



Keplerlaan 1, 2200 AG Noordwijk, The Netherlands
+31-71-5656565

DOCUMENT

Handbook for EO XML and Binary Schemas

PE-TN-ESA-GS-121 Handbook for EO XML and Binary Schemas 1.7.1

Prepared by Michele Zundo
Reference PE-TN-ESA-GS-121
Issue 1.7.1
Revision
Date of Issue 24-September-2015
Status Final
Document Type Technical Note
Distribution



APPROVAL

Title Handbook for EO XML and Binary Schemas	
Issue 1.7.1	Revision
Author Michele Zundo/Patrick Deghaye. EOP-PEP	Date 24-September-2015
Approved by Pierre Viau, EOP-PE	Date 24-September-2015

CHANGE LOG

Reason for change	Issue	Revision	Date
First public issue	1.3		18-Jan-2007
Changed Tile and clarified versioning	1.4		27-Jul-2007
General clean-up, inclusion of more BinaryXML languages, removal of SMOS specific sections	1.5		03-Nov-2011
First issue new ESA doc template	1.5.1		14-Nov-2011
Clarified syntax of schema reference, updated reference to DFDL language plus other minor clean-ups	1.6		01-May-2012
Aligned with EO FFS V3.0 and consequently the schema file naming convention, prescribes usage of DFDL for binary data schema, plus several editorial updates.	1.7		24-September-2015
Revised use of nonamespace in Section 7, 8.	1.7.1		24-September-2015

CHANGE RECORD

Issue	Revision		
Reason for change	Date	Pages	Paragraph(s)
1.3			
First public issue	18-Jan-2007	all	

Issue	Revision		
Reason for change	Date	Pages	Paragraph(s)
1.4			
Changed Tile and clarified versioning	27-Jul-2007	all	

Issue 1.5	Revision		
Reason for change	Date	Pages	Paragraph(s)
General clean-up, inclusion of more BinaryXML languages, removal of SMOS specific sections	03-Nov-2011	all	

Issue 1.5.1	Revision		
Reason for change	Date	Pages	Paragraph(s)
First issue new ESA doc template	14-Nov-2011	all	
Update Fig. 8	14-Nov-2011	Fig. 8	

Issue 1.6	Revision		
Reason for change	Date	Pages	Paragraph(s)
Correction of Fig. 4 and 5	01-May-2012	9, 11	
Expanded description of possible binary schema (XML vs XSD) to cope with DFDL binary schemas.	01-May-2012	9,10	Section 4.2.2
Updated section on BinX	01-May-2012	10	Section 4.2.2.1
Updated section on DFDL	01-May-2012	10	Section 4.2.2.3
Expanded description to allow both XML vs XSD filename extensions	01-May-2012	15,16	Section 6.2
Added explicit syntax of attributes for schema reference in both schema and instance file	01-May-2012	17,18	Section 7
Updated Figure 7 with a more recent example	01-May-2012	19	Section 8

Issue 1.7.1	Revision		
Reason for change	Date	Pages	Paragraph(s)
Aligned with EO FFS V3.0 allowing multiple Data Block files with relaxed definition of extension.	17-Apr-2015	All	
Updated List of Reference Document.	17-Apr-2015	3	
Prescribes usage of DFDL for binary data schema. Removed former references to BinX and DRB XML syntax and other languages.	17-Apr-2015	4.2.2, 7.2	
Updated Schema file naming to cope with multiple Data Block files and relaxed definition of extension.	17-Apr-2015	6	
Several editorial updates.	17-Apr-2015	All	
Updated section 7 and 8 to allow nonnamespace in schema references	24-Sep-2015	19	7.3, 8



Table of contents:

