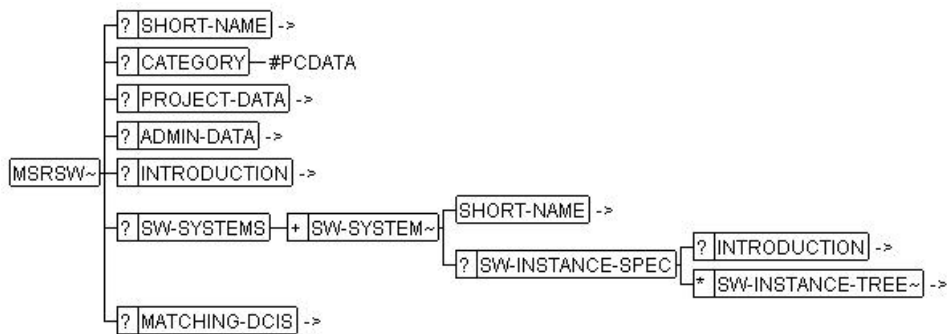# ASAM-MCD-2MC

Using ASAM-MCD-2MC 2.0 to implement Calibration Data File

a study

Abstract

This document describes, how *ASAM-MCD-2MC 2.0* can be utilized to implement Calibration Data File. The document refers to *[External Document: Calibration Data File Specification / Date: 2001-07-19 / Publisher: Jeff Kainz]* . The doucment comprises of two parts:

- Description structured according to the specification
- Element an Attribute description of an appropriate subset of *MSRSW.DTD 2.2.0* with is the result of the common ASAM/MSR workgorup.

This document show that mainly all requirements of *CDF* are covered.

Table of Contents

07/08/2002 17:05:45 msrsw-tr-cdf.xml

07/08/2002 17:05:45 msrsw-tr-cdf.xml

Introduction

Companies **MSR-MEDOC/ASAM-MCD Working Group [MSR-MEDOC]**

Name Roles	Departement	Address	Contact
Bernhard Weichel	Robert Bosch GmbH, GS/EMW4		Phone +(49) 711 1389969 berhard.weichel@de. bosch.com

Version Information

Document Part	Editor			
	Company	Version	State	Remarks
2 WD 2002-06-02 For details refer to nr. 1, Page 41				

07/08/2002 17:05:45 msrsw-tr-cdf.xml

1 Scope

The scope of the Calibration Data File Format specification (*CDF*) is to identify a commeon ASAM standard file format that can bw used to interchange data between vendor-independant calibration, simulation, documentation, spreadsheet and data acquisition tools. A secondary function is to allow calibration engineers to read the file into a text or *XML editor* and easily change values.

1.1 Identification

This document describes how these requirements can be satisfied by applying *ASAM-MCD-2MC 2.0* as a particular subset. The document is the result of the task defined at the ASAM-MCD Working group on 16.July.2001.

1.2 General approach

In order to illustrate the desired mapping, a subset of *MSRSW.DTD* was created:

- deleting all elements not required by the CDF use case

- Attributes were not removed

- Some of the elements were retained even if they were not there in CDF. This is in order to illustrate some additional possibilites.

- Some wrapper elements appear to make no sense. They are still retained in order to maintain compliance with *MSRSW.DTD* thus providing an instance compatibility if the additional features are required.

- cardinalities were not adapted according to the *CDF specification* because of the reasons mentioned in .

2   How to read this document

This document is divided in two parts:

It is recommended to start reading at Chapter 1 Mapping CDF to ASAM-MCD-2MC p. 10.

2.1   Terminology

The terminology of this document follows the one in the *CDF* specification whereever possible. In order to be consistent with the doucments of the *MSR/ASAM working group*, the DTD is referred as *MSRSW.DTD 2.2.0*.

2.2   Additional Documentation

For further information refer to additional documentation available:

- An introduction to the *ASAM-MCD-2MC.DTD*/*MSRSW.DTD* in particular dedicated to an ASAP based audience (*[External Document: ASAP2 V2.x, MSRSW V2.2.0 DTD final release]*). One should probably start reading this.

- The documentation to *MSRDCI* as finished so far. *[External Document: Structured Principles for the MSR - DCI]* . This may help to obtain an initial understanding of the MSRDCI approach.

- The design rules for MSR DTDs, given in *[External Document: Concepts of the MSR application profile V2.x.x]*

- The elements and attributes are described in *[External Document: MSRSW.DTD Elements and attributes]*

3 The following conventions apply

This document is written using *MSRREP.DTD*. The following conventions apply to this document:

<msrsw> SGML elements are noted as technical term **[type]**=*SGMLTAG*.

[type] SGML attributes are noted as technical term **[type]**=*SGML-attribute*.

sgml-attribute Values of SGML attributes or discrete values for elements are noted as technical term **[type]**=*code*

ASAM-MCD-2MC 2.0 The considered languages resp. DTDs are marked as technical term **[type]**=*product*.

ASAM The committees are noted as **[type]**=*organization*

CDF Keywords from *CDF* are marked as technical term **[type]**=*code*.

ECU Objects in general are marked as technical terms **[type]**=*other*. This might be automotive equipments general objects such as variables etc.

Note that an index of the technical terms is given at the end of the document.

3.1 Graphical conventions used in DTD diagrams

Since a *DTD* represents a grammar, it can be visualized graphically as a tree. The visualization in this document uses the notation given in .

Figure 1: Graphical notation of DTDs

The meaning of the symbols is:

PCDATA The element content is **Processable Character Data (PCDATA)**. This is data that consists of zero or more characters of both text and markup. PCDATA is used to indicate that all markup delimiters defined in the SGML declaration will be recognized by the parser as markup in the given element rather than data characters.

RCDATA The element content is **Replaceable Character Data (RCDATA)** . this is data that consists of zero or more characters, in which references to substitutions

	Using ASAM-MCD-2MC 2.0 to implement Calibration Data File	Page: 9 / 41
MSR		Date: 2002-06-02
	Graphical conventions used in DTD diagrams	State: WD

are not recognized (i.e. RCDATA may contain text and entity references (starting with "&") , but no sub-elements)[1].

CDATA — **Character Data (CDATA)** consists of zero or more text characters, where no markup of any kind is recognized.

ANY — a terminal type indicating that the object may contain text or any element defined in the model[2].

EMPTY — a terminal type keyword used to indicate that there is no data (i.e. no content, sub-elements or end-tags) for the object allowed in the document instance. This keyword is often used to describe elements that are placeholders or are pointers to external or system-generated data.

One — indicates that the element or the model group occurs exactly once.

ZERO-OR-ONE — indicates that the element or the model group is optional.

ONE-OR-MORE — indicates that the element or the model group occurs multiple times but at least once.

ZERO-OR-MORE — indicates that the element or the model group occurs multiple times but also can be missed (optional).

ORDERED — a connector used to specify that the sibling objects must appear in the document in the order shown in the model.

UNORDERED — a connector used to specify that the sibling objects can appear in any order in the document[3].

SELECTION — a connector used to specify that only one of the sibling objects can appear in the document.

ELEMENT — indicates a single SGML structure element.

COLLAPSED — indicates, that the content of the element is not displayed here.

Hint:
With respect to XML compliance, the following features are not used:

RCDATA, CDATA, ANY, EMPTY, UNORDERED connection,

---

[1]

[2]

[3]

07/08/2002 17:05:45 msrsw-tr-cdf.xml

1 Mapping CDF to ASAM-MCD-2MC

1.1 File format

MSRSW.DTD can be used as SGML as well as XML. The difference is mainly in the tools which are used. The instances are are compatible between XML and SGML.

1.1.1 Character Set Definition

When used in XML style, the *CDF* file may be encoded in any of the character sets permitted for XML. In western countries usage of ISO-8850-1 is a good choice. XML recommendation specifies UTF-8 as default, but this prevents non unicode edtitors from safely handling the files.

1.1.2 General

As specified in the *CDF* specification, an *MSRSW.DTD* file is also (as any XML file) composed of several sections, logcial groups. The file shall adere to XML respectively SGML standards.

If no information is available, then the field shall be left out. For this reason, every element is kept optional. The omission of an element is much easier to detect than the fact that an element is empty (e.g. because of insignificant whit space).

