters (see example). The sub-elements are created for every referenced parameter.
PDU/SIGNAL-INSTANCES/ SIGNAL-INSTANCE	Attribute: ID=" <identifier of the signal instance> "
PDU/SIGNAL-INSTANCES/ SIGNAL-INSTANCE/ SEQUENCE-NUMBER	Position of the parameter. Der first parameter is assigned the value 0 .
PDU/SIGNAL-INSTANCES/ SIGNAL-INSTANCE/ SIGNAL-REF	Attribute: ID-REF=" <identifier of a parameter> "
PDU/OP-TYPE-ID	<OPType ID> converted to decimal
PDU/OPERATION-KIND	Can be Command or Report

The following XML example shows the content of a FRAME object for an OPType "Get" without parameters. It is valid for the Fibex representation up to Fibex 2.x.

Example XML Code:

```
<fx:FRAME xsi:type="most:OPTYPE-TYPE" ID="_OP_1">
  <ho:SHORT-NAME>Get</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Get</ho:LONG-NAME>
  <fx:BYTE-LENGTH>0</fx:BYTE-LENGTH>
  <fx:FRAME-TYPE>OTHER</fx:FRAME-TYPE>
  <most:OP-TYPE-ID>1</most:OP-TYPE-ID>
  <most:OPERATION-KIND>Command</most:OPERATION-KIND>
</fx:FRAME>
```

The following XML example shows the content of a PDU object for an OPType "Get" without parameters. It is valid for the Fibex representation of Fibex 3.0 and later.

Example XML Code:

```
<fx:PDU xsi:type="most:OPTYPE-TYPE" ID="_OP_1">
  <ho:SHORT-NAME>Get</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Get</ho:LONG-NAME>
  <fx:BYTE-LENGTH>0</fx:BYTE-LENGTH>
  <fx:PDU-TYPE>OTHER</fx:PDU-TYPE>
  <most:OP-TYPE-ID>1</most:OP-TYPE-ID>
  <most:OPERATION-KIND>Command</most:OPERATION-KIND>
</fx:PDU>
```

The following XML example shows the content of a PDU object for an OPType "Status" with three parameter or data structure assignments. It is valid for the Fibex representation of Fibex 3.0 and later.

Example XML Code:

```
<fx:PDU xsi:type="most:OPTYPE-TYPE" ID="_OP_4">
  <ho:SHORT-NAME>Status</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Status</ho:LONG-NAME>
  <fx:BYTE-LENGTH>0</fx:BYTE-LENGTH>
  <fx:PDU-TYPE>OTHER</fx:PDU-TYPE>
  <fx:SIGNAL-INSTANCES>
    <fx:SIGNAL-INSTANCE ID='_SigInst_4'>
      <fx:SEQUENCE-NUMBER>0</fx:SEQUENCE-NUMBER>
      <fx:SIGNAL-REF ID-REF='_Signal_2' />
    </fx:SIGNAL-INSTANCE>
    <fx:SIGNAL-INSTANCE ID='_SigInst_5'>
      <fx:SEQUENCE-NUMBER>1</fx:SEQUENCE-NUMBER>
      <fx:SIGNAL-REF ID-REF='_Signal_3' />
    </fx:SIGNAL-INSTANCE>
    <fx:SIGNAL-INSTANCE ID='_SigInst_9'>
      <fx:SEQUENCE-NUMBER>2</fx:SEQUENCE-NUMBER>
      <fx:SIGNAL-REF ID-REF='_Signal_8' />
    </fx:SIGNAL-INSTANCE>
  </fx:SIGNAL-INSTANCES>
  <most:OP-TYPE-ID>C</most:OP-TYPE-ID>
  <most:OPERATION-KIND>Command</most:OPERATION-KIND>
</fx:PDU>
```

4.8 SIGNAL/CODING – MOST Parameter, Data Structures

The Fibex object SIGNAL with the corresponding Fibex object CODING are used in Fibex4MOST to represent MOST parameters and data structures. The Fibex4MOST representation of MOST data types and MOST data structures is described in the following sections.

In Fibex all numerical values are stored in decimal representation. The Fibex standard does not support any meta information about the numerical representation. Therefore for MOST this meta information will be stored in the MANUFACTURER-EXTENSION of the CODING object. An element META-DATA and an element META-DATA/DISPLAY-ENCODING, both optional, are added. Any mixed representation for a single parameter is not allowed. An example is given in section 4.8.1.3.

The SIGNAL objects consist of the elements in the following tables. Additional instructions for filling the elements and examples are provided in the following sub-sections.

Element	Instructions for Providing Content
SIGNAL	Attribute: type=" <MOST signal type of the SIGNAL object> " Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE=" <MOST data type> " The MOST data type to be used is described in the corresponding section and can be derived from the data type.
SIGNAL/SHORT-NAME	Name of the MOST parameter or data structure, corresponding to the use of the SIGNAL object.
SIGNAL/LONG-NAME	Attribute: lang=" en " Contains the same value as SHORT-NAME.
SIGNAL/DESC	Attribute: lang=" en " Attribute: TYPE=" Standard " English description of the data structure or the parameter
SIGNAL/DESC	Attribute: lang=" de " Attribute: TYPE=" Standard " German description of the data structure or the parameter
SIGNAL/ ELEMENT-REVISIONS/ ELEMENT-REVISION	The sub-elements REVISION-LABEL, STATE and DATE are used (see chapter 4) Caution: SIGNAL objects do not use the element TEAM-MEMBER-REF.
SIGNAL/CODING-REF	Attribute: ID-REF=" <identifier of the CODING object belonging to this SIGNAL object> "

The CODING objects consist of the elements in the following table. Examples are provided in the following sections.

Element	Instructions for Providing Content
CODING	Attribute: ID= "<identifier of this CODING object>"
CODING/SHORT-NAME	Contains the value of SIGNAL/SHORT-NAME of the corresponding SIGNAL object.
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/CODED-TYPE	The use of this element and instructions for filling the elements and sub-elements for MOST data types and data structures examples are provided individually in the corresponding sections.
CODING/COMPU-METHODS	The use of this element and instructions for filling the elements and sub-elements for MOST data types and data structures examples are provided individually in the corresponding sections.
CODING/COMPU-METHODS/ COMPU-METHOD/ INTERNAL-CONSTRS/ SCALE-CONSTR	Attribute: VALIDITY="NOT-VALID" Is created with it sub-elements for gateway-relevant parameters (see example in section 4.8.1.3).
CODING/COMPU-METHODS/ COMPU-METHOD/ INTERNAL-CONSTRS/ SCALE-CONSTR/LOWER-LIMIT	Contains the invalidity value of the gateway parameter.
CODING/COMPU-METHODS/ COMPU-METHOD/ INTERNAL-CONSTRS/ SCALE-CONSTR/UPPER-LIMIT	Contains the invalidity value of the gateway parameter. Content identical with LOWER-LIMIT
CODING/MANUFACTURER- EXTENSION/META- DATA/DISPLAY-ENCODING	Contains the information how a tool shall display the numerical values. Possible values are dec, hex and bin Default: A default value is not defined yet.

4.8.1 MOST Data Types

This section describes the modeling of the following MOST data types in Fibex:

- Boolean
- BitField
- Enum
- (Un)Signed Byte/Word/Long (incl. bit-coded)
- String
- Stream
- Classified Stream
- Short Stream

Examples for filling SIGNAL objects and CODING objects are provided.

4.8.1.1 Boolean

Caution: The representation shown here is only valid for Fibex 3.0 and later.

MOST parameters of MOST data type "Boolean" are declared in Fibex by the use of a SIGNAL- and a CODING object. A parameter of type "Boolean" in its Fibex representation corresponds to a bit-coded Unsigned Byte where only the LSB is used (see section 4.8.1.4).

The SIGNAL is defined by the following attributes:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="SIMPLE-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="Bool "

The CODING is to be filled as follows:

Element	Instructions for Providing Content
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE="A_UINT8" Attribute: CATEGORY="STANDARD-LENGTH-TYPE"
CODING/SHORT-NAME	Name of the Boolean parameter.
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/CODED-TYPE/ BIT-LENGTH	8 Data type Boolean is of length 1 Byte
CODING/COMPU-METHODS/ COMPU-METHOD/SHORT-NAME	Bool Value of the MOST-DATATYPE of the corresponding SIGNAL object.
CODING/COMPU-METHODS/ COMPU-METHOD/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/COMPU-METHODS/ COMPU-METHOD/CATEGORY	TEXTTABLE
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/COMPU-SCALE	The sub-elements of this element contain the coding details. Created for every coding.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: TYPE="code" Contains the coding; all bits apart from the bit in question have to be represented as don't care ("-"). For every coding a corresponding COMPU-SCALE with sub-elements is created.

Element	Instructions for Providing Content
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/ COMPU-CONST/VT	Contains the descriptions of the corresponding coding.

The following XML Example shows the Fibex representation of a parameter of data type "Boolean". The SIGNAL object and the corresponding CODING object are provided.

Example XML Code:

```
<fx:SIGNAL xsi:type="most:SIMPLE-SIGNAL-TYPE" ID="_Signal_14"
MOST-DATATYPE="Bool">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="_Coding_11" />
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_11">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:CODED-TYPE ho:BASE-DATA-TYPE="A_UINT8" CATEGORY="STANDARD-LENGTH-
TYPE">
  <ho:BIT-LENGTH>8</ho:BIT-LENGTH>
</ho:CODED-TYPE>
  <ho:COMPU-METHODS>
  <ho:COMPU-METHOD>
  <ho:SHORT-NAME>Bool</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Bool</ho:LONG-NAME>
  <ho:CATEGORY>TEXTTABLE</ho:CATEGORY>
  <ho:COMPU-INTERNAL-TO-PHYS>
  <ho:COMPU-SCALES>
  <ho:COMPU-SCALE>
  <ho:DESC TYPE="code">-----0</ho:DESC>
  <ho:COMPU-CONST>
  <ho:VT>Description false</ho:VT>
  </ho:COMPU-CONST>
</ho:COMPU-SCALE>
  <ho:COMPU-SCALE>
  <ho:DESC TYPE="code">-----1</ho:DESC>
  <ho:COMPU-CONST>
  <ho:VT>Description true</ho:VT>
  </ho:COMPU-CONST>
</ho:COMPU-SCALE>
</ho:COMPU-SCALES>
</ho:COMPU-INTERNAL-TO-PHYS>
</ho:COMPU-METHOD>
</ho:COMPU-METHODS>
</fx:CODING>
```

4.8.1.2 BitField

Caution: The representation shown here is only valid for Fibex 3.0 and later.

MOST parameters of MOST data type "BitField" are declared in Fibex by the use of a SIGNAL- and a CODING object.

The SIGNAL is defined by the following attributes:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="SIMPLE-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="BitField"

The CODING has to be filled as follows:

Element	Instructions for Providing Content
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE has different values, depending on the data type: A_UINT8 A_UINT16 A_UINT32 Attribute: CATEGORY="STANDARD-LENGTH-TYPE"
CODING/SHORT-NAME	Name of the BitField parameter.
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/CODED-TYPE/ BIT-LENGTH	Filled according to data type – possible values are: 8, 16, and 32
CODING/COMPU-METHODS/ COMPU-METHOD/SHORT-NAME	BitField Value of the MOST-DATATYPE of the corresponding SIGNAL object
CODING/COMPU-METHODS/ COMPU-METHOD/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/COMPU-METHODS/ COMPU-METHOD/CATEGORY	TEXTTABLE
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/COMPU-SCALE	The sub-elements of this element contain the coding details. Created for every coding.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: TYPE="code" Contains the coding; all bits apart from the bit in question have to be represented as don't care ("-"). For every coding a corresponding COMPU-SCALE with sub-elements is created. For a parameter of type BitField half of the data word is available for the coding.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/ COMPU-CONST/VT	Contains the German description of the coding.

The following XML Example shows the Fibex representation of a parameter of data type "BitField". The SIGNAL object and the corresponding CODING object are provided. The CODING object contains the codings for 2 bits.

Example XML Code:

```
<fx:SIGNAL xsi:type="most:SIMPLE-SIGNAL-TYPE" ID="_Signal_42"
MOST-DATATYPE="BitField">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="_Coding_187" />
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_187">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:CODED-TYPE ho:BASE-DATA-TYPE="A_UINT8" CATEGORY="STANDARD-LENGTH-TYPE"
">
  <ho:BIT-LENGTH>8</ho:BIT-LENGTH>
</ho:CODED-TYPE>
  <ho:COMPU-METHODS>
  <ho:COMPU-METHOD>
    <ho:SHORT-NAME>BitField</ho:SHORT-NAME>
    <ho:LONG-NAME xml:lang="en">BitField</ho:LONG-NAME>
    <ho:CATEGORY>TEXTTABLE</ho:CATEGORY>
    <ho:COMPU-INTERNAL-TO-PHYS>
      <ho:COMPU-SCALES>
        <ho:COMPU-SCALE>
          <ho:DESC TYPE="code">---0</ho:DESC>
          <ho:COMPU-CONST>
            <ho:VT>Description 1 false</ho:VT>
          </ho:COMPU-CONST>
        </ho:COMPU-SCALE>
        <ho:COMPU-SCALE>
          <ho:DESC TYPE="code">---1</ho:DESC>
          <ho:COMPU-CONST>
            <ho:VT>Description 1 true</ho:VT>
          </ho:COMPU-CONST>
        </ho:COMPU-SCALE>
        <ho:COMPU-SCALE>
          <ho:DESC TYPE="code">--0-</ho:DESC>
          <ho:COMPU-CONST>
            <ho:VT>Description 2 false</ho:VT>
          </ho:COMPU-CONST>
        </ho:COMPU-SCALE>
        <ho:COMPU-SCALE>
          <ho:DESC TYPE="code">--1-</ho:DESC>
          <ho:COMPU-CONST>
            <ho:VT>Description 2 true</ho:VT>
          </ho:COMPU-CONST>
        </ho:COMPU-SCALE>
      </ho:COMPU-SCALES>
    </ho:COMPU-INTERNAL-TO-PHYS>
  </ho:COMPU-METHOD>
</ho:COMPU-METHODS>
</fx:CODING>
```

4.8.1.3 Enum

MOST parameters of MOST data type "Enum" are declared in Fibex by the use of a SIGNAL object and a CODING object.

The SIGNAL is defined by the following attributes:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="SIMPLE-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="Enum"

The CODING has to be filled as follows:

Element	Instructions for Providing Content
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE="A_UINT8" Attribute: CATEGORY="STANDARD-LENGTH-TYPE"
CODING/CODED-TYPE/ BIT-LENGTH	8
CODING/COMPU-METHODS/ COMPU-METHOD/SHORT-NAME	Enum Value of the MOST-DATATYPE of the corresponding SIGNAL object
CODING/COMPU-METHODS/ COMPU-METHOD/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/COMPU-METHODS/ COMPU-METHOD/CATEGORY	TEXTTABLE
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/COMPU-SCALE	The sub-elements of this elements contain the coding details. Created for for every coding.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: TYPE="code" Contains the coding. Used for <i>binary</i> codings.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: lang="en" Attribute: TYPE="Standard" English <description > of the coding
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: lang="de" Attribute: TYPE="Standard" German <description> of the coding
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/LOWER-LIMIT	Contains the coding. Used for <i>non-binary</i> codings.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/UPPER-LIMIT	Contains the coding. Content identical to element LOWER-LIMIT Used for <i>non-binary</i> codings.

Element	Instructions for Providing Content
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/ COMPU-CONST/VT	Description of the corresponding coding.

The following XML example for decimal coding shows the Fibex representation of a parameter of data type "Enum".

Two codings are explicitly provided. Additional codings are suggested by the use of ellipses ("..."). In an actual Fibex document the codings have to be complete.

Apart from that, in this example the "Enum" is assumed to be a gateway parameter; thus, the invalidity value is provided (see section 4.8).

The example also includes the meta information, that a tool (e.g. MOST Editor) has to display the codings as hexadecimal encoded values. The coding itself is always stored in decimal encoding.

The representation of a binary coding corresponds to the representation of a "bit-coded Unsigned Byte/Word/Long" (see section 4.8.1.4).