1	Purpose and Scope	5
2	Applicable documents	5
3	Reference documents	5
4	Introduction	6
4.1	Data Files.....	6
4.2	Schema Files.....	7
4.2.1	Schemas for XML data.	9
4.2.2	Schemas for binary data	9
5	Schema File Structure	12
5.1	Schemas for XML Files	12
5.1.1	Separated Header file (.HDR) and Data Block file (<DBL_Ext>).....	13
5.1.2	Single .EOF file composed by an Header and a Data Block enclosed in <Earth_Observation_File> tag	13
5.2	Schemas for Binary Files	14
6	File Naming	15
6.1	Schema for XML files	15
6.1.1	Top level schema file in the hierarchy	15
6.1.2	Sub component in hierarchy	16
6.2	Schema for Binary Files	16
6.2.1	Top level schema file in the hierarchy	16
6.2.2	Sub component in hierarchy	17
7	File Syntax	18
7.1	Schema for XML Syntax	18
7.2	Schema for Binary Syntax.....	18
7.2.1	DFDL case	19
7.3	Reference to Schemas	19
8	Schemas Versioning	21
8.1	XML Files Schema Versioning	21
8.1.1	Schema Version Identification	21
8.1.2	Convention for Schema Versioning	23



1 PURPOSE AND SCOPE

Purpose of this Handbook is to complement the scope of [EO_FFS] to giving consistent guidelines for the preparation, naming, definition and reference of both XML Schemas and Binary Schemas (regardless of which one is selected).

This Handbook presents a standardised way to name and use schema references within Earth Observation files, which are compliant with [EO_FFS]. It includes also specific examples from the EO CFI libraries files (orbit and attitude) which are using this standard approach to format definition.

NB: To be used by a Project this Handbook should be tailored, in way similar to what is done for [EO_FFS] to identify any choice and deviation.

The [EO_FFS] refers to the information contained in this Handbook as recommendation for schema usage and definition.

2 APPLICABLE DOCUMENTS

[EO_FFS] Earth Observation Ground Segment File Format Standard, PE-TN-ESA-GS-0001, issue 2.0 (03-May-2012) and issue 3.0

3 REFERENCE DOCUMENTS

[W3C_SD]	XML Schema Part 0, Part 1 and Part 2. Second Edition, W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-0/ , http://www.w3.org/TR/xmlschema-1/ , http://www.w3.org/TR/xmlschema-2/
[XML_VERS]	XML Versioning from xml-dev list newsgroup : (http://www.xfront.com/BestPracticesHomepage.html)
[DFDL_SPEC]	Data Format Description Language (DFDL) v1.0 Specification, GFD-P-R.207, September 2014, http://www.ggf.org/dfdl/
[DFDL4S_DM]	DFDL4S Developer Manual, S2G-DME-TEC-SUM078 , 1.C, 06/03/2015 http://eop-cfi.esa.int/index.php/docs-and-mission-data/applications-docs?jsmallfib=1&dir=JSROOT/DFDL4S
[DAFFODIL]	DAFFODIL Open Source DFDL Parser : https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Daffodil%3A+Open+Source+DFDL
[SAFE_SPEC]	SAFE 2.0 Specification: <ul style="list-style-type: none"> - Standard Archive Format for Europe – Control Book – Volume 1 – Core Specifications - 2.3 ; PGSI-GSEG-EOPG-FS-05-0001 - Standard Archive Format for Europe – Control Book – Volume 2 – Recommendations for Specialisations - 2.3 ; PGSI-GSEG-EOPG-FS-05-0002 (http://earth.esa.int/SAFE ; Not updated for SAFE 2.0)

4 INTRODUCTION

4.1 Data Files

The [EO_FFS] defines simple and clear conventions and standards on how data files have to be structured and what their encoding, naming and syntax has to be. The [EO_FFS] standard is used in all recent ESA Earth Observations missions and proved to be a solid practical solutions covering the complete Ground Segments needs including Mission Planning, Orbital determination, Interface Handling and Data Products.

EO Files are defined as composed by one Header and one or several Data Blocks; Header is specified to be an XML file while a Data Block can be either an ASCII/XML file and either a User-defined binary file or a Standard/Commercial binary file. Detailed explanation is found in [EO_FFS].