The general structure of the file is given in the following illustration:
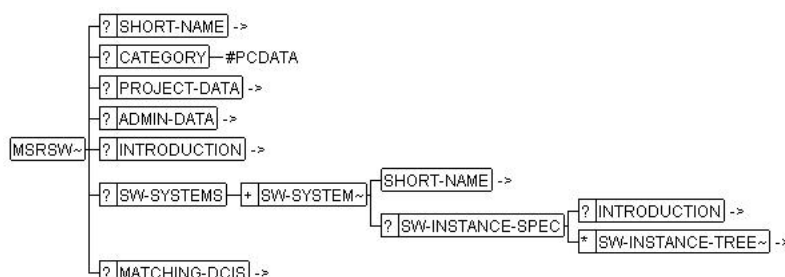
Figure 2: General structure

<MSRSW> is the root element of the file. Mainly it consists of the following sections:

<SHORT-NAME>	A short designator of the file. Usually this is something like filename body a project identifier.
<CATEGORY>	This is denotes the category of the file. In case of CDF the content would be CDF. The category here mainly corresponds to **<MATCHING-DCI>**.
<PROJECT-DATA>	This denotes all organisational and project relevant data.
<ADMIN-DATA>	This denotes al administrative data related to the current file.
<INTRODUCTION>	This is a general note that serves as an introduction to the current file.
<SW-SYSTEMS>	This section receives the intended content of the CDF file. As an extension to *CDF* it supports multiple ECUs within one file.
<MATCHING-DCIS>	This allows to refer to a formal specification which can be used to validate the current file.

	Using ASAM-MCD-2MC 2.0 to implement Calibration Data File	Page:	11 / 41
		Date:	2002-06-02
MSR	Elements of CDF specification	State:	WD

1.1.3 Filename Definition

The file shall be saved according to committed conventions.

it is recommended to kee the extension as XML because most XML tools refer to this extension. The file could be marked as CDF file by adding .CDF before the extension:

Example: mycalibrations.cdf.xml

1.2 XML Schemas and Namespaces

The DTD and schemas of released file formats shall be stored on the ASAM web server. Each file instance is required to supply a reference to the governig schema or DTD by an appropriate DOCTYPE declaration. By this, CDF instances are required to be valid XML documents.

XML schema as well as DTD is not strong enough to specifiy all details. Therefore it is recommended also to use Document Content Information. References to the matching DCI's is kept in the element **<matching-dcis>**.

Hint:
There are not many XML tools around which really support namespaces. It is therefore recommended not to request a namespace-attribute. In conjunction with a general ASAM-XML strategy, this item should be worked out.

1.3 Elements of CDF specification

The following table specifies and comments the correspondance between elements of *CDF* and *MSRSW.DTD*. It is given in the same sequence as *[External Document: Calibration Data File Specification / Date: 2001-07-19 / Publisher: Jeff Kainz / Relevant Position:]* .

Hint:
CDF covers one particular use case while *ASAM-MCD-2MC 2.0* and *MSRSW.DTD* covers multiple use cases. Therefore sometimes elements (mostly wrappers) occur which appear to be superfluous. These elements were retained in oder to keep structural compliance.

Table 1: Mapping CDF elements to ASAM-MCD-2MC 2.0

CDF-Element	ASAM-MCD-2MC 2.0	Comment
CAL-DATA	**<MSRSW>**	The root element
COMPANY-INFO	**<LONG-NAME>** in **<COMPA-NY>** within **<PROJECT-DATA>**	There is no direct correspondance. It appears that *CDF* uses it only for copyright purposes. Although an easier workaround may be possible, usage of a substructure of **<PROJECT-DATA>** is proposed

07/08/2002 17:05:45 msrsw-tr-cdf.xml

Table 1 (Cont.): Mapping CDF elements to ASAM-MCD-2MC 2.0

CDF-Element	ASAM-MCD-2MC 2.0	Comment
FILENAME		There is no direct correspondance. *MSRSW.DTD* allows to keep version and company specific filenames (**<ENTITY-NAME>** within **<ADMIN-DATA>**). Since the use case is not fully clear, this feature is not taken over the the proposal made in this document.
CHANGE-HISTORY	**<DOC-REVISIONS>**	The *MSRSW.DTD* is more powerful and therefore somewhat more complex in case of change-history. In particular it supports multiple changes per revision.
CHANGE	**<MODIFICATION>**	
USERNAME	**<TEAM-MEMBER-REF>**	If the instance also populates **<COMPANY>** with **<TEAM-MEMBERS>**, then using **[ID]**/**[IDREF]** is still possible. Otherwise Usage of ID/IDREF should be turned off.
DATE	**<DATE>**	
TOOL		There is no direct correspondance. The exact usecase shall be worked out. MSR-SW.DTD provides the following options: • **<PRIVATE-CODE>** in **<ADMIN-DATA>** • XML Comments • **<SPECIAL-DATA>**
NOTE in *CHANGE*	**<REASON>**	
DATA-VERISION	**<REVISION-LABEL>**	This mainly refers to the version of the file, not the version of the data in the file which might be different and therefore be kept within **<CS-HISTORY>**.
	<SW-CS-DATA-IDENTIFIER> within **<SW-INSTANCE-TREE>**	This must be worked out in greater detail depending on the intended use cases.
NOTE in *CAL-DATA*	**<INTRODUCTION>** in **<MSR-SW>**	This represents the introduction not to the entire file.

07/08/2002 17:05:45 msrsw-tr-cdf.xml

Table 1 (Cont.): Mapping CDF elements to ASAM-MCD-2MC 2.0

CDF-Element	ASAM-MCD-2MC 2.0	Comment
CAL-LIST	**<SW-INSTANCE-TREE>**	This is somewhat more complex because *MSRSW.DTD* supports multiple systems (e.g. ECUs) within one file. For each of these systems multiple calibration data sets are possible (as intended with the multiple *CAL-LIST*s).
NAME	**<SHORT-NAME>**	Each object (calibration lists as well as calibration objects) has a unique name.
SYMBOLIC-FILE		There is no direct correspondance. Since *CDF* works on a physical level, it is even possible to work without such a symbolic file. In order to handle this there are multiple options (see Chapter 1.3.1 Options for SYMBOLIC-FILE and TARGET-IMAGE p. 14)
NOTE in *CAL-LIST*	**<INTRODUCTION>** in **<SW-INSTANCE-SPEC>**	Since *MSRSW.DTD* supports multiple systems within one file, this is an enhancement to the *CDF* requests.
	<DESC> in **<SW-INSTANCE-TREE>**	This corresponds to the intention of *CDF specification*.
CAL-ITEM	**<SW-INSTANCE>**	Each calibration item is represented by an SW-INSTANCE. *MSRSW.DTD* itself provides more features (variant coding, object oriented desing, arrays, structures). Therefore the content models do not match one by one.
NAME in *CAL-ITEM*	**<SHORT-NAME>**	
AXIS	**<SW-AXIS-CONTS>**	The attributes of *AXIS* in *CDF* correspond to child elements in *MSRSW.DTD* such as **<SW-UNIT-REF> <SW-AXIS-INDEX>**
VALUE	**<V>** for numerical values **<VT>** for string values	*MSRSW.DTD* puts each value in an element of its own. This makes it much easier to handle the file in XML-tools. The format attribute is covered by using **<V>** respectively **<VT>**

07/08/2002 17:05:45 msrsw-tr-cdf.xml

	Using ASAM-MCD-2MC 2.0 to implement Calibration Data File	Page:	14 / 41
		Date:	2002-06-02
MSR	Options for SYMBOLIC-FILE and TARGET-IMAGE	State:	WD

Table 1 (Cont.): Mapping CDF elements to ASAM-MCD-2MC 2.0

CDF-Element	ASAM-MCD-2MC 2.0	Comment
ANNOTATION		in the *CDF specification*, the difference to NOTE is not really clear. If Note is a short description of the CAL-ITEM in general, then **<DESC>** witin **<SW-INSTANCE-PROPS-VARIANT>** is intended to be used.
NOTE within *CAL-ITEM*	**<REMARK>** within **<SW-CS-HISTORY>**	

1.3.1 Options for SYMBOLIC-FILE and TARGET-IMAGE

As *SYMBOLIC-FILE* and *TARGET-IMAGE* is not a matter of the pyhsical representation of calibration data, it is not handled in the proposale here. It appears to be an M:N relationship betweenn calibration data, target image and the symbolic file. In particular one physical set of data can be loaded in multiple systems if the calibration items are available. Also, one physical set of data may result in multiple target images according to the various possible symbolic files.