Example XML Code:

```
<fx:SIGNAL xsi:type="most:SIMPLE-SIGNAL-TYPE" ID="_Signal_74"
MOST-DATATYPE="Enum" >
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>1</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="_Coding_55" />
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_55">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:CODED-TYPE ho:BASE-DATA-TYPE="A_UINT8" CATEGORY="STANDARD-LENGTH-TYPE" >
    <ho:BIT-LENGTH>8</ho:BIT-LENGTH>
  </ho:CODED-TYPE>
  <ho:COMPU-METHODS>
    <ho:COMPU-METHOD>
      <ho:SHORT-NAME>Enum</ho:SHORT-NAME>
      <ho:CATEGORY>TEXTTABLE</ho:CATEGORY>
      <ho:INTERNAL-CONSTRS>
        <ho:SCALE-CONSTR VALIDITY="NOT-VALID" >
          <ho:LOWER-LIMIT>5</ho:LOWER-LIMIT>
          <ho:UPPER-LIMIT>5</ho:UPPER-LIMIT>
        </ho:SCALE-CONSTR>
      </ho:INTERNAL-CONSTRS>
      <ho:COMPU-INTERNAL-TO-PHYS>
        <ho:COMPU-SCALES>
          <ho:COMPU-SCALE>
```

```
<ho:DESC xml:lang="en" TYPE="Standard">Desc. coding 1</ho:DESC>
<ho:DESC xml:lang="de" TYPE="Standard">Beschr. Codierung 1</ho:DESC>
<ho:LOWER-LIMIT>0</ho:LOWER-LIMIT>
<ho:UPPER-LIMIT>0</ho:UPPER-LIMIT>
<ho:COMPU-CONST>
  <ho:VT>Top</ho:VT>
</ho:COMPU-CONST>
</ho:COMPU-SCALE>
<ho:COMPU-SCALE>
  <ho:DESC xml:lang="en" TYPE="Standard">Desc. coding 2</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschr. Codierung 2</ho:DESC>
  <ho:LOWER-LIMIT>1</ho:LOWER-LIMIT>
  <ho:UPPER-LIMIT>1</ho:UPPER-LIMIT>
  <ho:COMPU-CONST>
    <ho:VT>Bottom</ho:VT>
  </ho:COMPU-CONST>
</ho:COMPU-SCALE>
...
</ho:COMPU-SCALES>
</ho:COMPU-INTERNAL-TO-PHYS>
</ho:COMPU-METHOD>
</ho:COMPU-METHODS>
<fx:MANUFACTURER-EXTENSION>
  <most:META-DATA>
    <most:DISPLAY-ENCODING>hex</most:DISPLAY-ENCODING>
  </most:META-DATA>
</fx:MANUFACTURER-EXTENSION>
</fx:CODING>
```

4.8.1.4 (Un)Signed Byte/Word/Long

MOST parameters of MOST data type "(Un)Signed Byte/Word/Long" are declared in Fibex by the use of a SIGNAL object and a CODING object.

The special case of the MOST data type "bit-coded Unsigned Byte/Word/Long" is described further below.

(Un)Signed Byte/Word/Long

The SIGNAL is defined by the following attributes:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="NUMBER-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="Number "
SIGNAL/NUMBER	Attribute: STEP=" <step width> "

The CODING has to be filled as follows:

Element	Instructions for Providing Content
CODING/SHORT-NAME	Name of the number parameter.
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE has different values, depending on the data type: A_INT8 A_UINT8 A_INT16 A_UINT16 A_INT32 A_UINT32 Attribute: CATEGORY=" STANDARD-LENGTH-TYPE "
CODING/ CODED-TYPE / BIT-LENGTH	Filled according to data type – possible values are: 8, 16, and 32
CODING/COMPU-METHODS/ COMPU-METHOD/SHORT-NAME	Number Value of the MOST-DATATYPE of the corresponding SIGNAL object
CODING/COMPU-METHODS/ COMPU-METHOD/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/COMPU-METHODS/ COMPU-METHOD/CATEGORY	FORMULA
CODING/COMPU-METHODS/ COMPU-METHOD/UNIT-REF	Attribute: ID-REF=" <identifier of a unit> "
CODING/COMPU-METHODS/ COMPU-METHOD/ PHYS-CONSTRS/	This element and its sub-elements are only created if an additional restriction of the value range given in INTERNAL-CONSTR applies. For every restriction, a SCALE-CONSTR is created.
CODING/COMPU-METHODS/ COMPU-METHOD/ PHYS-CONSTRS/ SCALE-CONSTR/LOWER-LIMIT	Lower limit of the value range.
CODING/COMPU-METHODS/ COMPU-METHOD/ PHYS-CONSTRS/ SCALE-CONSTR/UPPER-LIMIT	Upper limit of the value range.

Element	Instructions for Providing Content
CODING/COMPU-METHODS/ COMPU-METHOD/ INTERNAL-CONSTRS/ SCALE-CONSTR/LOWER-LIMIT	Lower limit of the value range.
CODING/COMPU-METHODS/ COMPU-METHOD/ INTERNAL-CONSTRS/ SCALE-CONSTR/UPPER-LIMIT	Upper limit of the value range.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/ COMPU-GENERIC-MATH	10^<Exponent>
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: lang="en" Attribute: TYPE="Standard" English description of the coding
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: lang="de" Attribute: TYPE="Standard" German description of the coding
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/LOWER-LIMIT	Contains the coding.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/UPPER-LIMIT	Contains the coding. Content identical to element LOWER-LIMIT
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/ COMPU-CONST/VT	Description of the corresponding coding.

(Un)Signed Byte/Word/Long with value ranges

The following XML example shows a parameter with `BASE-DATA-TYPE="A_UINT8"`; this means that `BIT-LENGTH` contains the value 8. The step width is contained in the attribute `STEP` of the element `SIGNAL/NUMBER`. The value range in `INTERNAL-CONSTR` is 0 to 250. There is an additional restriction of the value range. This is specified in `PHYS-CONSTR`. Thus, the resulting value range is 0 to 150 and 200 to 250. The values 151 to 199 are not part of the value range.

If there were no additional value range restriction, the element `PHYS-CONSTR` including its sub-elements would not be present.

Example XML Code:

```
<fx:SIGNAL xsi:type="most:NUMBER-SIGNAL-TYPE" ID="_Signal_5"
MOST-DATATYPE="Number" >
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>1</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="_Coding_38" />
  <most:NUMBER STEP="1" />
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_38">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:CODED-TYPE CATEGORY="STANDARD-LENGTH-TYPE" ho:BASE-DATA-TYPE="A_UINT8"
  >
    <ho:BIT-LENGTH>8</ho:BIT-LENGTH>
  </ho:CODED-TYPE>
  <ho:COMPU-METHODS>
    <ho:COMPU-METHOD>
      <ho:SHORT-NAME>Number</ho:SHORT-NAME>
      <ho:LONG-NAME xml:lang="en">Number</ho:LONG-NAME>
      <ho:CATEGORY>FORMULA</ho:CATEGORY>
      <ho:UNIT-REF ID-REF="unit_none" />
      <ho:PHYS-CONSTRS>
        <ho:SCALE-CONSTR>
          <ho:LOWER-LIMIT>0</ho:LOWER-LIMIT>
          <ho:UPPER-LIMIT>100</ho:UPPER-LIMIT>
        </ho:SCALE-CONSTR>
        <ho:SCALE-CONSTR>
          <ho:LOWER-LIMIT>150</ho:LOWER-LIMIT>
          <ho:UPPER-LIMIT>250</ho:UPPER-LIMIT>
        </ho:SCALE-CONSTR>
      </ho:PHYS-CONSTRS>
      <ho:INTERNAL-CONSTRS>
        <ho:SCALE-CONSTR>
          <ho:LOWER-LIMIT>0</ho:LOWER-LIMIT>
          <ho:UPPER-LIMIT>250</ho:UPPER-LIMIT>
        </ho:SCALE-CONSTR>
      </ho:INTERNAL-CONSTRS>
      <ho:COMPU-INTERNAL-TO-PHYS>
      <ho:COMPU-SCALES>
```

```
<ho:COMPU-SCALE>
  <ho:COMPU-GENERIC-MATH>10^1</ho:COMPU-GENERIC-MATH>
</ho:COMPU-SCALE>
<ho:COMPU-SCALE>
  <ho:DESC xml:lang="en" TYPE="Standard">Description 1</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung 1</ho:DESC>
  <ho:LOWER-LIMIT>0</ho:LOWER-LIMIT>
  <ho:UPPER-LIMIT>100</ho:UPPER-LIMIT>
  <ho:COMPU-CONST>
    <ho:VT>Name 1</ho:VT>
  </ho:COMPU-CONST>
</ho:COMPU-SCALE>
<ho:COMPU-SCALE>
  <ho:DESC xml:lang="en" TYPE="Standard">Description 2</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung 2</ho:DESC>
  <ho:LOWER-LIMIT>150</ho:LOWER-LIMIT>
  <ho:UPPER-LIMIT>250</ho:UPPER-LIMIT>
  <ho:COMPU-CONST>
    <ho:VT>Name 2</ho:VT>
  </ho:COMPU-CONST>
</ho:COMPU-SCALE>
</ho:COMPU-SCALES>
</ho:COMPU-INTERNAL-TO-PHYS>
</ho:COMPU-METHOD>
</ho:COMPU-METHODS>
</fx:CODING>
```

Unsigned Word/Long with single codings

The following XML example shows the CODING object of a parameter with single codings.

The SIGNAL object is identical to that in the previous example and therefore omitted here.

Example XML Code (for decimal single codings):

```
<fx:CODING ID="_Coding_38">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <ho:CODED-TYPE CATEGORY="STANDARD-LENGTH-TYPE" ho:BASE-DATA-TYPE="A_UINT16"
">
  <ho:BIT-LENGTH>16</ho:BIT-LENGTH>
</ho:CODED-TYPE>
<ho:COMPU-METHODS>
  <ho:COMPU-METHOD>
    <ho:SHORT-NAME>Number</ho:SHORT-NAME>
    <ho:LONG-NAME xml:lang="en">Number</ho:LONG-NAME>
    <ho:CATEGORY>FORMULA</ho:CATEGORY>
    <ho:UNIT-REF ID-REF="unit_none" />
    <ho:COMPU-INTERNAL-TO-PHYS>
      <ho:COMPU-SCALES>
        <ho:COMPU-SCALE>
          <ho:COMPU-GENERIC-MATH>1</ho:COMPU-GENERIC-MATH>
        </ho:COMPU-SCALE>
        <ho:COMPU-SCALE>
          <ho:DESC xml:lang="en" TYPE="Standard">Description 1</ho:DESC>
          <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung 1</ho:DESC>
          <ho:LOWER-LIMIT>1</ho:LOWER-LIMIT>
          <ho:UPPER-LIMIT>1</ho:UPPER-LIMIT>
          <ho:COMPU-CONST>
            <ho:VT>Name 1</ho:VT>
          </ho:COMPU-CONST>
        </ho:COMPU-SCALE>
        <ho:COMPU-SCALE>
          <ho:DESC xml:lang="en" TYPE="Standard">Description 2</ho:DESC>
          <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung 2</ho:DESC>
          <ho:LOWER-LIMIT>2</ho:LOWER-LIMIT>
          <ho:UPPER-LIMIT>2</ho:UPPER-LIMIT>
          <ho:COMPU-CONST>
            <ho:VT>Name 2</ho:VT>
          </ho:COMPU-CONST>
        </ho:COMPU-SCALE>
      </ho:COMPU-SCALES>
    </ho:COMPU-INTERNAL-TO-PHYS>
  </ho:COMPU-METHOD>
</ho:COMPU-METHODS>
</fx:CODING>
```

Bit-coded Unsigned Byte/Word/Long

Caution: The representation shown here is only valid for Fibex 3.0 and later. Parameters of data type "bit-coded Unsigned Byte/Word/Long" are used by functions of function class "BoolField" (see section 4.6.1.5). The SIMPLE-SIGNAL-TYPE is used.

The SIGNAL is defined as follows:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="SIMPLE-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="BitCoded"

The CODING has to be filled as follows:

Element	Instructions for Providing Content
CODING/SHORT-NAME	Name of the number parameter.
CODING/LONG-NAME	Attribute: lang="en" English name of the number parameter.
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE has different values, depending on the data type: A_UINT8 A_UINT16 A_UINT32 Attribute: CATEGORY="STANDARD-LENGTH-TYPE"
CODING/CODED-TYPE/ BIT-LENGTH	Filled according to data type – possible values are: 8, 16, and 32
CODING/COMPU-METHODS/ COMPU-METHOD/SHORT-NAME	BitCoded Value of the MOST-DATATYPE of the corresponding SIGNAL object
CODING/COMPU-METHODS/ COMPU-METHOD/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/COMPU-METHODS/ COMPU-METHOD/CATEGORY	TEXTTABLE
CODING/COMPU-METHODS/ COMPU-METHOD/UNIT-REF	Attribute: ID-REF=" <reference to the unit> "
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: TYPE="code " Don't care bits are represented by "-"
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: lang="en" Attribute: TYPE="Standard" English description of the coding.
CODING/COMPU-METHODS/ COMPU-METHOD/ COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/DESC	Attribute: lang="de" Attribute: TYPE="Standard" German description of the coding.
CODING/COMPU-METHODS/ COMPU-METHOD/	Name of the corresponding coding.

Element	Instructions for Providing Content
COMPU-INTERNAL-TO-PHYS/ COMPU-SCALES/ COMPU-SCALE/ COMPU-CONST/VT	

The following XML example shows a "bit-coded Unsigned Byte" as it is used as data type of a parameter for a function of function class "BoolField". Every bit is individually used as a boolean variable. In the example, only two codings are provided.

Example XML Code (bit-coded Unsigned Byte/Word/Long):

```
<fx:SIGNAL xsi:type="most:SIMPLE-SIGNAL-TYPE" ID="_Signal_2"
MOST-DATATYPE="BitCoded">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="_Coding_5" />
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_5">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:CODED-TYPE ho:BASE-DATA-TYPE="A_UINT8" CATEGORY="STANDARD-LENGTH-TYPE"
  >
    <ho:BIT-LENGTH>8</ho:BIT-LENGTH>
  </ho:CODED-TYPE>
  <ho:COMPU-METHODS>
    <ho:COMPU-METHOD>
      <ho:SHORT-NAME>BitCoded</ho:SHORT-NAME>
      <ho:LONG-NAME xml:lang="en">BitCoded</ho:LONG-NAME>
      <ho:CATEGORY>TEXTTABLE</ho:CATEGORY>
      <ho:COMPU-INTERNAL-TO-PHYS>
        <ho:COMPU-SCALES>
          <ho:COMPU-SCALE>
            <ho:DESC TYPE="code">-----0</ho:DESC>
            <ho:COMPU-CONST>
              <ho:VT>Description 1 false</ho:VT>
            </ho:COMPU-CONST>
          </ho:COMPU-SCALE>
          <ho:COMPU-SCALE>
            <ho:DESC TYPE="code">-----1</ho:DESC>
            <ho:COMPU-CONST>
              <ho:VT>Description 1 true</ho:VT>
            </ho:COMPU-CONST>
          </ho:COMPU-SCALE>
          <ho:COMPU-SCALE>
            <ho:DESC TYPE="code">-----0-</ho:DESC>
            <ho:COMPU-CONST>
              <ho:VT>Description 2 false</ho:VT>
            </ho:COMPU-CONST>
          </ho:COMPU-SCALE>
        </ho:COMPU-SCALES>
      </ho:COMPU-INTERNAL-TO-PHYS>
    </ho:COMPU-METHOD>
  </ho:COMPU-METHODS>
</fx:CODING>
```

```
<ho:COMPU-SCALE>
  <ho:DESC TYPE="code">-----1</ho:DESC>
  <ho:COMPU-CONST>
    <ho:VT>Description 2 true</ho:VT>
  </ho:COMPU-CONST>
</ho:COMPU-SCALE>
</ho:COMPU-SCALES>
</ho:COMPU-INTERNAL-TO-PHYS>
</ho:COMPU-METHOD>
</ho:COMPU-METHODS>
</fx:CODING>
```

4.8.1.5 String

MOST parameters of MOST data type "String" are declared in Fibex by the use of a SIGNAL object and a CODING object.

The SIGNAL is defined by the following attributes:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="SIMPLE-SIGNAL-TYPE" Attribute: ID="<identifier of this SIGNAL object>" Attribute: MOST-DATATYPE="String"

The CODING has to be filled as follows:

Element	Instructions for Providing Content
CODING/SHORT-NAME	Name of the String parameter.
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE="A_ASCIISTRING" Attribute: CATEGORY="MIN-MAX-LENGTH-TYPE" Attribute: TERMINATION="ZERO"
CODING/CODED-TYPE/ MIN-LENGTH	Minimal length of the string in characters.
CODING/CODED-TYPE/ MAX-LENGTH	Maximum length of the string in characters.

The following XML example shows the Fibex representation of a parameter of data type "String".

In this example, the minimal length of the string is 1 character and the maximum length is 11 characters.

Example XML Code:

```
<fx:SIGNAL xsi:type="most:SIMPLE-SIGNAL-TYPE" ID="_Signal_151"
MOST-DATATYPE="String">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>1</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="_Coding_123" />
```

```
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_123">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:CODED-TYPE BASE-DATA-TYPE="A_ASCIISTRING" CATEGORY="MIN-MAX-LENGTH-
  TYPE" TERMINATION="ZERO">
    <ho:MIN-LENGTH>1</ho:MIN-LENGTH>
    <ho:MAX-LENGTH>11</ho:MAX-LENGTH>
  </ho:CODED-TYPE>
</fx:CODING>
```

4.8.1.6 Stream

Data type "Stream" appears in four different variants, the "Simple Stream", the "Composed Stream", the "List in Stream", and the "MUX Stream". These variants are described here.

Stream

MOST parameters of MOST data type "Stream" are declared in Fibex by the use of a SIGNAL object and a CODING object.

The SIGNAL has to be filled as follows:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="STRUCTURE-SIGNAL-TYPE" Attribute: ID="<identifier of this SIGNAL object>" Attribute: MOST-DATATYPE="Stream"
SIGNAL/STRUCTURE	Is created as empty element for a simple stream.

The CODING has to be filled as follows:

Element	Instructions for Providing Content
CODING/SHORT-NAME	Name of the Stream parameter.
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE="OTHER" Attribute: CATEGORY="MIN-MAX-LENGTH-TYPE"
CODING/CODED-TYPE/ MIN-LENGTH	<Min. number of bytes> If this value is missing, the default value is 0.
CODING/CODED-TYPE/ MAX-LENGTH	<Max. number of bytes > If this value is missing, the default value is the largest possible value (65535).

The following XML example shows the Fibex representation of a simple "Stream".

Example XML Code (simple "Stream" parameter):

```
<fx:SIGNAL xsi:type="most:STRUCTURE-SIGNAL-TYPE" ID="_Signal_43"
MOST-DATATYPE="Stream">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>offen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
```



```

</fx:ELEMENT-REVISION>
</fx:ELEMENT-REVISIONS>
<fx:CODING-REF ID-REF="_Coding_15" />
<most:STRUCTURE/>
</fx:SIGNAL>
    
```

```

<fx:CODING ID="_Coding_15">
<ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
<ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
<ho:CODED-TYPE ho:BASE-DATA-TYPE="OTHER" CATEGORY="MIN-MAX-LENGTH-TYPE">
<ho:MIN-LENGTH>0</ho:MIN-LENGTH>
<ho:MAX-LENGTH>32768</ho:MAX-LENGTH>
</ho:CODED-TYPE>
</fx:CODING>
    
```

Composed Stream

A stream parameter can be composed of multiple different sub-parameters. This construct is called "Composed Stream" in this document (example: ComposedStream = Param1, Param2, ...)

In such a "Composed Stream", a PDU object references exactly one SIGNAL object as stream parameter with the attributes `type="STRUCTURE-SIGNAL-TYPE"` and `MOST-DATATYPE="Stream"`. This stream parameter references the individual sub-parameters.

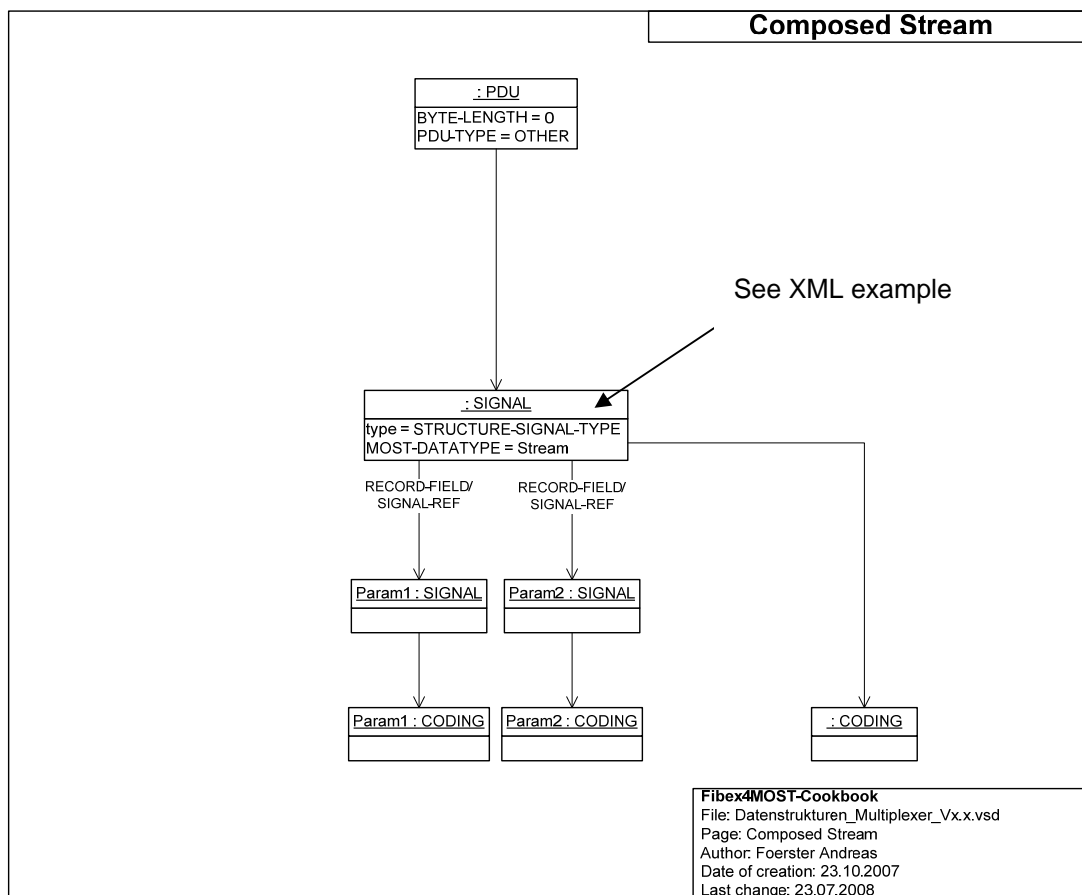


Figure 1: Object diagram of the Fibex model of data type "Stream" as "Composed Stream"

The SIGNAL object with attribute `type="STRUCTURE-SIGNAL-TYPE"` has to be filled as follows:

Element	Instructions for Providing Content
SIGNAL	Attribute: <code>type="STRUCTURE-SIGNAL-TYPE"</code> Attribute: <code>ID="<identifier of this SIGNAL object>"</code> Attribute: <code>MOST-DATATYPE="Stream"</code>
SIGNAL/STRUCTURE/ RECORD-FIELDS/ RECORD-FIELD	Created for every sub-parameter and ist sub-elements.
SIGNAL/STRUCTURE/ RECORD-FIELDS/ RECORD-FIELD/ SEQUENCE-NUMBER	Position of the sub-parameter in the "Composed Stream"
SIGNAL/STRUCTURE/ RECORD-FIELDS/ RECORD-FIELD/ SIGNAL-REF	Attribute: <code>ID-REF="<identifier of the SIGNAL object of the sub-parameter>"</code>

The CODING object has to be filled like a simple "Stream" (see above).

In the following XML example, only the structuring object of type `STRUCTURE-SIGNAL-TYPE`, which is referenced by the PDU object, is shown. The `STRUCTURE-SIGNAL-TYPE` has the `ID="_Signal_2"`. The sub-parameter with the IDs `ID="_Signal_Param1"` and `ID="_Signal_Param2"` are not described any further in this example. Apart from that, the CODING object with the length information of the "Composed Stream" is not shown.

Example XML Code ("Composed Stream" Parameter):

```
<fx:SIGNAL xsi:type="most:STRUCTURE-SIGNAL-TYPE" ID="_Signal_2"
MOST-DATATYPE="Stream">
  <ho:SHORT-NAME>Parameter_Stream_of</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Parameter_Stream_of</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>offen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="Coding_2" />
  <most:STRUCTURE>
    <most:RECORD-FIELDS>
      <most:RECORD-FIELD>
        <fx:SEQUENCE-NUMBER>0</fx:SEQUENCE-NUMBER>
        <fx:SIGNAL-REF ID-REF="_Signal_Param1" />
      </most:RECORD-FIELD>
      <most:RECORD-FIELD>
        <fx:SEQUENCE-NUMBER>1</fx:SEQUENCE-NUMBER>
        <fx:SIGNAL-REF ID-REF="_Signal_Param2" />
      </most:RECORD-FIELD>
    </most:RECORD-FIELDS>
  </most:STRUCTURE>
</fx:SIGNAL>
```

List in Stream

A stream parameter can consist of one or more recurring sub-parameter. This construct is called "List in Stream" in this document. The sub-parameter can be a simple parameter (Stream, Enum, etc.) or a "Composed Stream" (see above). (Example: ListInStream = {Param1, Param2}).

In the context of a "List in Stream", the PDU object references as stream parameter exactly one SIGNAL object with the attributes `type="ARRAY-SIGNAL-TYPE"` and `MOST-DATATYPE="Stream"`. This stream parameter references the recurring sub-parameter. The SIGNAL object of type `ARRAY-SIGNAL-TYPE` represents the repetition of the referenced sub-parameters. In Figure 2, this sub-parameter is a "Composed Stream".

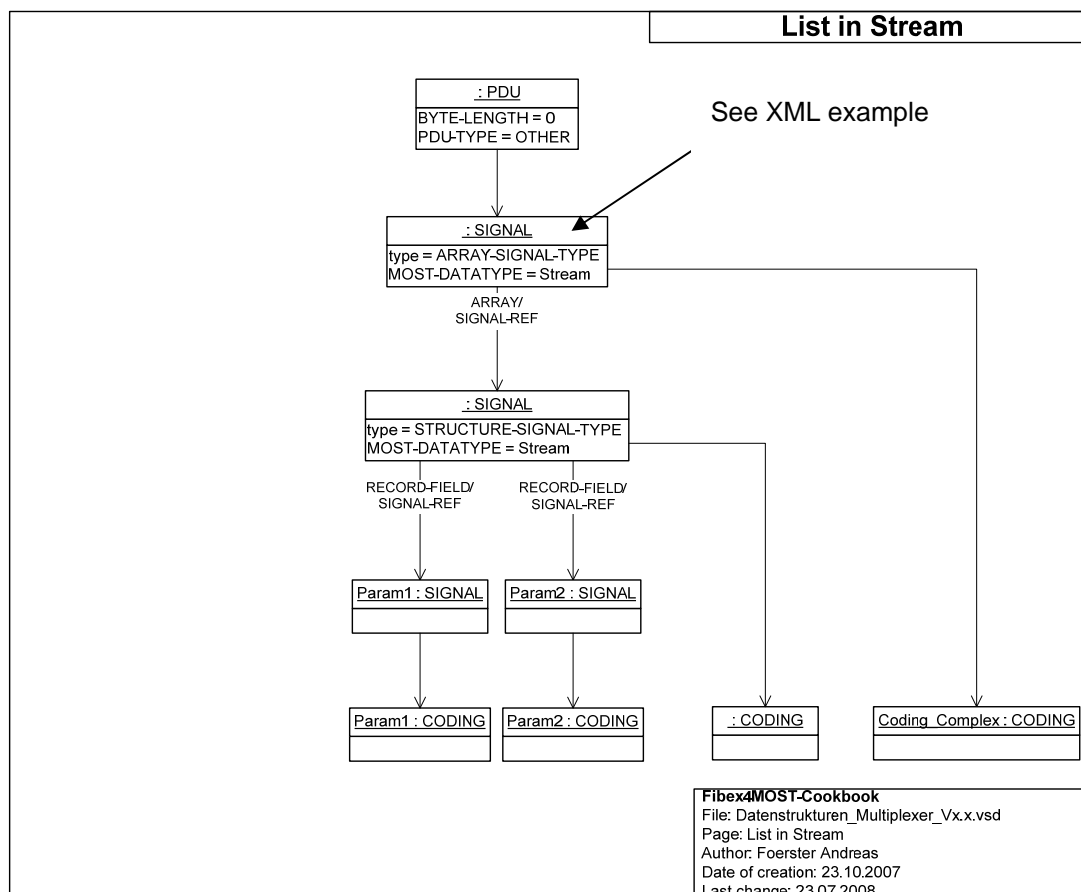


Figure 2: Object diagram of the Fibex model for data type "Stream" as "List in Stream"

The SIGNAL object with the attribute `type="ARRAY-SIGNAL-TYPE"` is only used for structuring the Fibex XML and does not contain any coding information. It therefore references the CODING object "Coding_Complex". The SIGNAL object has to be filled as follows.

Element	Instructions for Providing Content
SIGNAL	Attribute: <code>type="ARRAY-SIGNAL-TYPE"</code> Attribute: <code>ID=" <identifier of this SIGNAL object> "</code> Attribute: <code>MOST-DATATYPE="Stream"</code>
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	Minimum number of repetitions. If this value is missing, the default value is 0.
SIGNAL/ARRAY/ ELEMENT-SIZE/UPPER-LIMIT	Maximum number of repetitions. If this value is missing, the default value is 65535.
SIGNAL/ARRAY/SIGNAL-REF	Attribute: <code>ID-REF=" <identifier of the contained SIGNAL object> "</code>

In the following paragraphs, a "List in Stream" with two recurring parameters is described (see Figure 2). To model the "List in Stream", the PDU object references a SIGNAL object of type ARRAY-SIGNAL-TYPE with MOST-DATATYPE="Stream", which contains the minimum and maximum number of repetitions. The SIGNAL object of type ARRAY-SIGNAL-TYPE in the case of a recurring "Composed Stream" references a SIGNAL of type STRUCTURE-SIGNAL-TYPE with MOST-DATATYPE "Stream". This SIGNAL object of type STRUCTURE-SIGNAL-TYPE references the individual sub-parameters. In the object diagram in Figure 2 and the corresponding XML example of a "List in Stream", these are the parameters Param1 and Param2.

If the stream consists of only one recurring sub-parameter, the sub-parameter is referenced directly by the ARRAY-SIGNAL-TYPE (e.g., "FktIDList").

In the following XML example (matching Figure 1), only the structuring object of type ARRAY-SIGNAL-TYPE, which is referenced by the PDU object, is shown. The structure of the SIGNAL object with attribute type="STRUCTURE-SIGNAL-TYPE" corresponds to that of the "Composed Stream" (see above). The SIGNAL that is referenced by the PDU object, in this case contains ID="_Signal_1". The example does not contain a real description. By the use of ELEMENT-SIZE the minimum number of repetitions of the referenced SIGNAL object with ID="_Signal_2" is set to 1 and the maximum number of repetitions is set to 200. According to Figure 2, the referenced SIGNAL with ID="_Signal_2" is a "Composed Stream".

Example XML Code ("List in Stream" Parameter):

```
<fx:SIGNAL xsi:type="most:ARRAY-SIGNAL-TYPE" ID="_Signal_1"
MOST-DATATYPE="Stream">
  <ho:SHORT-NAME>Data</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Data</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="Coding_Complex" />
  <most:ARRAY>
    <most:ELEMENT-SIZE>
      <most:LOWER-LIMIT>1</most:LOWER-LIMIT>
      <most:UPPER-LIMIT>200</most:UPPER-LIMIT>
    </most:ELEMENT-SIZE>
    <fx:SIGNAL-REF ID-REF="_Signal_2" />
  </most:ARRAY>
</fx:SIGNAL>
```

MUX-Stream

The stream parameter can consist of different sub-parameters, depending on a multiplex-parameter. This construct is called an "MUX-Stream" in this document. The sub-parameter alternatives are called "StreamCases".