The following options would apply .

- Derive the filename from **<SW-CS-PROGRAM-IDENTIFIER>**

- Keep this information within **<PRIVATE-CODES>** within **<ADMIN-DATA>** (this is currently removed in the proposal here)

- Keep this information within **<SPECIAL-DATA>** (this is currently removed in the proposal here).

- Keep this information in a separate file (e.g. a Catalog using *MSRCC.DTD*).

2 CDF relevant Elements in ASAM-MCD-2MC 2.0/M-SRSW.DTD 2.2.0

This chapter describes all proposed elements in alphabetical order.

2.1 ADMIN-DATA

Description

<ADMIN-DATA> can be used to capture all information regarding document management such as used language and versioning.

Example

Properties

Used in: MSRSW p. 22

Content: LANGUAGE p. 20,   DOC-REVISIONS p. 18
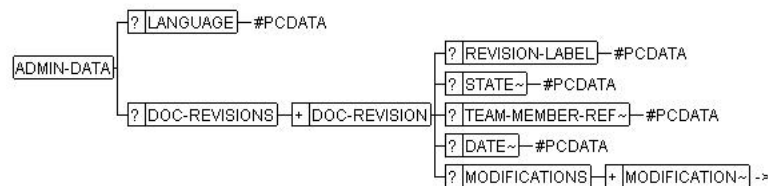
2.2 CATEGORY

Description

<CATEGORY> allows to denote specific, mainly process dependant modes of its parent element. This can be used by generic checkers (e.g. a *DCI checker*) to perform a more detailed evaluation of the substructure.

The possible values are described at the parent elements.

Example

The following example woud mark an MSRSR-File as an CDF flie.

Properties

Used in: MSRSW p. 22,   SW-INSTANCE p. 33,   SW-INSTANCE-TREE p. 35

Content: Text

2.3 CHANGE

Description

<CHANGE> can be used to describe the change performed in one particular modification.

Example

For an example, see Chapter 2.1 ADMIN-DATA p. 15.

Properties

Used in: MODIFICATION p. 21

Content: Text

```
CHANGE—#PCDATA
```

2.4 COMPANIES

Description

<COMPANIES> can be used to describe the companies being involved in the procject. With respect to *CDF* this is used mainly to specify copyright information and the involved persons. Usually there is more than one company involved in the project. The company which generated the file is determined by following **<TEAM-MEMBER-REF>** within **<ADMIN-DATA>**. In simple cases there is only one company, so this lookup is not really necessary.

Example

For an example, see Chapter 2.22 PROJECT-DATA p. 24

Properties

Used in: PROJECT p. 23

Content: COMPANY p. 16

```
COMPANIES—+—COMPANY~—┬—LONG-NAME ->
                      ├—SHORT-NAME ->
                      └—*—TEAM-MEMBER~—┬—LONG-NAME ->
                                       ├—SHORT-NAME ->
                                       ├—?—ROLES~—+—ROLE ->
                                       └—?—DEPARTMENT—#PCDATA
```
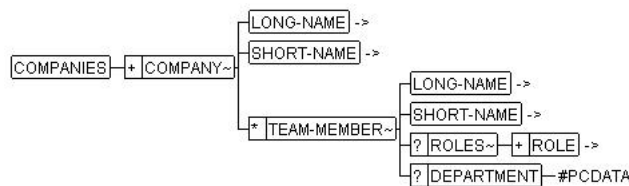
2.5 COMPANY

Description

<COMPANY> is used to capture the properties of one particular company in the project. With respect to *CDF*-Format, **<LONG-NAME>** within **<COMPANY>** corresponds to *COMPANY-INFO*.

Example

For an example, see Chapter 2.22 PROJECT-DATA p. 24

07/08/2002 17:05:45 msrsw-tr-cdf.xml

Properties

Used in: COMPANIES p. 16

Content: LONG-NAME p. 20, SHORT-NAME p. 26, TEAM-MEMBER p. 38
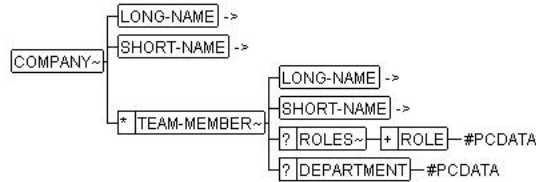
Attribute	Type	legal content	Remarks
[ROLE] (required)	namedtokengroup	• MANUFAC- TURER • SUPPLIER	Role of the company within the current project

2.6 DATE

Description

<DATE> ist used to capture a time stamp. It must match to one of the following syntaxes:

The last pattern is the most preferred one, since it reflects a common use in US.

Example

Properties

Used in: DOC-REVISION p. 18, SW-CS-ENTRY p. 29

Content: Text
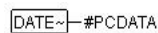
2.7 DEPARTMENT

Description

<DEPARTMENT> is used to capture the organizational unit of the team member in question.

Example

For an example, see Chapter 2.22 PROJECT-DATA p. 24

Properties

Used in: TEAM-MEMBER p. 38

Content: Text

DEPARTMENT—#PCDATA

2.8 DESC

Description

<DESC> represents a general but brief description of the object in question.

Example

Using **<DESC>** corresponding to *ANNOTATION* in *CDF* format would look like this:

Further example, see Chapter 2.22 PROJECT-DATA p. 24

Properties

Used in: PROJECT p. 23,   SW-INSTANCE-PROPS-VARIANT p. 33

Content: Text

DESC—#PCDATA

2.9 DOC-REVISION

Description

<DOC-REVISION> gets all information about one particular revision of the file.

Example

For an example, see Chapter 2.1 ADMIN-DATA p. 15.

Properties

Used in: DOC-REVISIONS p. 18

Content: REVISION-LABEL p. 25,   STATE p. 27,   TEAM-MEMBER-REF p. 38,   DATE p. 17,   MODIFICATIONS p. 22

DOC-REVISION
- ? REVISION-LABEL—#PCDATA
- ? STATE~—#PCDATA
- ? TEAM-MEMBER-REF~—#PCDATA
- ? DATE~—#PCDATA
- ? MODIFICATIONS—+ MODIFICATION~—CHANGE—#PCDATA / REASON—#PCDATA
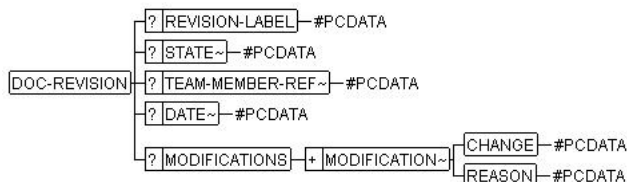
2.10 DOC-REVISIONS

Description

<DOC-REVISIONS> is a container to get the entire change-history. Although the sequence of changes can be determined by sorting **<DOC-REVISIONS>** according to **<DOC-REVISION>** /**<DATE>**, it is recommended to put the most recent **<DOC-REVISION>** as the first child of **<DOC-REVISIONS>**.

Example

For an example, see Chapter 2.1 ADMIN-DATA p. 15.

Properties

Used in: ADMIN-DATA p. 15

Content: DOC-REVISION p. 18
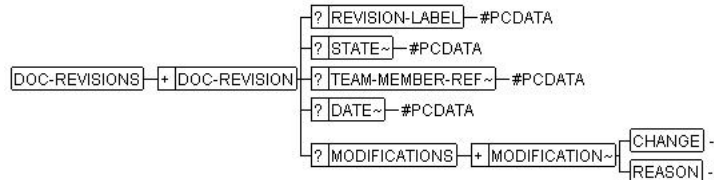
2.11   INTRODUCTION

Description

<INTRODUCTION> is used to provide a geneal introduction to the object in question. Although not requested by *CDF*, there are two options:

- Use **<P>** to allow the processing systems to perform a word wrapping.
- Use **<VERBATIM>** if whitespace is siginifcant.

According to the *CDF* a restriction to only one paragraph is allowed, even if *MSRSW.DTD* has a more powerful model.

Example

Properties

Used in: MSRSW p. 22,   SW-INSTANCE-SPEC p. 35

Content: P p. 23,   VERBATIM p. 39

2.12   LABEL

Description

<LABEL> is used as a long designator (similar to **<LONG-NAME>**) for objects which cannot be referenced. In this case it is the name of the corresponding project.

Example

For an example, see Chapter 2.22 PROJECT-DATA p. 24, where **<LABEL>** is the name of the corresponding project.

Properties

Used in: MATCHING-DCI p. 20,   PROJECT p. 23

Content: Text

`LABEL`—#PCDATA

2.13 LANGUAGE

Description

<LANGUAGE> represents the human language used within the file. It is mainly used to cause the tools to switch to an appropriate language.

This element follows ISO 639-1 two letter language codes (*[External Document: Codes for the Representation of Names of Languages / URL: http://lcweb.loc.gov/standards/iso639-2/termcodes.html]*). Mostly used codes are given in Table 2 Most common language codes (alphabetical) p. 20:

Table 2: Most common language codes (alphabetical)

Code	Language
de	German
en	English
es	Spanish
fr	French
it	Italian
jp	Japanese

Example

For an example, see Chapter 2.1 ADMIN-DATA p. 15.

Properties

Used in: ADMIN-DATA p. 15

Content: Text

`LANGUAGE`—#PCDATA

2.14 LONG-NAME

Description

<LONG-NAME> is used as a long designate for object such as companies, calibration Items.

Example

For an example, see Chapter 2.22 PROJECT-DATA p. 24.

Properties

Used in: COMPANY p. 16,   SW-INSTANCE   p. 33,   SW-INSTANCE-TREE   p. 35,
TEAM-MEMBER p. 38

Content: Text

`LONG-NAME`—#PCDATA

2.15 MATCHING-DCI

Description

<MATCHING-DCI> represents a reference to a *DCI* instance (Document Content Information) tho wich the current file is supposed to match to. The official identification of the DCI is denoted by **<URL>** which points to the DCI file. The other elements are provided for informational purposes.

Example:

This example correspond to the information given in *[External Document: Calibration Data File Specification / Date: 2001-07-19 / Publisher: Jeff Kainz / Relevant Position:]* .

Properties

Used in: MATCHING-DCIS p. 21

Content: LABEL p. 19,   SHORT-LABEL p. 26,   URL p. 39,   REMARK p. 25
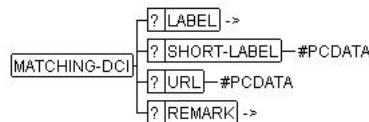
2.16 MATCHING-DCIS

Description

<MATCHING-DCIS> represents all Document Content Information specifications the current file is supposed to be compliant to. As there may be multiple DCI instances, the current file is supposed to match to all of them. Nevertheless the check is only requested according to the current use case.

In particular, in the CDF usecase, a *CDF processor* is obliged to check, if the CDF-DCI is mentioned within **<MATCHING-DCIS>**.

Example

In the following example the current file would match *CDF* as well as *PACO* which would support more of the features in *MSRSW.DTD*.

Properties

Used in: MSRSW p. 22

Content: MATCHING-DCI p. 20

07/08/2002 17:05:45 msrsw-tr-cdf.xml
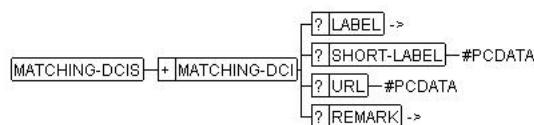
2.17 MODIFICATION

Description

<MODIFICATION> describes the modification of the file related to the predescessor.

Example

For an example, see .

Properties

Used in:
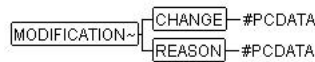
Content: ,

Attribute	Type	legal content	Remarks
[TYPE] (required)	namedtokengroup	• CONTENT-RELATED • DOC-RELATED	This attribute describes the level of the modification. In particular, this allows to specify, if the modification is relevant to the described systems.

2.18 MODIFICATIONS

Description

<MODIFICATIONS> represents the container which receives all **<MODIFICATION>**s.

Example

For an example, see .

Properties

Used in:

Content:

2.19 MSRSW

Description

<MSRSW> is the root element of the file. Mainly it consists of the following sections:

<SHORT-NAME> A short designator of the file. Usually this is something like filename body a project identifier.

\<CATEGORY\> This is denotes the category of the file. In case of CDF the content would be CDF. The category here mainly corresponds to **\<MATCHING-DCI\>**.

\<PROJECT-DATA\> This denotes all organisational and project relevant data.

\<ADMIN-DATA\> This denotes al administrative data related to the current file.

\<INTRODUCTION\> This is a general note that serves as an introduction to the current file.

\<SW-SYSTEMS\> This section receives the intended content of the CDF file. As an extension to *CDF* it supports multiple ECUs within one file.

\<MATCHING-DCIS\> This allows to refer to a formal specification which can be used to validate the current file.

Example

Properties

Used in: Root

Content: SHORT-NAME p. 26, CATEGORY p. 15, PROJECT-DATA p. 24, ADMIN-DATA p. 15, INTRODUCTION p. 19, SW-SYSTEMS p. 37, MATCHING-DCIS p. 21
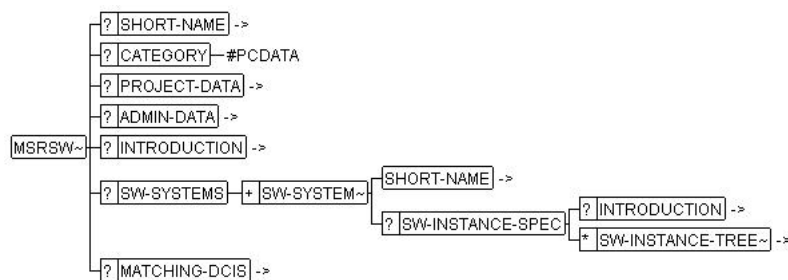
2.20 P

Description

\<P\> represents one particular paragraph.

Properties

Used in: INTRODUCTION p. 19, REMARK p. 25

Content: Text
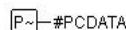
2.21 PROJECT

Description

Use **\<PROJECT\>** to capture information about the project the file belongs to.

Example

For an example, see Chapter 2.22 PROJECT-DATA p. 24.

Properties

Used in: PROJECT-DATA p. 24

Content: LABEL p. 19, DESC p. 18, COMPANIES p. 16
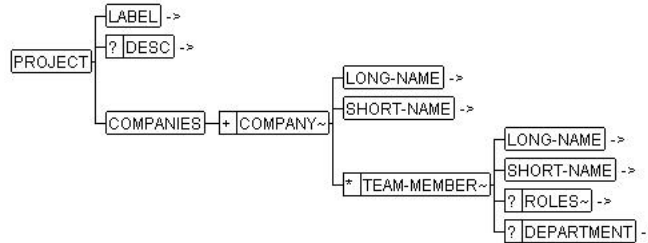
2.22 PROJECT-DATA

Description

Use **<PROJECT-DATA>** to capture information about the project the file belongs to. Although this wrapper elmenet seems to be superfluous in *CDF*, exists because more project related information is possible in *MSRSW.DTD*.

Example

Properties

Used in: MSRSW p. 22

Content: PROJECT p. 23
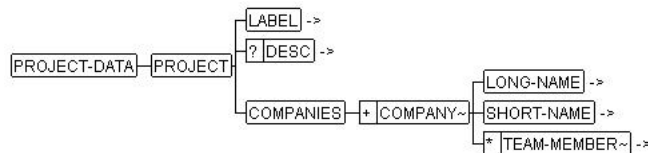
2.23 REASON

Description

Use **<REASON>** to specify the reason of a particular modification.

Example

For an example, see Chapter 2.1 ADMIN-DATA p. 15.