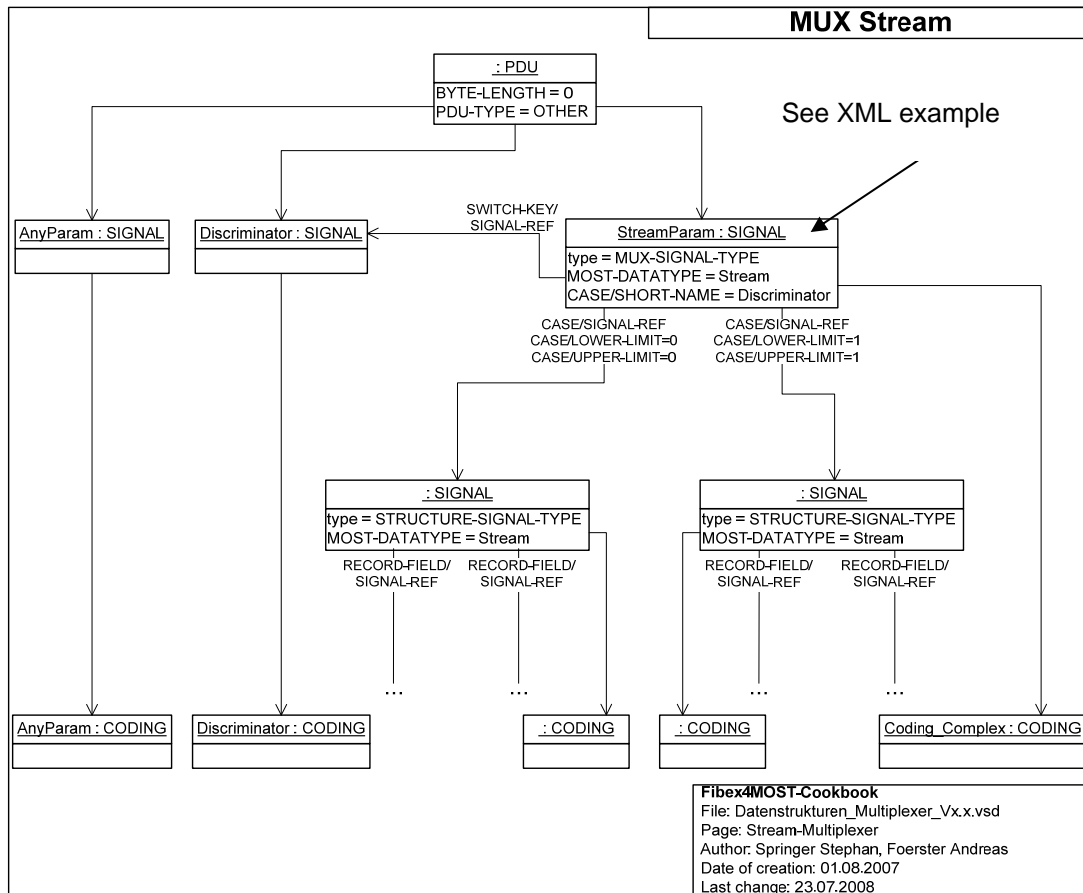


Figure 3: Object diagram of the Fibex model of data type "Stream" as "MUX Stream" with 2 StreamCases

The PDU object references a multiplex-parameter ("Discriminator" in Figure 3) and a SIGNAL object with attributes `type="MUX-SIGNAL-TYPE"` and `MOST-DATATYPE="Stream"` ("StreamParam" in Figure 3). The two "Composed Stream"s in Figure 3 are the possible StreamCases, which were selected depending on the multiplex-parameter.

In the StreamParam, for every StreamCase one CASE-Element is created. Depending on the multiplex-parameter "Discriminator" the corresponding StreamCase is selected.

The SIGNAL object with the attribute `type="MUX-SIGNAL-TYPE"` is only used for structuring the Fibex XML and does not contain any coding information. It references the CODING object "Coding_Complex".

The SIGNAL object of type `MUX-SIGNAL-TYPE` has to be filled as follows:

Element	Instructions for Providing Content
SIGNAL	Attribute: <code>type="MUX-SIGNAL-TYPE"</code> Attribute: <code>ID=" <identifier of this SIGNAL object> "</code> Attribute: <code>MOST-DATATYPE="Stream"</code>
SIGNAL/MUX/POSDISC	Contains the description of the construction of the StreamCases. Combined for all StreamCases.
SIGNAL/MUX/SWITCH-KEY/SIGNAL-REF	Attribute: <code>ID-REF=" <identifier of the Discriminator SIGNAL object> "</code>

Element	Instructions for Providing Content
SIGNAL/MUX/CASES/CASE	Created with ist sub-elements for every StreamCase.
SIGNAL/MUX/CASES/CASE/ SHORT-NAME	<SHORT-NAME> of the Discriminator SIGNAL object
SIGNAL/MUX/CASES/CASE/ LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
SIGNAL/MUX/CASES/CASE/ LOWER-LIMIT	Contains the value of the multiplexer (Discriminator) of the corresponding StreamCase
SIGNAL/MUX/CASES/CASE/ UPPER-LIMIT	Contains the value of the multiplexer (Discriminator) of the corresponding StreamCase
SIGNAL/MUX/CASES/CASE/ SIGNAL-REF	Attribute: ID-REF="<identifier of the SIGNAL object of the sub-parameters>"

In the following XML example, only the structuring object of type MUX-SIGNAL-TYPE, which is referenced by the PDU object, is shown. The multiplexer is referenced through SWITCH-KEY/SIGNAL-REF. Two StreamCases exist for selection.

Example XML Code ("MUX-Stream" Parameter):

```
<fx:SIGNAL xsi:type="most:MUX-SIGNAL-TYPE" ID="_Signal_584"
MOST-DATATYPE="Stream">
  <ho:SHORT-NAME>StreamParam</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">StreamParam</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>offen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="Coding_Complex" />
  <most:MUX>
    <most:SWITCH-KEY>
      <fx:SIGNAL-REF ID-REF="_Signal_583" />
    </most:SWITCH-KEY>
    <most:CASES>
      <most:CASE>
        <ho:SHORT-NAME>DataDescription_1</ho:SHORT-NAME>
        <ho:LONG-NAME xml:lang="en">DataDescription_1</ho:LONG-NAME>
        <most:LOWER-LIMIT>0</most:LOWER-LIMIT>
        <most:UPPER-LIMIT>0</most:UPPER-LIMIT>
        <fx:SIGNAL-REF ID-REF="_Signal_1273" />
      </most:CASE>
      <most:CASE>
        <ho:SHORT-NAME>DataDescription_2</ho:SHORT-NAME>
        <ho:LONG-NAME xml:lang="en">DataDescription_2</ho:LONG-NAME>
        <most:LOWER-LIMIT>1</most:LOWER-LIMIT>
        <most:UPPER-LIMIT>1</most:UPPER-LIMIT>
        <fx:SIGNAL-REF ID-REF="_Signal_1274" />
      </most:CASE>
    </most:CASES>
  </most:MUX>
</fx:SIGNAL>
```

Bit-encoded MUX Stream

A MUX Stream can be constructed based on a bit-coded Discriminator. In this case, the corresponding bit of the Discriminators selects the sub-parameter, which shall be part of the StreamCase. The relative order of sub-parameters based on the SEQUENCE-NUMBER has to be respected.

To distinguish a bit-encoded MUX Stream from a regular MUX Stream, the element POSDESC is filled with the keyword "Bitencoded".

The SIGNAL object of type MUX-SIGNAL-TYPE has to be filled as follows:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="MUX-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="Stream"
SIGNAL/MUX/POSDESC	Bitencoded
SIGNAL/MUX/SWITCH-KEY/ SIGNAL-REF	Attribute: ID-REF=" <identifier of the Discriminator SIGNAL object> "
SIGNAL/MUX/CASES/CASE	Created with ist sub-elements for every StreamCase
SIGNAL/MUX/CASES/CASE/ SHORT-NAME	<SHORT-NAME> of the Discriminator SIGNAL object
SIGNAL/MUX/CASES/CASE/ LOWER-LIMIT	Contains the position of the controlling bit of the multiplexers (Discriminators) of the sub-parameters
SIGNAL/MUX/CASES/CASE/ UPPER-LIMIT	Contains the position of the controlling bit of the multiplexers (Discriminators) of the sub-parameters. Contains identical value as LOWER-LIMIT
SIGNAL/MUX/CASES/CASE/ SIGNAL-REF	Attribute: ID-REF=" <identifier of the SIGNAL object of the sub-parameter> "

StreamSignal

A Stream can be divided into sections, which can have a length that is not byte-aligned, with the help of the "StreamSignal". Such a bit group can have any length; the length unit is bits. A StreamSignal is, like the sub-parameter of a Stream, referenced by the corresponding stream.

Element	Instructions for Providing Content
SIGNAL	Attribute: type="STRUCTURE-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="Stream"
SIGNAL/STRUCTURE	Elements remains empty because a StreamSignal does not have any sub-parameters.

Element	Instructions for Providing Content
CODING	Attribute: type="STRUCTURE-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="Stream"
CODING/SHORT-NAME	Name of the Stream parameter.
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE="OTHER" Attribute: CATEGORY="STANDARD-LENGTH-TYPE"
CODING/CODED-TYPE/ BIT-LENGTH	Contains the length of the StreamSignals in bits

The following XML example shows a Stream of length 2 Byte, which is divided into three StreamSignals of length 3 bits. The SIGNAL object corresponds to that of a streams but the CODING object contains the length in bits.

Example XML Code (divided Stream):

```
<fx:SIGNAL xsi:type="most:STRUCTURE-SIGNAL-TYPE" ID="_Signal_94"
MOST-DATATYPE="Stream">
  <ho:SHORT-NAME>StreamParamWithSignals</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">StreamParamWithSignals</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Stream with 3 signals</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Stream mit 3 Signalen</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="_Coding_60" />
  <most:STRUCTURE>
    <most:RECORD-FIELDS>
      <most:RECORD-FIELD>
        <fx:SEQUENCE-NUMBER>0</fx:SEQUENCE-NUMBER>
        <fx:SIGNAL-REF ID-REF="_Signal_95" />
      </most:RECORD-FIELD>
      <most:RECORD-FIELD>
        <fx:SEQUENCE-NUMBER>1</fx:SEQUENCE-NUMBER>
        <fx:SIGNAL-REF ID-REF="_Signal_95" />
      </most:RECORD-FIELD>
      <most:RECORD-FIELD>
        <fx:SEQUENCE-NUMBER>2</fx:SEQUENCE-NUMBER>
        <fx:SIGNAL-REF ID-REF="_Signal_95" />
      </most:RECORD-FIELD>
    </most:RECORD-FIELDS>
  </most:STRUCTURE>
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_60">
  <ho:SHORT-NAME>Stream</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Stream</ho:LONG-NAME>
  <ho:CODED-TYPE ho:BASE-DATA-TYPE="OTHER" CATEGORY="MIN-MAX-LENGTH-TYPE">
    <ho:MIN-LENGTH>2</ho:MIN-LENGTH>
    <ho:MAX-LENGTH>2</ho:MAX-LENGTH>
  </ho:CODED-TYPE>
</fx:CODING>
```


Example XML Code (StreamSignal):

```
<fx:SIGNAL xsi:type="most:STRUCTURE-SIGNAL-TYPE" ID="_Signal_95" MOST-DATATYPE="Stream">
  <ho:SHORT-NAME>Signal3Bit</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Signal3Bit</ho:LONG-NAME>
  <ho:DESC>A signal with three bits.</ho:DESC>
  <fx:CODING-REF ID-REF="_Coding_61" />
  <most:STRUCTURE />
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_61">
  <ho:SHORT-NAME>StreamSignal</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">StreamSignal</ho:LONG-NAME>
  <ho:CODED-TYPE ho:BASE-DATA-TYPE="OTHER" CATEGORY="STANDARD-LENGTH-TYPE">
    <ho:BIT-LENGTH>3</ho:BIT-LENGTH>
  </ho:CODED-TYPE>
</fx:CODING>
```

4.8.1.7 Classified Stream

MOST parameters of MOST data type “Classified Stream” are declared in Fibex by the use of a SIGNAL object and a CODING object.

The SIGNAL has to be filled as follows:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="CSTREAM-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="ClassifiedStream"
SIGNAL/MEDIA-TYPES/ MEDIA-TYPE	Determines the coding (MIME-Type) of the ClassifiedStream

The CODING has to be filled as follows:

Element	Instructions for Providing Content
CODING/CODED-TYPE	Attribute: BASE-DATA-TYPE="OTHER" Attribute: CATEGORY="MIN-MAX-LENGTH-TYPE"
CODING/SHORT-NAME	Name of the Classified Stream parameter.
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
CODING/CODED-TYPE/ MIN-LENGTH	Minimum length of the ClassifiedStream in bytes. If this value is missing, the default value is 0.
CODING/CODED-TYPE/ MAX-LENGTH	Maximum length of the ClassifiedStreams in bytes. If this value is missing, the default value is 65535.

The SIGNAL object in the following XML example is of type CSTREAM-SIGNAL-TYPE with MOST-DATATYPE="ClassifiedStream". It contains information about the MIME-Type in use.

Example XML Code:

```
<fx:SIGNAL xsi:type="most:CSTREAM-SIGNAL-TYPE" ID="_Signal_99"  
  MOST-DATATYPE="ClassifiedStream">  
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>  
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>  
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>  
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>  
  <fx:ELEMENT-REVISIONS>  
    <fx:ELEMENT-REVISION>  
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>  
      <ho:STATE>offen</ho:STATE>  
    </fx:ELEMENT-REVISION>  
  </fx:ELEMENT-REVISIONS>  
  <fx:CODING-REF ID-REF="_Coding_56" />  
  <most:MEDIA-TYPES>  
    <most:MEDIA-TYPE>text/html</most:MEDIA-TYPE>  
  </most:MEDIA-TYPES>  
</fx:SIGNAL>
```

```
<fx:CODING ID="_Coding_56">  
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>  
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>  
  <ho:CODED-TYPE CATEGORY="STANDARD-LENGTH-TYPE" BASE-DATA-TYPE="OTHER">  
    <ho:MIN-LENGTH>1</ho:MIN-LENGTH>  
    <ho:MAX-LENGTH>11</ho:MAX-LENGTH>  
  </ho:CODED-TYPE>  
</fx:CODING>
```

4.8.1.8 Short Stream

MOST parameters of MOST data type "ShortStream" are declared in Fibex by the use of a SIGNAL object and a CODING object.

The SIGNAL is defined by the following attributes:

Element	Instructions for Providing Content
SIGNAL	Attribute: type="STRUCTURE-SIGNAL-TYPE" Attribute: ID=" <identifier of this SIGNAL object> " Attribute: MOST-DATATYPE="ShortStream"

The content of the CODING corresponds to the CODING of a simple "Stream" (see section 4.8.1.6).
 Caution: The maximum length of a ShortStream is 255 bytes.

The following XML example shows the SIGNAL object of a "ShortStream".

Example XML Code:

```
<fx:SIGNAL xsi:type="most:STRUCTURE-SIGNAL-TYPE" ID="_Signal_105"
MOST-DATATYPE="ShortStream">
  <ho:SHORT-NAME>ParameterName</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">ParameterName</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>1</ho:REVISION-LABEL>
      <ho:STATE>geschlossen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="_Coding_122" />
</fx:SIGNAL>
```

MUX ShortStream

Analogous to Streams, ShortStreams support the use of a multiplexer. The use corresponds to that of the stream multiplexers (see section 4.8.1.6).

StreamSignal ShortStream

Analogous to Streams, ShortStreams support structuring by StreamSignals. The use corresponds to that of the StreamSignals for a regular Stream (see section 4.8.1.6).

4.8.2 Data Structures

This section describes the Fibex4MOST model of the following data structures:

- Record
- Array
- DynamicArray
- LongArray
- Map

The SIGNAL objects with the attributes `type="STRUCTURE-SIGNAL-TYPE"` or `type="ARRAY-SIGNAL-TYPE"` are only used for structuring the Fibex XML and do not contain coding information. They reference the CODING object "Coding_Complex". This CODING is created once, if required, and does not contain any information that is actually used. The "Coding_Complex" is required because every SIGNAL has to reference a CODING. It is built as follows:

Element	Beschreibung
CODING	Attribute: ID="Coding_Complex"
CODING/SHORT-NAME	Complex_Parameter
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.

Example XML Code:

```
<fx:CODING ID="Coding_Complex">
  <ho:SHORT-NAME>Complex_Parameter</ho:SHORT-NAME>
</fx:CODING>
```

Access to single elements of the data structure is not modeled in Fibex. The access mode for individual data structure elements has to be identified by Fibex4MOST tool based on the underlying function class. To describe access restrictions for data structures in Fibex, access modes are explicitly allowed or blocked within `ARRAY-SIGNAL-TYPE` and `STRUCTURE-SIGNAL-TYPE`. The access restrictions are only given in the `ARRAY-SIGNAL-TYPE` or `STRUCTURE-SIGNAL-TYPE`, respectively, which is referenced directly by the PDU object. If access information is not contained for a data structure, the default is that all access modes of the corresponding function class are allowed.

Note: Access restrictions under FIBEX 3.1 are described in section 4.8.3 Data Structures with `ACCESS-RESTRICTION-SIGNAL-TYPE`

4.8.2.1 Data Structures Record and Record with Array

The data structure "Record" appears in two forms in MOST: A simple "Record" and a "Record" which contains an "Array" (called "Record with Array").

To describe data structures of type "Record" and "Record with Array", multiple Fibex structure elements SIGNAL and CODING are required. In following table, the required central SIGNAL object is described.