Properties

Used in: MODIFICATION p. 21

Content: Text

07/08/2002 17:05:45 msrsw-tr-cdf.xml
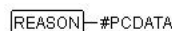
2.24 REMARK

Description

<REMARK> is used to give a comment e.g. for the particular calibration state. Although not reuqested by *CDF*, there are two options:

- Use **<P>** to allow the processing systems to perform a word wrapping.
- Use **<VERBATIM>** if whitespace is siginifcant.

According to the *CDF* a restriction to only one paragraph is allowed, even if *MSRSW.DTD* has a more powerful model.

Properties

Used in: MATCHING-DCI p. 20,   SW-CS-ENTRY p. 29

Content: P p. 23,   VERBATIM p. 39

2.25 REVISION-LABEL

Description

<REVISION-LABEL> is used to denote version number of the current file.

Example

For an example, see Chapter 2.1 ADMIN-DATA p. 15.

Properties

Used in: DOC-REVISION p. 18

Content: Text

2.26 ROLE

Description

<ROLE> denotes one particular role taken by the team member in question within the current project. Roles are such as "Author", "Calibration engineer", "Supporter", "Quality assurance".

Example

For an example, see Chapter 2.22 PROJECT-DATA p. 24.

Properties

Used in: ROLES p. 26

Content: Text

07/08/2002 17:05:45 msrsw-tr-cdf.xml

`ROLE─#PCDATA`

2.27 ROLES

Description

<ROLES> is a wrapper containing all roles one particular team member takes within the project.

Example

For an examaple, see Chapter 2.22 PROJECT-DATA p. 24

Properties

Used in: TEAM-MEMBER p. 38

Content: ROLE p. 25

`ROLES~─+ ROLE─#PCDATA`

2.28 SHORT-LABEL

Description

<SHORT-LABEL> represents a short designator (similar to **<SHORT-NAME>**) for objects which **cannot** be referred to.

Example

For an example, see Chapter 2.16 MATCHING-DCIS p. 21.

Properties

Used in: MATCHING-DCI p. 20

Content: Text

`SHORT-LABEL─#PCDATA`

2.29 SHORT-NAME

Description

<SHORT-NAME> represents a short designator for objects which **can** be referred to. In *CDF*, this mainly corresponds to *NAME*.

Example

For an example, see Chapter 2.45 SW-INSTANCE p. 33.

Properties

Used in: COMPANY p. 16,   MSRSW p. 22,   SW-INSTANCE p. 33,   SW-INSTANCE-TREE p. 35,   SW-SYSTEM p. 36,   TEAM-MEMBER p. 38

07/08/2002 17:05:45 msrsw-tr-cdf.xml

Content: Text

SHORT-NAME —#PCDATA

2.30 STATE

Description

<STATE> represents the current state of the current file according to the configuration management plan.

Example

Properties

Used in: DOC-REVISION p. 18

Content: Text

STATE~ —#PCDATA

Attribute	Type	Remarks
[SI] (implied)	cdata	This can be used to match to formal state definitions.

2.31 SW-AXIS-CONT

Description

<SW-AXIS-CONT> takes the contents of one particular axis (denoted by **<SW-AXIS-INDEX>**) of the current calibration item.

If there are more than one axis, then the axis with the index 1 is incrementing first. This means that the values are given in the following order:

v(1,1)	...	v(n,1)
v(1,2)	...	v(n,2)
...		
v(1,m)	...	v(n, m)

Example

For an example, see Chapter 2.45 SW-INSTANCE p. 33.

Properties

Used in: SW-AXIS-CONTS p. 28

Content: SW-UNIT-REF p. 37,   SW-AXIS-INDEX p. 28,   SW-VALUES-PHYS p. 38

07/08/2002 17:05:45 msrsw-tr-cdf.xml
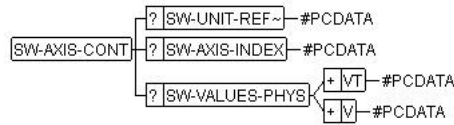
2.32   SW-AXIS-CONTS

Description

<SW-AXIS-CONTS> represents the contents of all axes of one particular calibration item.

Example

For an example, see Chapter 2.45 SW-INSTANCE p. 33.

Properties

  Used in: SW-INSTANCE-PROPS-VARIANT p. 33

  Content: SW-AXIS-CONT p. 27
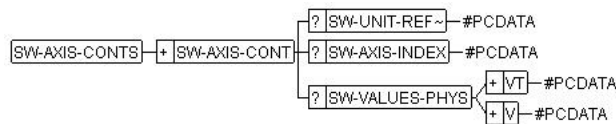
2.33   SW-AXIS-INDEX

Description

<SW-AXIS-INDEX> denotes the index of the current axis for which the contents is specified. The index uses the following convention:

  0: This is the value axis of a curve or a map

  1: This is the X-axis of a curve or a map.

  2: This is the Y-axis of a map

  3: This is the Z-axis of a cuboid ...

Example

For an example, see Chapter 2.45 SW-INSTANCE p. 33.

Properties

  Used in: SW-AXIS-CONT p. 27

  Content: Text

2.34   SW-CS-CONTEXT

Description

<SW-CS-CONTEXT> denotes the context in which the current calibration item was calibrated in order to achieve the calibration state in question. This context may e.g. refer to a test procedure (e.g. summer test) or a particular work package.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

```
SW-CS-CONTEXT—#PCDATA
```

2.35   SW-CS-DATA-IDENTIFIER

Description

<SW-CS-DATA-IDENTIFIER> refers to the data version in which the calibration item was set to its current contents.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

```
SW-CS-DATA-IDENTIFIER—#PCDATA
```

2.36   SW-CS-ENTRY

Description

<SW-CS-ENTRY> is one particular entry in the calibration history.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-HISTORY p. 30

Content: SW-CS-STATE   p. 32,   SW-CS-CONTEXT   p. 28,   SW-CS-PROJECT-INFO
p. 31,   SW-CS-TARGET-VARIANT   p. 32,   SW-CS-TEST-OBJECT   p. 33,
SW-CS-PROGRAM-IDENTIFIER   p. 31,   SW-CS-DATA-IDENTIFIER   p. 29,

07/08/2002 17:05:45 msrsw-tr-cdf.xml

```
                        ┌─? SW-CS-STATE ─ #PCDATA
                        ├─? SW-CS-CONTEXT ─ #PCDATA
                        ├─? SW-CS-PROJECT-INFO ─ #PCDATA
                        ├─? SW-CS-TARGET-VARIANT ─ #PCDATA
                        ├─? SW-CS-TEST-OBJECT ─ #PCDATA
SW-CS-ENTRY ────────────┼─? SW-CS-PROGRAM-IDENTIFIER ─ #PCDATA
                        ├─? SW-CS-DATA-IDENTIFIER ─ #PCDATA
                        ├─? SW-CS-PERFORMED-BY ─ #PCDATA
                        ├─? REMARK ->
                        ├─? DATE~ ─ #PCDATA
                        └─* SW-CS-FIELD~ ─ #PCDATA
```
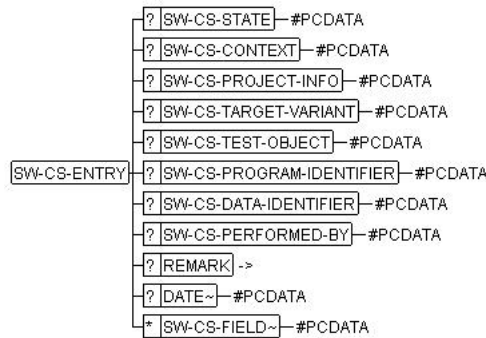
2.37 SW-CS-FIELD

Description

<SW-CS-FIELD> allows to add furhter information to the calibration history. This may be used to support specific use cases and procedures.

Example

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

```
SW-CS-FIELD~ ─ #PCDATA
```

Attribute	Type	Remarks
[SI] (required)	cdata	This attribute denotes the specific semantic of the **<SW-CS-FIELD>**.

2.38 SW-CS-HISTORY

Description

<SW-CS-HISTORY> allows to track the history of the calibration process for one particular calibration item as well as for the entire calibration list. The history therefore allows to keep more than one history entry.