Element	Instructions for Providing Content
SIGNAL	Attribute: <code>type="STRUCTURE-SIGNAL-TYPE"</code> Attribute: ID=" <identifier of this DS object> " Attribute: <code>MOST-DATATYPE="Record"</code>
SIGNAL/CODING-REF	Attribute: ID-REF="Coding_Complex" This object does not contain a real CODING
SIGNAL/STRUCTURE/ RECORD-FIELDS/ RECORD-FIELD/ SEQUENCE-NUMBER	Position of the referenced SIGNAL in the directly following SIGNAL-REF The position is defined by a number greater or equal 0
SIGNAL/STRUCTURE/	Attribute: ID-REF=" <identifier of a SIGNAL object> "

Element	Instructions for Providing Content
RECORD-FIELDS/ RECORD-FIELD/ SIGNAL-REF	The referenced SIGNAL can describe a MOST parameter or an Array data structure (<code>type="ARRAY-SIGNAL-TYPE"</code>). ("Record with Array")
SIGNAL/ACCESS (currently not supported)	Used with ist sub-elements to describe access restrictions for data structures. Only used in the following data structures: "Record" and "Record with Array"
SIGNAL/ACCESS/COLUMN (currently not supported)	Determines whether individual columns of the data structure may be accessed (<code>true</code>) or not (<code>false</code>).
SIGNAL/ACCESS/ELEMENT (currently not supported)	Determines whether individual elements of the data structure may be accessed (<code>true</code>) or not (<code>false</code>).
SIGNAL/ACCESS/FULL (currently not supported)	Determines whether the data structure may be accessed in its entirety (<code>true</code>) or not (<code>false</code>).
SIGNAL/ACCESS/ROW (currently not supported)	Determines whether individual rows of the data structure may be accessed (<code>true</code>) or not (<code>false</code>).

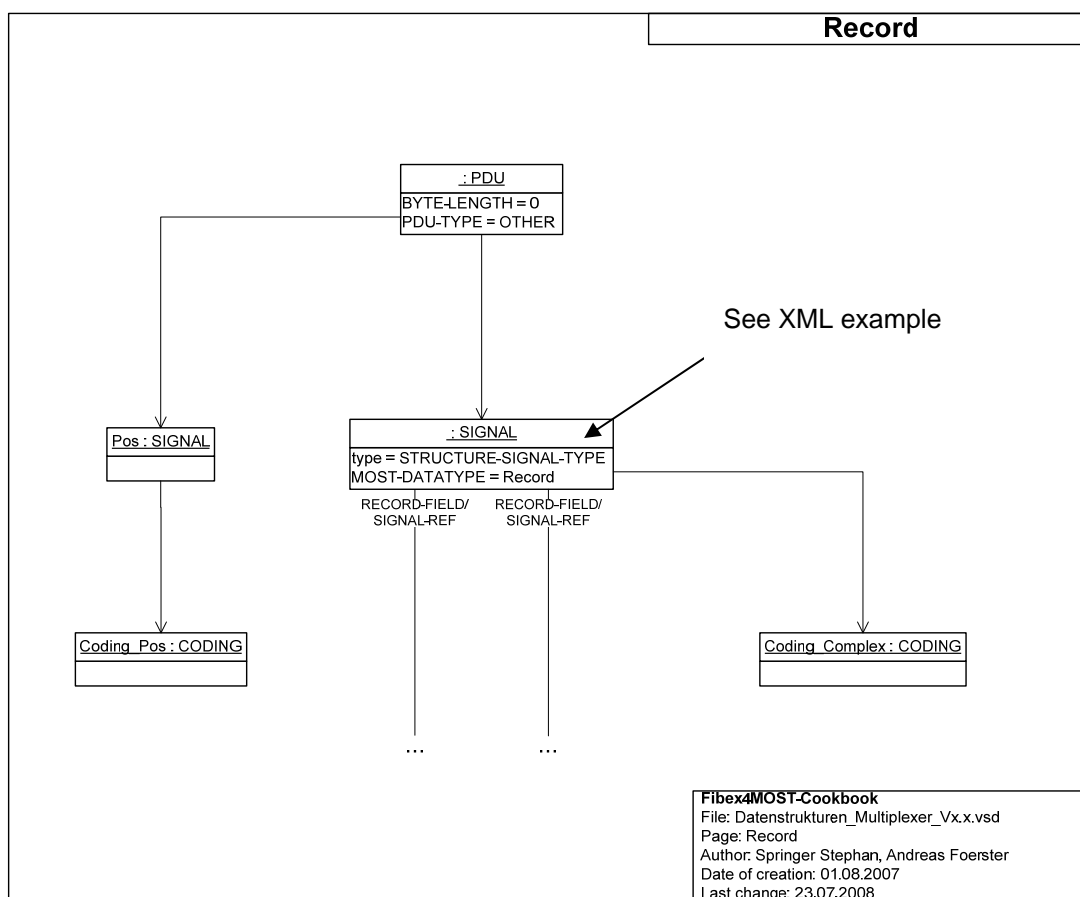


Figure 4: Object diagram of the Fibex model of the simple data structure "Record"

The following XML example shows the use of the SIGNAL object in Figure 9. The use of the other objects is not shown.

Example XML Code („Record“):

```
<fx:SIGNAL xsi:type="most:STRUCTURE-SIGNAL-TYPE" ID="_Signal_46"  
MOST-DATATYPE="Record">  
  <ho:SHORT-NAME>Data</ho:SHORT-NAME>  
  <ho:LONG-NAME xml:lang="en">Data</ho:LONG-NAME>  
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>  
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>  
  <fx:ELEMENT-REVISIONS>  
    <fx:ELEMENT-REVISION>  
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>  
      <ho:STATE>geschlossen</ho:STATE>  
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>  
    </fx:ELEMENT-REVISION>  
  </fx:ELEMENT-REVISIONS>  
  <fx:CODING-REF ID-REF="Coding_Complex" />  
  <most:STRUCTURE>  
    <most:RECORD-FIELDS>  
      <most:RECORD-FIELD>  
        <fx:SEQUENCE-NUMBER>0</fx:SEQUENCE-NUMBER>  
        <fx:SIGNAL-REF ID-REF="_Signal_47" />  
      </most:RECORD-FIELD>  
      <most:RECORD-FIELD>  
        <fx:SEQUENCE-NUMBER>1</fx:SEQUENCE-NUMBER>  
        <fx:SIGNAL-REF ID-REF="_Signal_48" />  
      </most:RECORD-FIELD>  
    </most:RECORD-FIELDS>  
  </most:STRUCTURE>  
</fx:SIGNAL>
```

Record with Array

The "Record" data structure can contain an "Array" as content of the last field. This construct is called a "Record with Array" in this document. The Fibex model of a "Record with Array" is illustrated in Figure 10.

The structure of the SIGNAL object shown in Figure 10 with the attributes `type="STRUCTURE-SIGNAL-TYPE"` and `MOST-DATATYPE="Record"` is identical to that of a "Record" (see above). The structure of the SIGNAL object with the attributes `type="ARRAY-SIGNAL-TYPE"` and `MOST-DATATYPE="Array"` is described in the "Array" section (see 4.8.3.3).

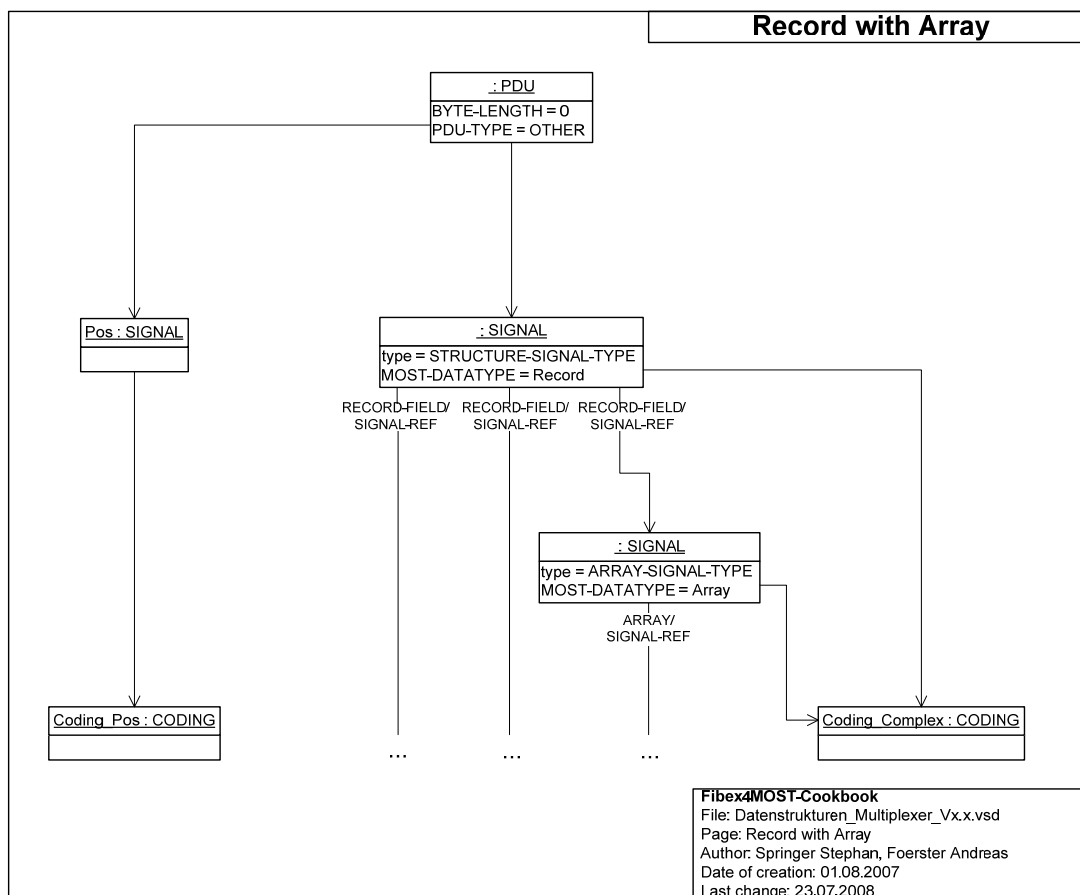


Figure 5: Object diagram of the Fibex model for data structure "Record with Array"

4.8.2.2 Data Structure Array and Array of Record

The data structure "Array" appears in two different forms in MOST: A simple "Array" and an "Array", which contains a "Record" (called "Array of Record").

An "Array" references exactly one sub-parameter. In the context of an "Array of Record", this sub-parameter is a "Record" data structure.

Array

To describe data structures of type "Array", the Fibex structuring elements SIGNAL and CODING are required. The following tables contain the ones that are necessary.

Element	Instructions for Providing Content
SIGNAL	Attribute: type="ARRAY-SIGNAL-TYPE" Attribute: ID=" <identifier of this DS object> " Attribute: MOST-DATATYPE="Array"
SIGNAL/CODING-REF	Attribute: ID-REF="Coding_Complex" This object does not contain a real CODING
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	<minimum length of the Array> Lower length limit of the Array
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	<maximum length of the Array> Upper length limit of the Array (max=255)
SIGNAL/ARRAY/SIGNAL-REF	Attribute: ID-REF=" <identifier of a SIGNAL object> " The referenced SIGNAL can describe a MOST parameter or a data structure "Record" (type="STRUCTURE-SIGNAL-TYPE"). ("Array of Record")
SIGNAL/ACCESS (currently not supported)	Used with ist sub-elements to describe access restrictions for data structures. Only used in the following data structures: "Array", "Array of Record", "DynamicArray", "LongArray", and "Map"
SIGNAL/ACCESS/COLUMN (currently not supported)	Determines whether individual columns of the data structure may be accessed (true) or not (false).
SIGNAL/ACCESS/ELEMENT (currently not supported)	Determines whether individual elements of the data structure may be accessed (true) or not (false).
SIGNAL/ACCESS/FULL (currently not supported)	Determines whether the data structure may be accessed in its entirety (true) or not (false).
SIGNAL/ACCESS/ROW (currently not supported)	Determines whether individual rows of the data structure may be accessed (true) or not (false).

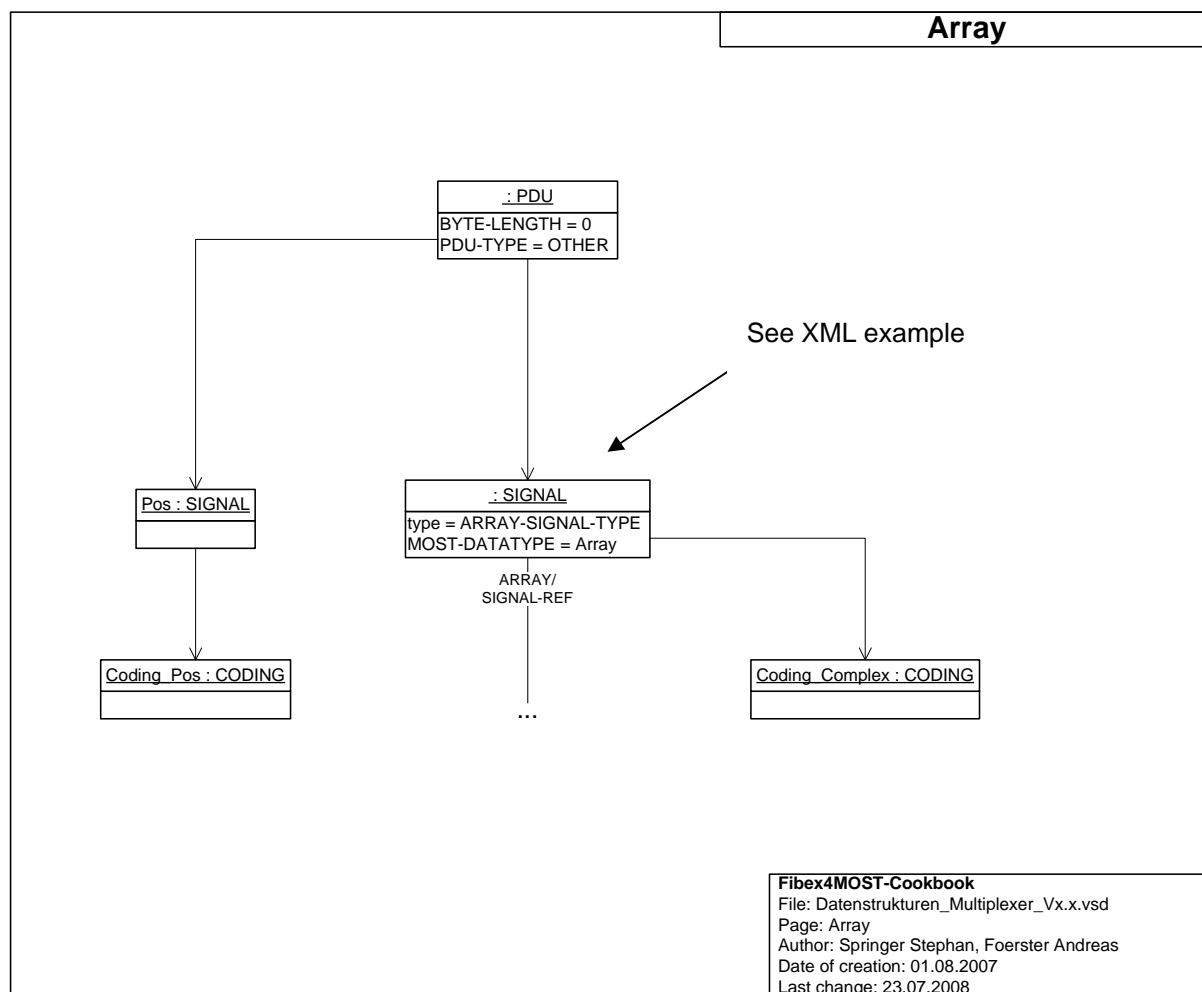


Figure 6: Object diagram of the Fibex model for data structure "Array"

The following table shows, as an example, the length definition of an Arrays and the corresponding representation of the Fibex export:

Length of Array	ELEMENT-SIZE / LOWER-LIMIT	ELEMENT-SIZE / UPPER-LIMIT
>=0	0	Maximum length for data structure
>0	1	Maximum length for data structure
1..10	1	10
25	25	25

The following XML example shows the use of the SIGNAL object that is marked in Figure 11.

Example XML Code („Array“):

```
<fx:SIGNAL xsi:type="most:ARRAY-SIGNAL-TYPE" ID="_Signal_585"
MOST-DATATYPE="Array">
<ho:SHORT-NAME>Data</ho:SHORT-NAME>
<ho:LONG-NAME xml:lang="en">Data</ho:LONG-NAME>
<ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
<ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
<fx:ELEMENT-REVISIONS>
<fx:ELEMENT-REVISION>
```

```

<ho:REVISION-LABEL>3</ho:REVISION-LABEL>
<ho:STATE>offen</ho:STATE>
<ho:DATE>2001-03-12T10:42:42</ho:DATE>
</fx:ELEMENT-REVISION>
</fx:ELEMENT-REVISIONS>
<fx:CODING-REF ID-REF="Coding_Complex" />
<most:ARRAY>
  <most:ELEMENT-SIZE>
    <most:LOWER-LIMIT>255</most:LOWER-LIMIT>
    <most:UPPER-LIMIT>255</most:UPPER-LIMIT>
  </most:ELEMENT-SIZE>
  <fx:SIGNAL-REF ID-REF="_Signal_586" />
</most:ARRAY>
</fx:SIGNAL>
    
```

Array of Record

An "Array" data structure can contain exactly one parameter of type "Record". Such an "Array of Record" is shown in Figure 12. The structure of the SIGNAL object of type ARRAY-SIGNAL-TYPE corresponds to that of a simple "Array" (see above). The structure of the SIGNAL object with the attributes type="STRUCTURE-SIGNAL-TYPE" and MOST-DATATYPE="Record" is described in the "Array" section (see 4.8.3.2).