Example

Properties

Used in: SW-INSTANCE-PROPS-VARIANT p. 33,   SW-INSTANCE-TREE p. 35

Content: SW-CS-ENTRY p. 29

07/08/2002 17:05:45 msrsw-tr-cdf.xml
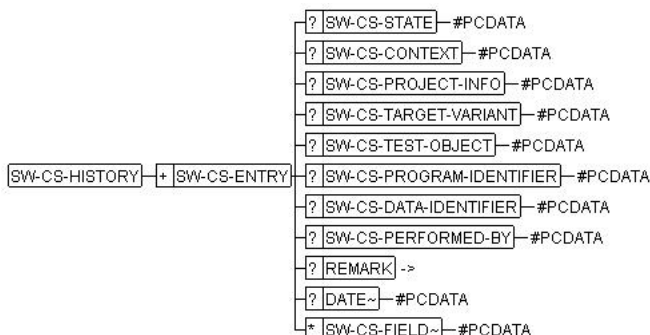
2.39 SW-CS-PERFORMED-BY

Description

<SW-CS-PERFORMED-BY> specifies the particular individual which assigned the current calibration state. Although this could be a reference to **<TEAM-MEMBER>**, it is simply a string. This is because data might be copied from project to project, which would make the formal reference invalid.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

2.40 SW-CS-PROGRAM-IDENTIFIER

Description

<SW-CS-PROGRAM-IDENTIFIER> denotes the identifier of the ECU program which was used while calibrating the current calibration item up to the state of the current calibration history.

To some extent, this should give the basis to determine the matching *SYMBOLIC-FILE* according to *CDF* specification.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

07/08/2002 17:05:45 msrsw-tr-cdf.xml

2.41 SW-CS-PROJECT-INFO

Description

<SW-CS-PROJECT-INFO> allows to denote a project identifier in order to specify the project in which the current calibration state was achieved. This makes sense, because, calibration items can be copied from one project to another. Therefore the information in **<PROJECT-DATA>** is not sufficient for this use case.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

SW-CS-PROJECT-INFO — #PCDATA

2.42 SW-CS-STATE

Description

<SW-CS-STATE> is used to denote the current state of the calibration item. The legal values must be defined within the process.

Example

For a further example, see SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

SW-CS-STATE — #PCDATA

2.43 SW-CS-TARGET-VARIANT

Description

<SW-CS-TARGET-VARIANT> is used to specify the variant of the target system which was used to achieve the current calibration state.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

SW-CS-TARGET-VARIANT—#PCDATA

2.44 SW-CS-TEST-OBJECT

Description

<SW-CS-TEST-OBJECT> denotes the object witin the the current calibration item was calibrated. This is mainly a vehicle or a particular test environment.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-CS-ENTRY p. 29

Content: Text

SW-CS-TEST-OBJECT—#PCDATA

2.45 SW-INSTANCE

Description

<SW-INSTANCE> represents one particular calibration item and therefore corresponds to *CAL-ITEM* in CDF.

The model of **<SW-INSTANCE>** and its children is more complex than the one in *CDF* because of the broader set of use cases proposed for *ASAM-MCD-2MC 2.0*, e.g. with respect of variant coding.

Example

Properties

Used in: SW-INSTANCE-TREE p. 35

Content: LONG-NAME p. 20, SHORT-NAME p. 26, CATEGORY p. 15, SW-INSTANCE-PROPS-VA
p. 34

07/08/2002 17:05:45 msrsw-tr-cdf.xml
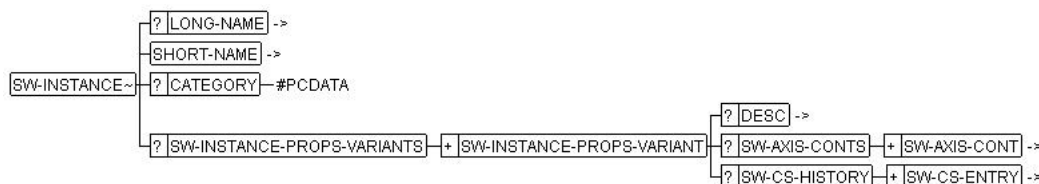
2.46 SW-INSTANCE-PROPS-VARIANT

Description

<SW-INSTANCE-PRPS-VARIANT> describes one particular variant of the current calibration item. Since *CDF* did not request this, there is one and only one variant per calibration item which can be specified.

Example

For an example, see Chapter 2.38 SW-CS-HISTORY p. 30.

Properties

Used in: SW-INSTANCE-PROPS-VARIANTS p. 34

Content: DESC p. 18,   SW-AXIS-CONTS p. 28,   SW-CS-HISTORY p. 30
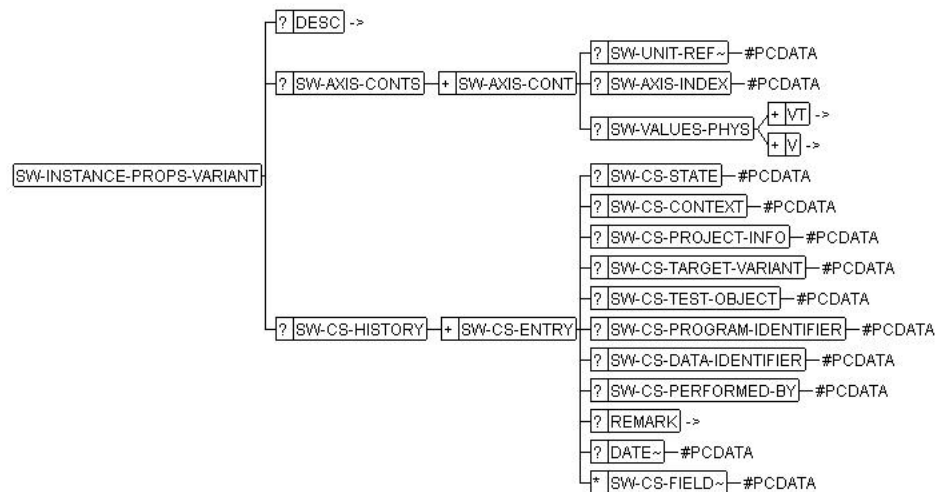
2.47 SW-INSTANCE-PROPS-VARIANTS

Description

<SW-INSTANCE-PROPS-VARIANTS> is the container which takes all variants of one particular calibration item.

Example

For an example, see Chapter 2.45 SW-INSTANCE p. 33.

Properties

Used in: SW-INSTANCE p. 33

Content: SW-INSTANCE-PROPS-VARIANT p. 33

07/08/2002 17:05:45 msrsw-tr-cdf.xml

```
                                            ┌─┐
                                            │?│ DESC ->
                                            └─┘
                                                              ┌─┐
                                                              │?│ SW-UNIT-REF~ ->
                                            ┌─┐  ┌─┐           └─┘
                                            │?│ SW-AXIS-CONTS ─┤+│ SW-AXIS-CONT │?│ SW-AXIS-INDEX ->
                                            └─┘  └─┘           ┌─┐
                                                              │ │ SW-VALUES-PHYS ->
                                                              └─┘
  ┌───────────────────────┐  ┌─┐  ┌──────────────────────┐
  │SW-INSTANCE-PROPS-VARIANTS│─┤+│ │SW-INSTANCE-PROPS-VARIANT│
  └───────────────────────┘  └─┘  └──────────────────────┘
                                                              ┌─┐
                                                              │?│ SW-CS-STATE ->
                                                              ┌─┐
                                                              │?│ SW-CS-CONTEXT ->
                                                              ┌─┐
                                                              │?│ SW-CS-PROJECT-INFO ->
                                                              ┌─┐
                                                              │?│ SW-CS-TARGET-VARIANT ->
                                                              ┌─┐
                                                              │?│ SW-CS-TEST-OBJECT ->
                                            ┌─┐  ┌─┐          ┌─┐
                                            │?│ SW-CS-HISTORY ─┤+│ SW-CS-ENTRY │?│ SW-CS-PROGRAM-IDENTIFIER ->
                                            └─┘  └─┘          ┌─┐
                                                              │?│ SW-CS-DATA-IDENTIFIER ->
                                                              ┌─┐
                                                              │?│ SW-CS-PERFORMED-BY ->
                                                              ┌─┐
                                                              │?│ REMARK ->
                                                              ┌─┐
                                                              │?│ DATE~ ->
                                                              ┌─┐
                                                              │*│ SW-CS-FIELD~ ->
                                                              └─┘
```

2.48   SW-INSTANCE-SPEC

Description

<SW-INSTANCE-SPEC> is the section within a **<SW-SYSTEM>** which takes all information about the instances of data items within the ECU, in particular the calibration items.