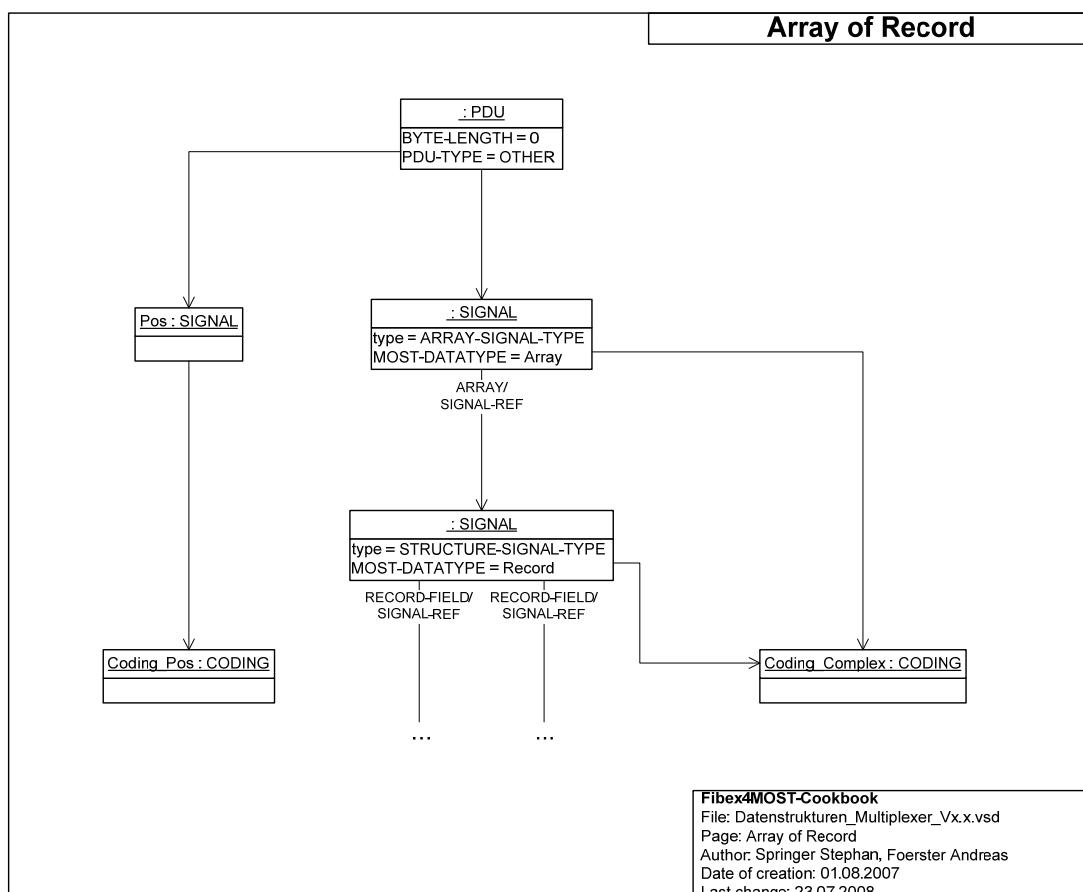


Figure 7: Object diagram of the Fibex model for the data structure "Array of Record"

4.8.2.3 Data Structure DynamicArray

To describe a data structure of type "DynamicArray", multiple Fibex structuring elements SIGNAL and CODING are required. The following tables contain the ones that are necessary.

Element	Instructions for Providing Content
SIGNAL	Attribute: type="ARRAY-SIGNAL-TYPE" Attribute: ID=" <identifier of this DS object> " Attribute: MOST-DATATYPE="DynamicArray"
SIGNAL/CODING-REF	Attribute: ID-REF="Coding_Complex" This object does not contain a real CODING
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	<minimum length of the Array> Lower length limit of the Array
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	<maximum length of the Array> Upper length limit of the Array (max=65535)
SIGNAL/ARRAY/ SIGNAL-REF	Attribute: ID-REF=" <identifier of a SIGNAL object> " The referenced SIGNAL is a data structure "Record" (type="STRUCTURE-SIGNAL-TYPE" and MOST-DATATYPE="Record")
SIGNAL/ACCESS (currently not supported)	Used with ist sub-elements to describe access restrictions for data structures.
SIGNAL/ACCESS/COLUMN (currently not supported)	Determines whether individual columns of the data structure may be accessed (true) or not (false).
SIGNAL/ACCESS/ELEMENT (currently not supported)	Determines whether individual elements of the data structure may be accessed (true) or not (false).
SIGNAL/ACCESS/FULL (currently not supported)	Determines whether the data structure may be accessed in its entirety (true) or not (false).
SIGNAL/ACCESS/ROW (currently not supported)	Determines whether individual rows of the data structure may be accessed (true) or not (false).

Element	Instructions for Providing Content
SIGNAL	Attribute: type="STRUCTURE-SIGNAL-TYPE" Attribute: ID=" <identifier of this DS object> " Attribute: MOST-DATATYPE="Record"
SIGNAL/CODING-REF	Attribute: ID-REF="Coding_Complex" This object does not contain a real CODING

The length definition of the "DynamicArray" is applied in the same way as when dealing with the data structure "Array" (see section 4.8.3.3).

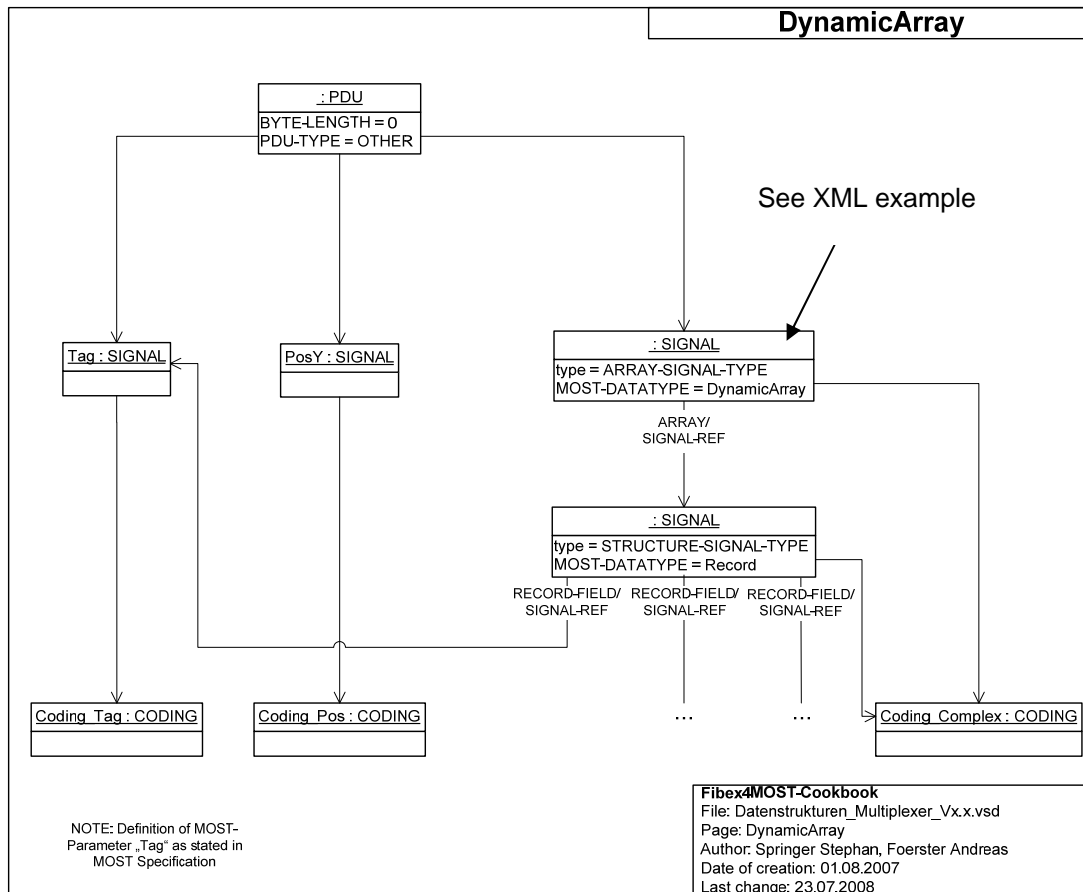


Figure 8: Object diagram of the Fibex model for data structure "DynamicArray"

The following XML example shows the use of the SIGNAL object that is marked in Figure 13. The structure of the referenced SIGNAL object with the attributes `type="STRUCTURE-SIGNAL-TYPE"` and `MOST-DATATYPE="Record"` is described under "Record" (see section 4.8.3.2).

Example XML Code:

```
<fx:SIGNAL xsi:type="most:ARRAY-SIGNAL-TYPE" ID="_Signal_585"
MOST-DATATYPE="DynamicArray">
  <ho:SHORT-NAME>Data</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Data</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>offen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="Coding_Complex" />
  <most:ARRAY>
    <most:ELEMENT-SIZE>
      <most:LOWER-LIMIT>255</most:LOWER-LIMIT>
      <most:UPPER-LIMIT>255</most:UPPER-LIMIT>
    </most:ELEMENT-SIZE>
    <fx:SIGNAL-REF ID-REF="_Signal_Record" />
  </most:ARRAY>
</most:ACCESS>
```

```
<most:COLUMN>false</most:COLUMN>
<most:ELEMENT>>true</most:ELEMENT>
<most:FULL>>true</most:FULL>
<most:ROW>>true</most:ROW>
</most:ACCESS>
</fx:SIGNAL>
```

4.8.2.4 Data Structure LongArray

The structure and use of the data structure "LongArray" are—apart from the attribute `MOST-DATATYPE="LongArray"` of des SIGNAL object of the type `ARRAY-SIGNAL-TYPE`—identical to those of a "DynamicArray" (see section 4.8.3.4).

Element	Instructions for Providing Content
SIGNAL	Attribute: <code>type="ARRAY-SIGNAL-TYPE"</code> Attribute: <code>ID=" <identifier of this DS object> "</code> Attribute: <code>MOST-DATATYPE="LongArray"</code>

A MOST function of function class `ArrayWindow` accesses the data structure of type `LongArray` of the corresponding `MotherArray`. Therefore such an `ArrayWindow` uses a data structure of type `LongArray` as well. The only difference is the – usually – shorter length of the data structure; i.e. the number of rows (`ELEMENT-SIZE` of the SIGNAL object `ARRAY-SIGNAL-TYPE`) is limited to the length of the `ArrayWindow`.

4.8.2.5 Data Structure Map

The structure and use of data structure "Map" are—apart from the attribute `MOST-DATATYPE="Map"` of the SIGNAL object of the type `ARRAY-SIGNAL-TYPE`—identical to those of a "DynamicArray" (see section 4.8.3.4).

Element	Instructions for Providing Content
SIGNAL	Attribute: <code>type="ARRAY-SIGNAL-TYPE"</code> Attribute: <code>ID=" <identifier of this DS object> "</code> Attribute: <code>MOST-DATATYPE="Map"</code>

4.8.3 Data Structures with ACCESS-RESTRICTION-SIGNAL-TYPE

This section describes the Fibex4MOST model of the following data structures:

- Record
- Array
- DynamicArray
- LongArray
- Map

The SIGNAL objects with the attributes `type="STRUCTURE-SIGNAL-TYPE"` or `type="ARRAY-SIGNAL-TYPE"` are only used for structuring the Fibex XML and do not contain coding information. They reference the CODING object "Coding_Complex". This CODING is created once, if required, and does not contain any information that is actually used. The "Coding_Complex" is required because every SIGNAL has to reference a CODING. It is built as follows:

Element	Beschreibung
CODING	Attribute: ID="Coding_Complex"
CODING/SHORT-NAME	Complex_Parameter
CODING/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.

Example XML Code:

```
<fx:CODING ID="Coding_Complex">
  <ho:SHORT-NAME>Complex_Parameter</ho:SHORT-NAME>
</fx:CODING>
```

Access to single elements of the data structure is not modeled in Fibex. The access mode for individual data structure elements has to be identified by Fibex4MOST tool based on the underlying function class.

4.8.3.1 ACCESS-RESTRICTION-SIGNAL-TYPE

The ACCESS-RESTRICTION is used to represent the allowed access (i.e. full, column-wise, row-wise, and element-wise) to data structures. The ACCESS-RESTRICTION is to be used with arrays, records, array of records, dynamic arrays, long arrays, array windows, and maps.

Using with data structures array and record only the elements FULL and ELEMENT are allowed.

Element	Instructions for Providing Content
SIGNAL/ACCESS-RESTRICTION-SIGNAL-TYPE	Used with ist sub-elements to describe access restrictions for data structures. Only used in the following data structures: "Array", "Array of Record", "DynamicArray", "LongArray", and "Map"
SIGNAL/ACCESS-RESTRICTION-SIGNAL-TYPE /ACCESS/COLUMN	Determines whether individual columns of the data structure may be accessed (<code>true</code>) or not (<code>false</code>).
SIGNAL/ACCESS-RESTRICTION-SIGNAL-TYPE /ACCESS/ELEMENT	Determines whether individual elements of the data structure may be accessed (<code>true</code>) or not (<code>false</code>).
SIGNAL/ACCESS-RESTRICTION-SIGNAL-TYPE /ACCESS/FULL	Determines whether the data structure may be accessed in its entirety (<code>true</code>) or not (<code>false</code>).
SIGNAL/ACCESS-RESTRICTION-SIGNAL-TYPE /ACCESS/ROW	Determines whether individual rows of the data structure may be accessed (<code>true</code>) or not (<code>false</code>).

4.8.3.2 Data Structures Record and Record with Array

The data structure "Record" appears in two forms in MOST: A simple "Record" and a "Record" which contains an "Array" (called "Record with Array").

To describe data structures of type "Record" and "Record with Array", multiple Fibex structure elements SIGNAL and CODING are required. In following table, the required central SIGNAL object is described.

Element	Instructions for Providing Content
SIGNAL	Attribute: type="STRUCTURE-SIGNAL-TYPE" Attribute: ID=" <identifier of this DS object> " Attribute: MOST-DATATYPE="Record"
SIGNAL/CODING-REF	Attribute: ID-REF="Coding_Complex" This object does not contain a real CODING
SIGNAL/STRUCTURE/ RECORD-FIELDS/ RECORD-FIELD/ SEQUENCE-NUMBER	Position of the referenced SIGNAL in the directly following SIGNAL-REF The position is defined by a number greater or equal 0
SIGNAL/STRUCTURE/ RECORD-FIELDS/ RECORD-FIELD/ SIGNAL-REF	Attribute: ID-REF=" <identifier of a SIGNAL object> " The referenced SIGNAL can describe a MOST parameter or an Array data structure (type="ARRAY-SIGNAL-TYPE"). ("Record with Array")

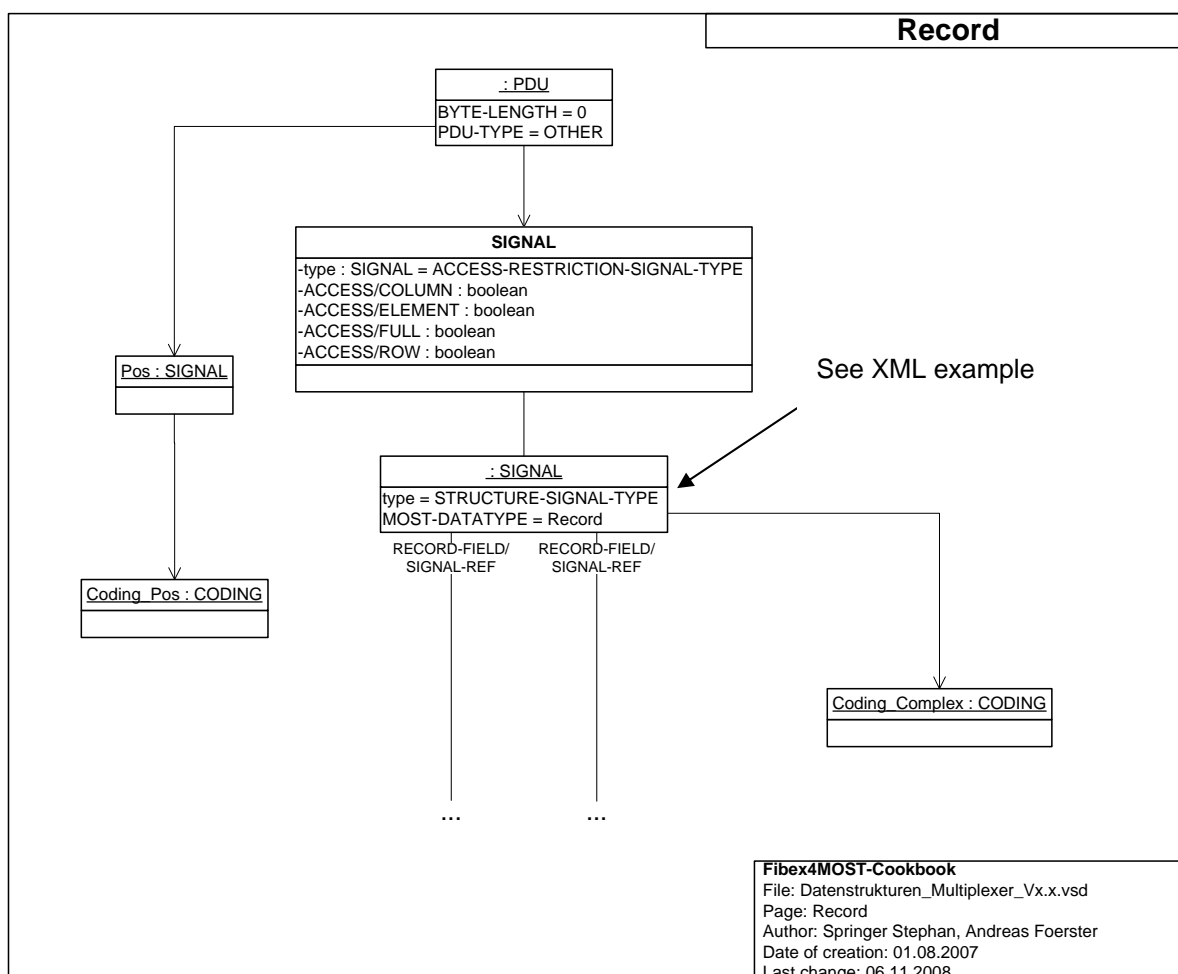


Figure 9: Object diagram of the Fibex model of the simple data structure "Record"

The following XML example shows the use of the SIGNAL object in Figure 9. The use of the other objects is not shown.

Example XML Code („Record“):

```
<fx:SIGNAL xsi:type="most:STRUCTURE-SIGNAL-TYPE" ID="_Signal_46"  
MOST-DATATYPE="Record">  
  <ho:SHORT-NAME>Data</ho:SHORT-NAME>  
  <ho:LONG-NAME xml:lang="en">Data</ho:LONG-NAME>  
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>  
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>  
  <fx:ELEMENT-REVISIONS>  
    <fx:ELEMENT-REVISION>  
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>  
      <ho:STATE>geschlossen</ho:STATE>  
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>  
    </fx:ELEMENT-REVISION>  
  </fx:ELEMENT-REVISIONS>  
  <fx:CODING-REF ID-REF="Coding_Complex" />  
  <most:STRUCTURE>  
    <most:RECORD-FIELDS>  
      <most:RECORD-FIELD>  
        <fx:SEQUENCE-NUMBER>0</fx:SEQUENCE-NUMBER>  
        <fx:SIGNAL-REF ID-REF="_Signal_47" />  
      </most:RECORD-FIELD>  
      <most:RECORD-FIELD>  
        <fx:SEQUENCE-NUMBER>1</fx:SEQUENCE-NUMBER>  
        <fx:SIGNAL-REF ID-REF="_Signal_48" />  
      </most:RECORD-FIELD>  
    </most:RECORD-FIELDS>  
  </most:STRUCTURE>  
</fx:SIGNAL>
```


Record with Array

The "Record" data structure can contain an "Array" as content of the last field. This construct is called a "Record with Array" in this document. The Fibex model of a "Record with Array" is illustrated in Figure 10.

The structure of the SIGNAL object shown in Figure 10 with the attributes `type="STRUCTURE-SIGNAL-TYPE"` and `MOST-DATATYPE="Record"` is identical to that of a "Record" (see above). The structure of the SIGNAL object with the attributes `type="ARRAY-SIGNAL-TYPE"` and `MOST-DATATYPE="Array"` is described in the "Array" section (see 4.8.3.3).

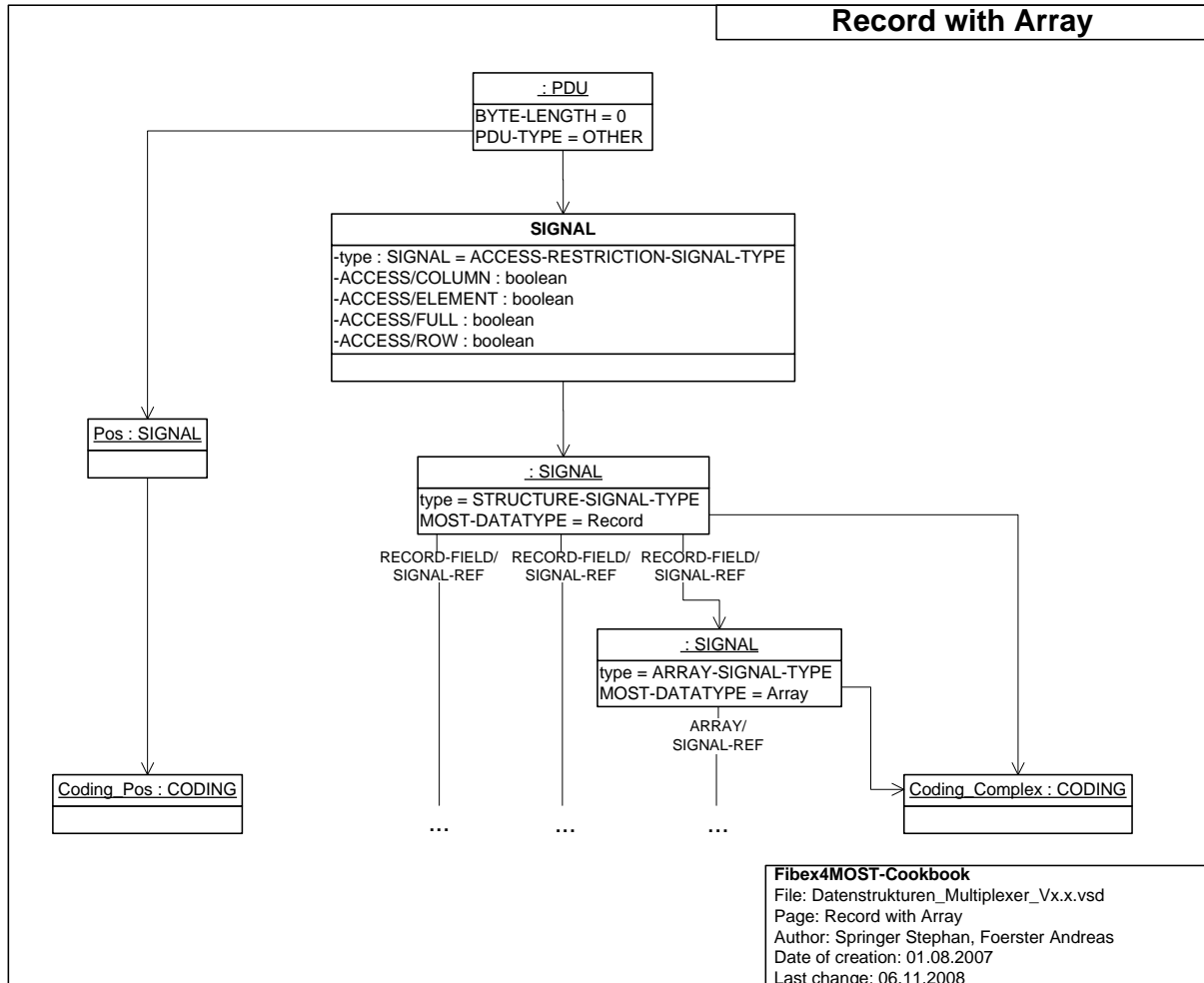


Figure 10: Object diagram of the Fibex model for data structure "Record with Array"

4.8.3.3 Data Structure Array and Array of Record

The data structure "Array" appears in two different forms in MOST: A simple "Array" and an "Array", which contains a "Record" (called "Array of Record").

An "Array" references exactly one sub-parameter. In the context of an "Array of Record", this sub-parameter is a "Record" data structure.

Array

To describe data structures of type "Array", the Fibex structuring elements SIGNAL and CODING are required. The following tables contain the ones that are necessary.

Element	Instructions for Providing Content
SIGNAL	Attribute: type="ARRAY-SIGNAL-TYPE" Attribute: ID=" <identifier of this DS object> " Attribute: MOST-DATATYPE="Array"
SIGNAL/CODING-REF	Attribute: ID-REF="Coding_Complex" This object does not contain a real CODING
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	<minimum length of the Array> Lower length limit of the Array
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	<maximum length of the Array> Upper length limit of the Array (max=255)
SIGNAL/ARRAY/SIGNAL-REF	Attribute: ID-REF=" <identifier of a SIGNAL object> " The referenced SIGNAL can describe a MOST parameter or a data structure "Record" (type="STRUCTURE-SIGNAL-TYPE"). ("Array of Record")

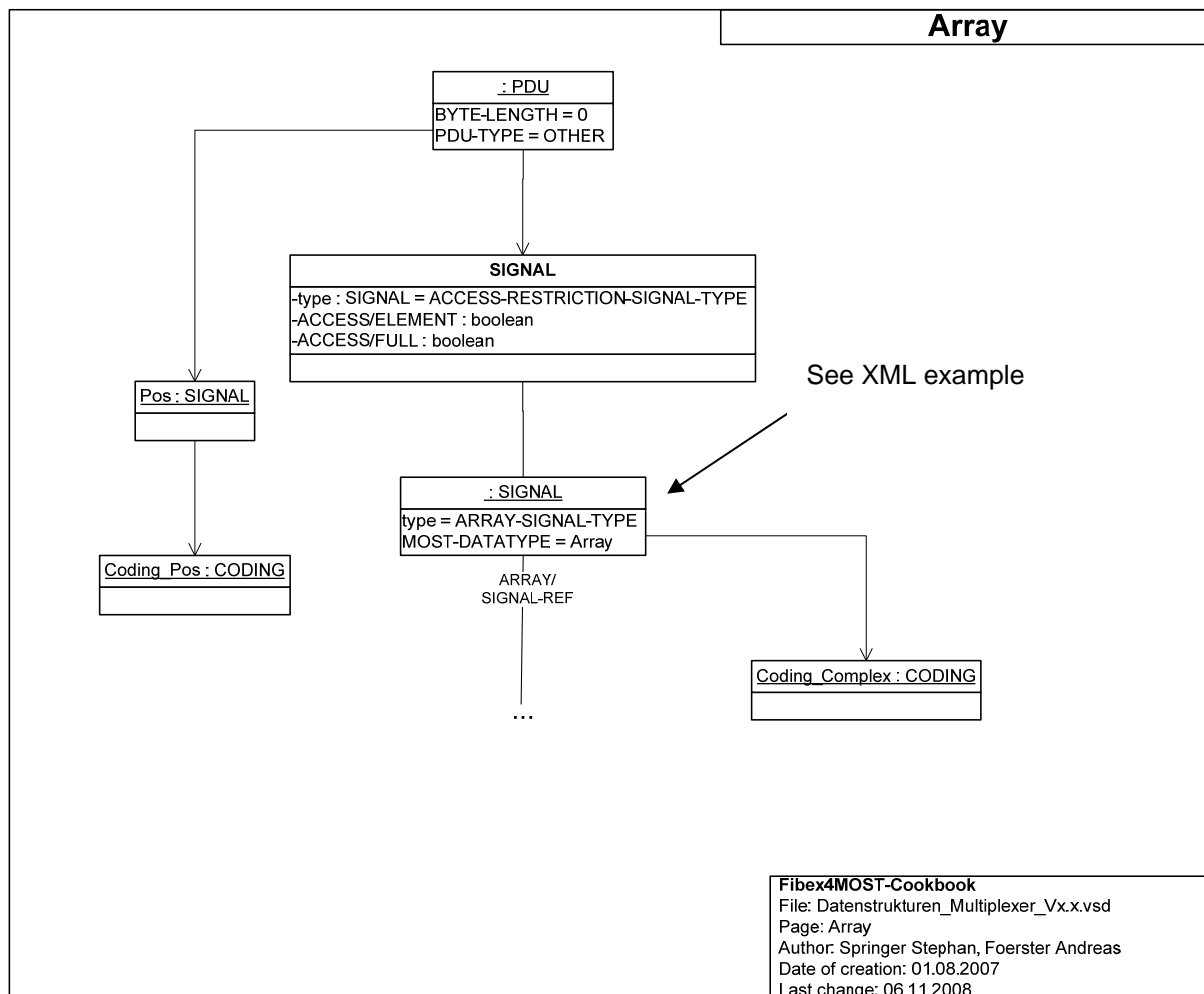


Figure 11: Object diagram of the Fibex model for data structure "Array"

The following table shows, as an example, the length definition of an Arrays and the corresponding representation of the Fibex export:

Length of Array	ELEMENT-SIZE / LOWER-LIMIT	ELEMENT-SIZE / UPPER-LIMIT
≥ 0	0	Maximum length for data structure
> 0	1	Maximum length for data structure
1..10	1	10
25	25	25

The following XML example shows the use of the SIGNAL object that is marked in Figure 11.

Example XML Code („Array“):

```
<fx:SIGNAL xsi:type="most:ARRAY-SIGNAL-TYPE" ID="_Signal_585"  
MOST-DATATYPE="Array">  
  <ho:SHORT-NAME>Data</ho:SHORT-NAME>  
  <ho:LONG-NAME xml:lang="en">Data</ho:LONG-NAME>  
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>  
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>  
  <fx:ELEMENT-REVISIONS>  
    <fx:ELEMENT-REVISION>  
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>  
      <ho:STATE>offen</ho:STATE>  
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>  
    </fx:ELEMENT-REVISION>  
  </fx:ELEMENT-REVISIONS>  
  <fx:CODING-REF ID-REF="Coding_Complex" />  
  <most:ARRAY>  
    <most:ELEMENT-SIZE>  
      <most:LOWER-LIMIT>255</most:LOWER-LIMIT>  
      <most:UPPER-LIMIT>255</most:UPPER-LIMIT>  
    </most:ELEMENT-SIZE>  
    <fx:SIGNAL-REF ID-REF="_Signal_586" />  
  </most:ARRAY>  
</fx:SIGNAL>
```

Array of Record

An "Array" data structure can contain exactly one parameter of type "Record". Such an "Array of Record" is shown in Figure 12. The structure of the SIGNAL object of type ARRAY-SIGNAL-TYPE corresponds to that of a simple "Array" (see above). The structure of the SIGNAL object with the attributes type="STRUCTURE-SIGNAL-TYPE" and MOST-DATATYPE="Record" is described in the "Array" section (see 4.8.3.2).