As **<SW-SYSTEM>** does not allow to place a note or an introduction, **<INTRODUCTION>** of **<SW-INSTANCE-SPEC>** corresponds to *NOTE* in *CAL-LIST* specified by *CDF*.

Example:

Properties

Used in: SW-SYSTEM p. 36

Content: INTRODUCTION p. 19,   SW-INSTANCE-TREE p. 35

```
                    ┌─┐
                    │?│ INTRODUCTION ->
                    └─┘
                                    ┌─┐
                                    │?│ LONG-NAME ->
                                    ┌─────────┐
                                    │SHORT-NAME│ ->
                                    ┌─┐        ┌──────┐
                                    │?│ CATEGORY│#PCDATA│
                                    └─┘        └──────┘
  ┌────────────────┐                                      ┌─┐
  │SW-INSTANCE-SPEC │                                      │?│ SW-CS-STATE ->
  └────────────────┘                                      ┌─┐
                                                          │?│ SW-CS-CONTEXT ->
                                                          ┌─┐
                                                          │?│ SW-CS-PROJECT-INFO ->
                                                          ┌─┐
                                                          │?│ SW-CS-TARGET-VARIANT ->
                                                          ┌─┐
                                                          │?│ SW-CS-TEST-OBJECT ->
                    ┌─┐                ┌─┐  ┌─┐            ┌─┐
                    │*│ SW-INSTANCE-TREE~ │?│ SW-CS-HISTORY ─┤+│ SW-CS-ENTRY │?│ SW-CS-PROGRAM-IDENTIFIER ->
                    └─┘                └─┘  └─┘            ┌─┐
                                                          │?│ SW-CS-DATA-IDENTIFIER ->
                                                          ┌─┐
                                                          │?│ SW-CS-PERFORMED-BY ->
                                                          ┌─┐
                                                          │?│ REMARK ->
                                                          ┌─┐
                                                          │?│ DATE~ ->
                                                          ┌─┐
                                                          │*│ SW-CS-FIELD~ ->
                                                          └─┘
                                    ┌─┐
                                    │?│ LONG-NAME ->
                                    ┌─────────┐
                                    │SHORT-NAME│ ->
                    ┌─┐             ┌─┐        ┌──────┐
                    │*│ SW-INSTANCE~ │?│ CATEGORY│#PCDATA│
                    └─┘             └─┘        └──────┘
                                    ┌─┐                      ┌─┐
                                    │?│ SW-INSTANCE-PROPS-VARIANTS │+│ SW-INSTANCE-PROPS-VARIANT ->
                                    └─┘                      └─┘
```

07/08/2002 17:05:45 msrsw-tr-cdf.xml

2.49 SW-INSTANCE-TREE

Description

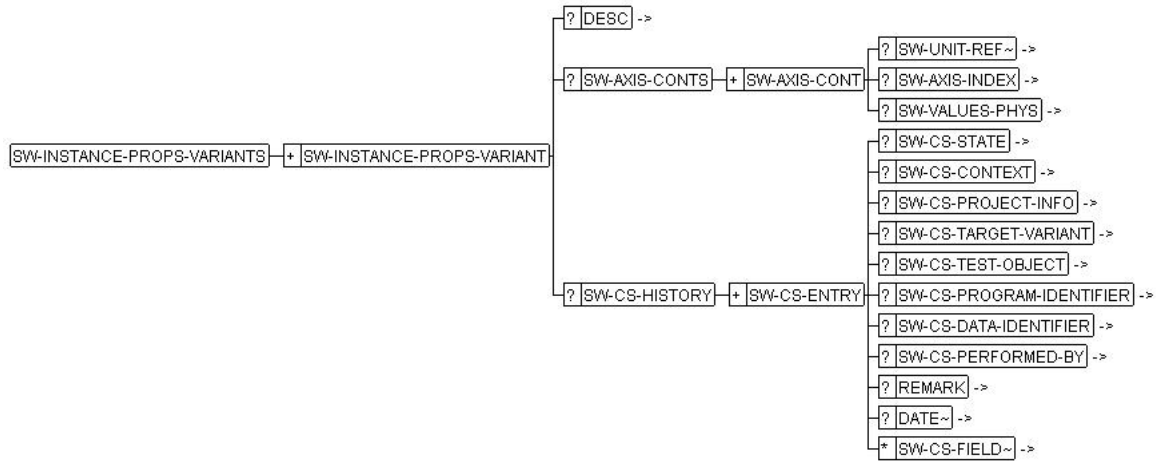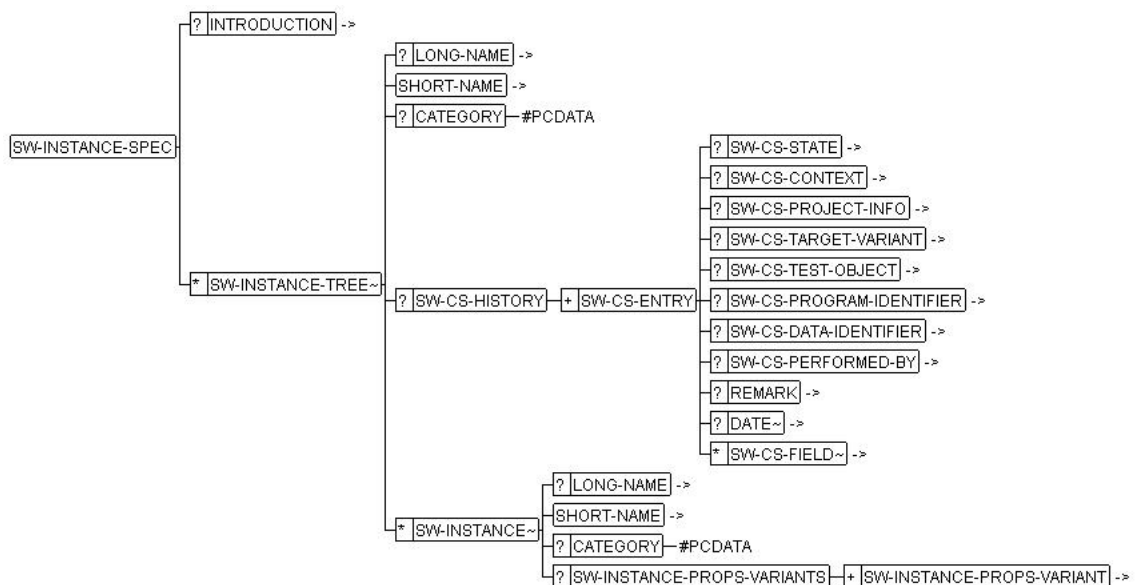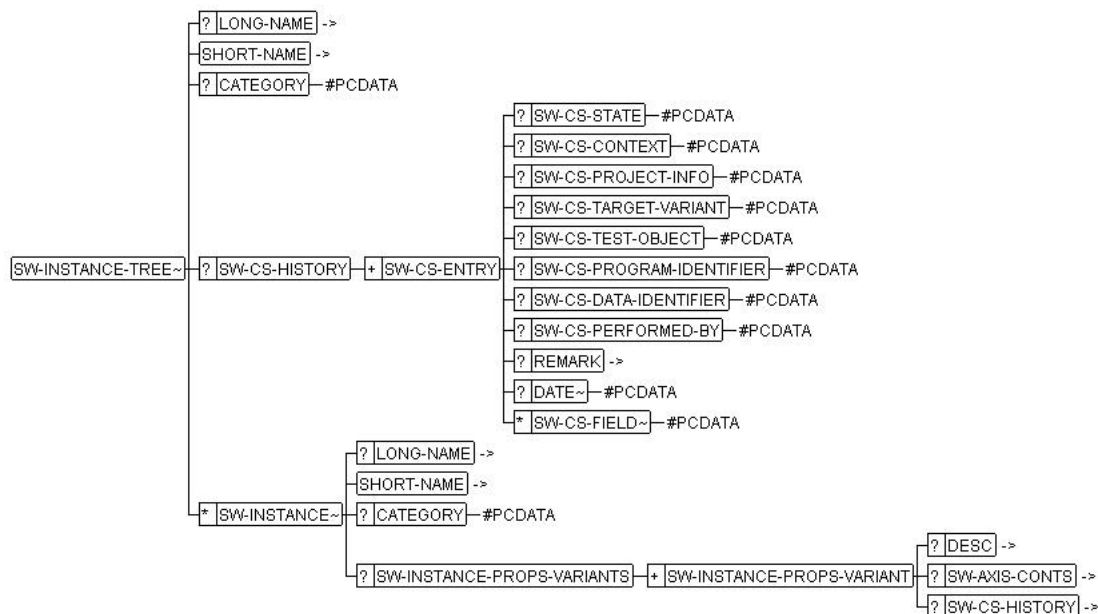**<SW-INSTANCE-TREE>** receives one particular set of calibration items. Therefore it corresponds to *CAL-LIST* within *CDF* specification.