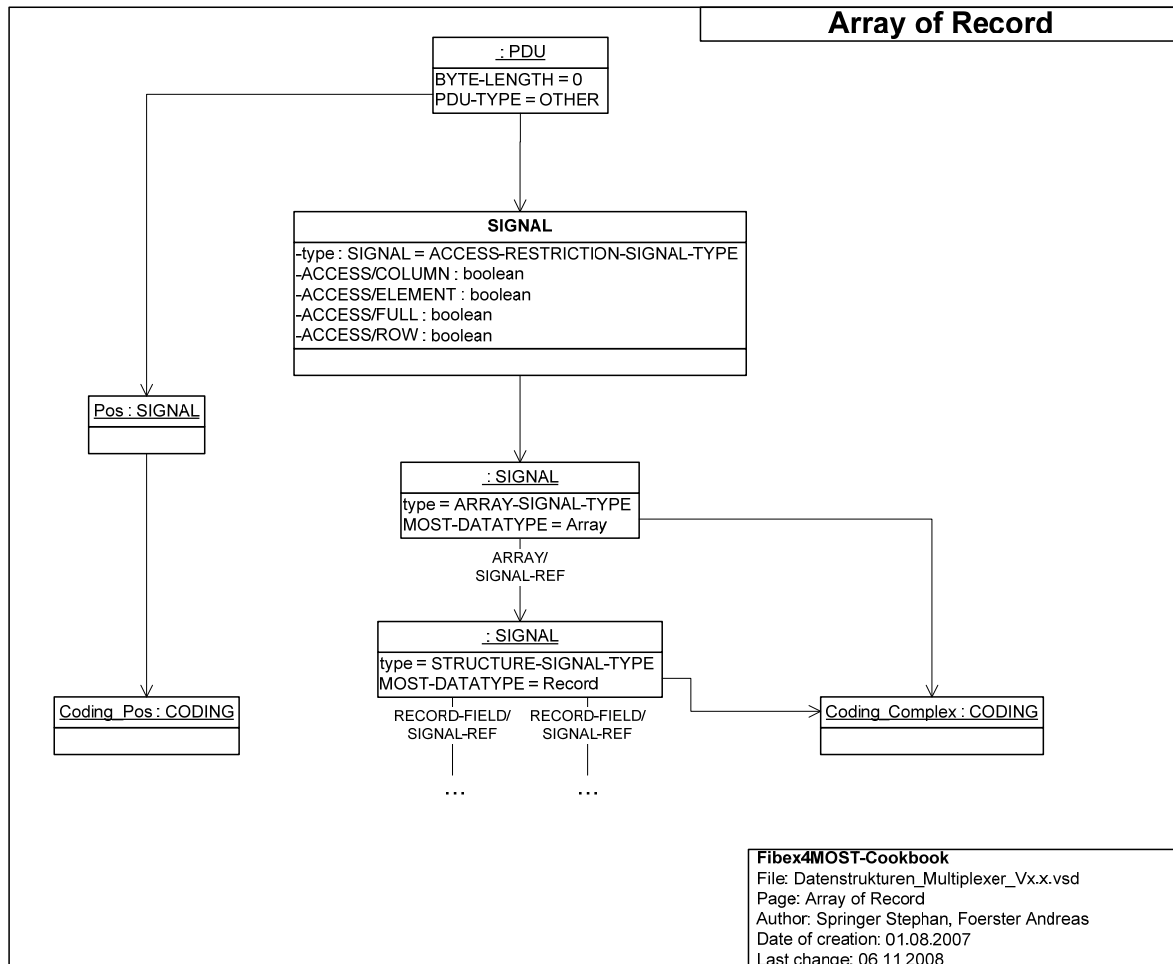


Figure 12: Object diagram of the Fibex model for the data structure "Array of Record"

4.8.3.4 Data Structure DynamicArray

To describe a data structure of type "DynamicArray", multiple Fibex structuring elements SIGNAL and CODING are required. The following tables contain the ones that are necessary.

Element	Instructions for Providing Content
SIGNAL	Attribute: type="ARRAY-SIGNAL-TYPE" Attribute: ID=" <identifier of this DS object> " Attribute: MOST-DATATYPE="DynamicArray"
SIGNAL/CODING-REF	Attribute: ID-REF="Coding_Complex" This object does not contain a real CODING
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	<minimum length of the Array> Lower length limit of the Array
SIGNAL/ARRAY/ ELEMENT-SIZE/LOWER-LIMIT	<maximum length of the Array> Upper length limit of the Array (max=65535)
SIGNAL/ARRAY/ SIGNAL-REF	Attribute: ID-REF=" <identifier of a SIGNAL object> " The referenced SIGNAL is a data structure "Record" (type="STRUCTURE-SIGNAL-TYPE" and MOST-DATATYPE="Record")

Element	Instructions for Providing Content
SIGNAL	Attribute: type="STRUCTURE-SIGNAL-TYPE" Attribute: ID=" <identifier of this DS object> " Attribute: MOST-DATATYPE="Record"
SIGNAL/CODING-REF	Attribute: ID-REF="Coding_Complex" This object does not contain a real CODING

The length definition of the "DynamicArray" is applied in the same way as when dealing with the data structure "Array" (see section 4.8.3.3).

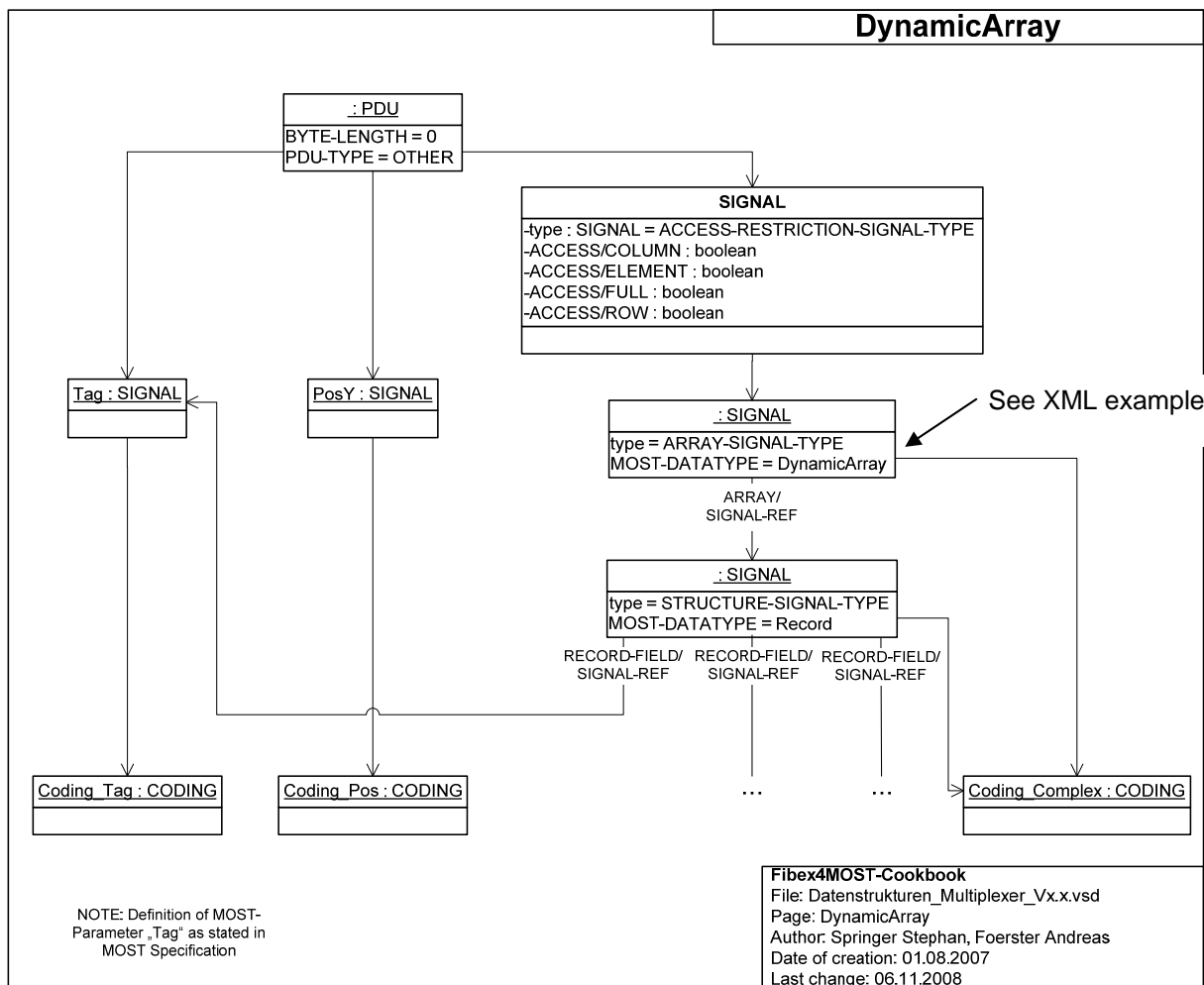


Figure 13: Object diagram of the Fibex model for data structure "DynamicArray"

The following XML example shows the use of the SIGNAL object that is marked in Figure 13. The structure of the referenced SIGNAL object with the attributes `type="STRUCTURE-SIGNAL-TYPE"` and `MOST-DATATYPE="Record"` is described under "Record" (see section 4.8.3.2).

Example XML Code:

```
<fx:SIGNAL xsi:type="most:ARRAY-SIGNAL-TYPE" ID="_Signal_585"
MOST-DATATYPE="DynamicArray">
  <ho:SHORT-NAME>Data</ho:SHORT-NAME>
  <ho:LONG-NAME xml:lang="en">Data</ho:LONG-NAME>
  <ho:DESC xml:lang="en" TYPE="Standard">Description</ho:DESC>
  <ho:DESC xml:lang="de" TYPE="Standard">Beschreibung</ho:DESC>
  <fx:ELEMENT-REVISIONS>
    <fx:ELEMENT-REVISION>
      <ho:REVISION-LABEL>3</ho:REVISION-LABEL>
      <ho:STATE>offen</ho:STATE>
      <ho:DATE>2001-03-12T10:42:42</ho:DATE>
    </fx:ELEMENT-REVISION>
  </fx:ELEMENT-REVISIONS>
  <fx:CODING-REF ID-REF="Coding_Complex" />
  <most:ARRAY>
    <most:ELEMENT-SIZE>
      <most:LOWER-LIMIT>255</most:LOWER-LIMIT>
    </most:ARRAY>
  </fx:SIGNAL>
```

```

    <most:UPPER-LIMIT>255</most:UPPER-LIMIT>
  </most:ELEMENT-SIZE>
  <fx:SIGNAL-REF ID-REF="_Signal_Record" />
</most:ARRAY>
<most:ACCESS>
  <most:COLUMN>>false</most:COLUMN>
  <most:ELEMENT>>true</most:ELEMENT>
  <most:FULL>>true</most:FULL>
  <most:ROW>>true</most:ROW>
</most:ACCESS>
</fx:SIGNAL>

```

4.8.3.5 Data Structure LongArray

The structure and use of the data structure "LongArray" are—apart from the attribute `MOST-DATATYPE="LongArray"` of des SIGNAL object of the type `ARRAY-SIGNAL-TYPE`—identical to those of a "DynamicArray" (see section 4.8.3.4).

Element	Instructions for Providing Content
SIGNAL	Attribute: <code>type="ARRAY-SIGNAL-TYPE"</code> Attribute: <code>ID=" <identifier of this DS object> "</code> Attribute: <code>MOST-DATATYPE="LongArray"</code>

A MOST function of function class `ArrayWindow` accesses the data structure of type `LongArray` of the corresponding `MotherArray`. Therefore such an `ArrayWindow` uses a data structure of type `LongArray` as well. The only difference is the – usually – shorter length of the data structure; i.e. the number of rows (`ELEMENT-SIZE` of the SIGNAL object `ARRAY-SIGNAL-TYPE`) is limited to the length of the `ArrayWindow`.

4.8.3.6 Data Structure Map

The structure and use of data structure "Map" are—apart from the attribute `MOST-DATATYPE="Map"` of the SIGNAL object of the type `ARRAY-SIGNAL-TYPE`—identical to those of a "DynamicArray" (see section 4.8.3.4).

Element	Instructions for Providing Content
SIGNAL	Attribute: <code>type="ARRAY-SIGNAL-TYPE"</code> Attribute: <code>ID=" <identifier of this DS object> "</code> Attribute: <code>MOST-DATATYPE="Map"</code>

5 UNIT-SPEC - Declaration of Units

Within the object UNIT-SPEC the physical quantities and units are described. Three Fibex object groups are used: PHYSICAL-DIMENSIONS, UNITGROUPS and UNITS.

UnitDefCode, which is used exclusively for MOST, is stored in a DESC element with the attribute TYPE="UnitDefCode". This element is only used for MOST units.

Element	Instructions for Providing Content
UNIT-SPEC/ PHYSICAL-DIMENSIONS/ PHYSICAL-DIMENSION	Attribute: ID=" <Name> " Used as identifier of a physical quantity. For every physical quantity, this element and its sub-elements are created.
UNIT-SPEC/ PHYSICAL-DIMENSIONS/ PHYSICAL-DIMENSION/ SHORT-NAME	Contains the automatically generated short name of the physical quantity; corresponds to the English name (only letters and underscore allowed).
UNIT-SPEC/ PHYSICAL-DIMENSIONS/ PHYSICAL-DIMENSION/ SHORT-NAME/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
UNIT-SPEC/ PHYSICAL-DIMENSIONS/ PHYSICAL-DIMENSION/DESC	Attribute: lang="en" English description of the physical quantity
UNIT-SPEC/ PHYSICAL-DIMENSIONS/ PHYSICAL-DIMENSION/	Possible sub-elements are: LENGTH-EXP MASS-EXP TIME-EXP CURRENT-EXP TEMPERATURE-EXP MOLAR-AMOUNT-EXP LUMINOUS-INTENSITY-EXP
UNIT-SPEC/UNITGROUPS/ UNITGROUP/SHORT-NAME	<Name of the UnitDefGroup> e.g.: Distance, Frequency, ...
UNIT-SPEC/UNITGROUPS/ UNITGROUP/CATEGORY	EQUIV-UNITS
UNIT-SPEC/UNITGROUPS/ UNITGROUP/UNIT-REFS/ UNIT-REF	Attribute: ID-REF=" <identifier of the UNIT object> " Created for every UNIT that belongs to this UNITGROUP
UNIT-SPEC/UNITS/UNIT	Attribute: ID=" <Name> " Used as identifier of the unit for referencing. For every unit, UNIT with its sub-elements is created
UNIT-SPEC/UNITS/UNIT/ SHORT-NAME	Contains the automatically generated short name of the unit, which corresponds to the English name (only letters and underscore allowed).
UNIT-SPEC/UNITS/UNIT/ SHORT-NAME/LONG-NAME	Attribute: lang="en" Contains the same value as SHORT-NAME.
UNIT-SPEC/UNITS/UNIT/ DESC	Attribute: lang="en" English description of the unit
UNIT-SPEC/UNITS/UNIT/ DESC	Attribute: TYPE="UnitDefCode" Contains the value "UnitDefCode" of a MOST unit

Element	Instructions for Providing Content
UNIT-SPEC/UNITS/UNIT/ DISPLAY-NAME	Contains a sequence of characters that is the display name of the unit
UNIT-SPEC/UNITS/UNIT/ PHYSICAL-DIMENSION-REF	Attribute: ID-REF=" <identifier of the corresponding physical quantity> "

In the following XML example, only a part of the UNIT-SPEC object is shown. In an actual Fibex document, all used physical quantities and units have to be defined.

Example XML Code:

```

<ho:UNIT-SPEC>
  <ho:PHYSICAL-DIMENSIONS>
    <ho:PHYSICAL-DIMENSION ID="siComb_specific_volume">
      <ho:SHORT-NAME>specific_volume</ho:SHORT-NAME>
      <ho:LONG-NAME xml:lang="en">specific volume</ho:LONG-NAME>
      <ho:DESC xml:lang="en">derived quantity: specific volume</ho:DESC>
      <ho:LENGTH-EXP>3</ho:LENGTH-EXP>
      <ho:MASS-EXP>-1</ho:MASS-EXP>
    </ho:PHYSICAL-DIMENSION>
    ...
  </ho:PHYSICAL-DIMENSIONS>
  <ho:UNITGROUPS>
    <ho:UNITGROUP>
      <ho:SHORT-NAME>Distance</ho:SHORT-NAME>
      <ho:CATEGORY>EQUIV-UNITS</ho:CATEGORY>
      <ho:UNIT-REFS>
        <ho:UNIT-REF ID-REF="_unit_2" />
        ...
      </ho:UNIT-REFS>
    </ho:UNITGROUP>
    ...
  </ho:UNITGROUPS>
  <ho:UNITS>
    <ho:UNIT ID="unit_day">
      <ho:SHORT-NAME>day</ho:SHORT-NAME>
      <ho:LONG-NAME xml:lang="en">day</ho:LONG-NAME>
      <ho:DESC TYPE="UnitDefCode">21</ho:DESC>
      <ho:DISPLAY-NAME>d</ho:DISPLAY-NAME>
      <ho:OFFSET-SI-TO-UNIT>86400</ho:OFFSET-SI-TO-UNIT>
      <ho:PHYSICAL-DIMENSION-REF ID-REF="siBase_sec" />
    </ho:UNIT>
    ...
  </ho:UNITS>
</ho:UNIT-SPEC>

```

Notes:

6 Appendix A: Index of Figures

Figure 1: Object diagram of the Fibex model of data type "Stream" as "Composed Stream"	49
Figure 2: Object diagram of the Fibex model for data type "Stream" as "List in Stream"	51
Figure 3: Object diagram of the Fibex model of data type "Stream" as "MUX Stream" with 2 StreamCases	53
Figure 4: Object diagram of the Fibex model of the simple data structure "Record"	61
Figure 5: Object diagram of the Fibex model for data structure "Record with Array"	63
Figure 6: Object diagram of the Fibex model for data structure "Array"	65
Figure 7: Object diagram of the Fibex model for the data structure "Array of Record"	66
Figure 8: Object diagram of the Fibex model for data structure "DynamicArray"	68
Figure 9: Object diagram of the Fibex model of the simple data structure "Record"	71
Figure 10: Object diagram of the Fibex model for data structure "Record with Array"	73
Figure 11: Object diagram of the Fibex model for data structure "Array"	75
Figure 12: Object diagram of the Fibex model for the data structure "Array of Record"	77
Figure 13: Object diagram of the Fibex model for data structure "DynamicArray"	79

7 Appendix B: Index of Tables

Table 1-1: Document references7