Example

Properties

Used in: SW-INSTANCE-SPEC p. 35

Content: LONG-NAME p. 20,   SHORT-NAME p. 26,   CATEGORY p. 15,   SW-CS-HISTORY p. 30,   SW-INSTANCE p. 33

2.50 SW-SYSTEM

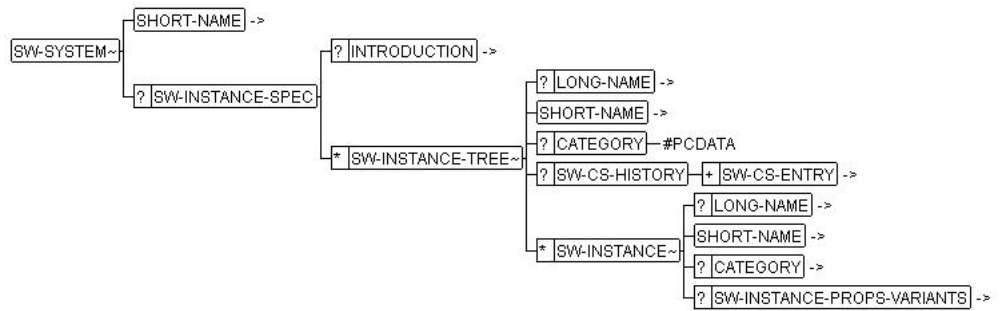**Description**

<SW-SYSTEM> describes one particular system within the current file. This supports to transfer the calibration data of multi ecu systems within one file.

Example

Properties

Used in: SW-SYSTEMS p. 37

Content: SHORT-NAME p. 26,   SW-INSTANCE-SPEC p. 35
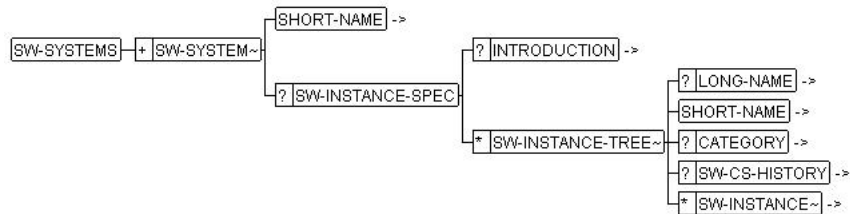
2.51 SW-SYSTEMS

Description

<SW-SYSTEMS> is the wrapper to recieve al **<SW-SYSTEM>** in the current file.

Example

Properties

Used in: MSRSW p. 22

Content: SW-SYSTEM p. 36

2.52 SW-UNIT-REF

Description

<SW-UNIT-REF> is a formal reference to a measurement unit. Since *CDF* did not request this, the reference is treated as a measurement unit designator.

Caution:
The content of **<SW-UNIT-REF>** must follow the restrictions of **<SHORT-NAME>**, in particular must not contain whitespace and special characters.

Example

For an example, see Chapter 2.45 SW-INSTANCE p. 33.

Properties

Used in: SW-AXIS-CONT p. 27

Content: Text

SW-UNIT-REF~—#PCDATA

2.53 SW-VALUES-PHYS

Description

<SW-VALUES-PHYS> receives the physical values of the calibration item. Although there may be other instance specific properties, *CDF* requested physical values only.

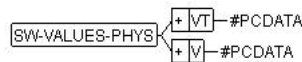According to *CDF* specification, the values can either be numeric (**<V>**) or textual (**<VT>**) but not mixed.

Example

For an example, see Chapter 2.45 SW-INSTANCE p. 33.

Properties

Used in: SW-AXIS-CONT p. 27

Content: VT p. 40,   V p. 39

SW-VALUES-PHYS — + VT — #PCDATA
             — + V — #PCDATA

2.54 TEAM-MEMBER

Description

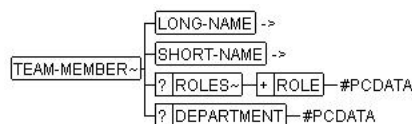**<TEAM-MEMBER>** is the information about one particular indivdual being part of the project team.

Example

For an example, see Chapter 2.22 PROJECT-DATA p. 24.

Properties

Used in: COMPANY p. 16

Content: LONG-NAME p. 20,   SHORT-NAME p. 26,   ROLES p. 26,   DEPARTMENT p. 17

TEAM-MEMBER~ — LONG-NAME ->
            — SHORT-NAME ->
            — ? ROLES~ — + ROLE — #PCDATA
            — ? DEPARTMENT — #PCDATA

2.55 TEAM-MEMBER-REF

Description

<TEAM-MEMBER-REF> is the pointer to one particular team member. The content is the **<SHORT-NAME>** of the corresponding **<TEAM-MEMBER>**.

Example

For an example, see Chapter 2.1 ADMIN-DATA p. 15.

Properties

Used in: DOC-REVISION p. 18

Content: Text

```
TEAM-MEMBER-REF~──#PCDATA
```

2.56   URL

Description

<URL> is the uniform resource locator, e.g. of the file with the document content information.

Example

For an example, see Chapter 2.16 MATCHING-DCIS p. 21.

Properties

Used in: MATCHING-DCI p. 20

Content: Text

```
URL──#PCDATA
```

2.57   V

Description

<V> represents one particular numerical value of the the calibration item.

Example

For an example, see Chapter 2.45 SW-INSTANCE p. 33.

Properties

Used in: SW-VALUES-PHYS p. 38

Content: Text

```
V──#PCDATA
```

2.58   VERBATIM

Description

<VERBATIM> is a paragraph in which whitespace (in particular blanks and line feeds) is obeyed. This allows to perform basic preformatting which can be displayed on even dumb devices. The behavior is the same as PRE in *HTML*.

07/08/2002 17:05:45 msrsw-tr-cdf.xml

Example

Properties

Used in: INTRODUCTION p. 19,   REMARK p. 25

Content: Text

VERBATIM~ — #PCDATA

2.59   VT

Description

<VT> represents one particular textual value of the the calibration item.

Example

Properties

Used in: SW-VALUES-PHYS p. 38

Content: Text

VT — #PCDATA

Documentadministration

Table : team members

Name	Company	
Bernhard Weichel	MSR-MEDOC/ASAM-MCD Working Group	Phone: +(49) 711 1389969 Department: Robert Bosch GmbH, GS/EMW4 berhard.weichel@de.bosch.com

Table : version overview

Version	Date	Publisher	State
2	2002-06-02		WD
1	2001-08-15	Bernhard Weichel	CD

Table : modifications

Version	Change	Related to
2	Migration to XML Reason: -	Document
1	Created the document Reason: According to video-conference held at Jul 2001	Document

Table : modifications included

Date	Chapter	Change	Related to
Nr. 1, 2002-06-02	Gesamt	Migration to XML Reason: -	Document
Nr. 2, 2001-08-15	Gesamt	Created the document Reason: According to video-conference held at Jul 2001	Document