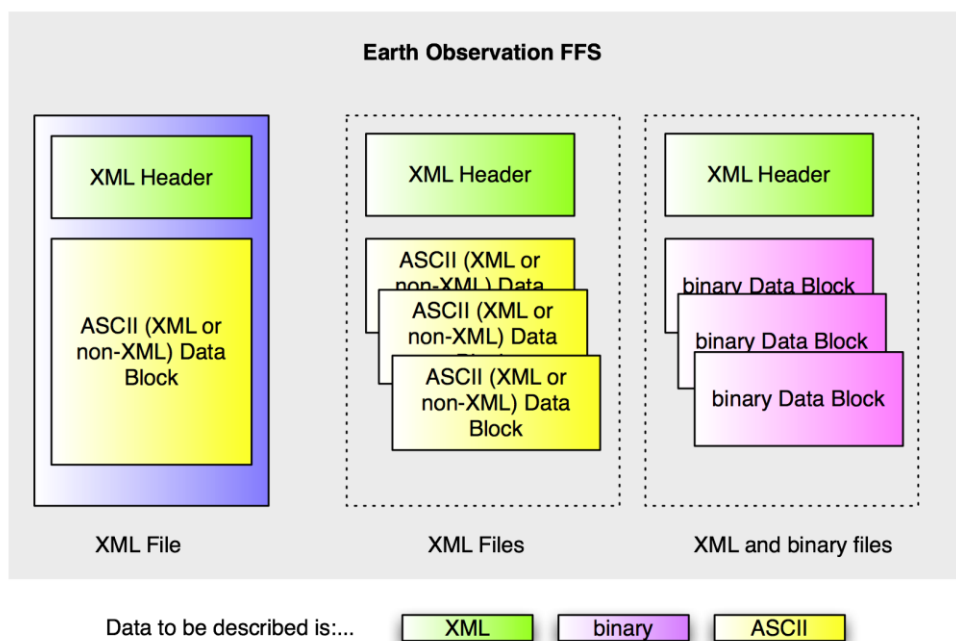


Fig. 1 EO Files structure/organisation

Note:

A Data Block could also be formatted as a self-describing format (e.g. netCDF or HDF5). This is not in conflict with the [EO_FFS] and is fully compatible with it and with the standardisation of Schemas described in this handbook which recommends a purely descriptive (rather than prescriptive) approach for binary data.

4.2 Schema Files

[EO_FFS] requires the use of Schemas but does not define the rules and the syntax to ensure consistent/common Schema usage. This handbook complements the [EO_FFS] by addressing this issue, as well as by defining a mechanism for format versioning by means of Schemas..

A Schema file is defined as a separate file used to describe the structure and syntax of a XML and (as a non-W3C extension) or of binary files/data. Both cases hereafter will be referred simply as "*Schema*". A Schema might also include information to validate the fields in the instance file defining constraints on the values that an element within the described file can have e.g. Element <Power> can only have value ON or OFF and element <Temperature> must be a float with 3 decimal digits.

Schemas shall mandatorily be defined for any XML part (Header or Data Block) while are optional (even though recommended) for binary (Data Block only) parts.

The Schema describing a specific (file) data structure and syntax consists of one or multiple files structured hierarchically (see example in Fig. 2 and 3) with basic types definition separated in dedicated Schema files.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://eop-cfi.esa.int/CFI"          targetNamespace="http://eop-cfi.esa.int/CFI"
  elementFormDefault="qualified"          attributeFormDefault="unqualified"
  version="2.2">
  <xs:include schemaLocation="EO_OPER_MPL_ORBPRES.HeaderTypes_0202.XSD"></xs:include>
  <xs:include schemaLocation="EO_OPER_MPL_ORBPRES.DataBlockTypes_0201.XSD"></xs:include>
  <xs:annotation>
    <xs:documentation>Predicted Orbit File</xs:documentation>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation>$Revision: 1.1 $ $Date: 2014-03-03 11:22:53 $</xs:documentation>
  </xs:annotation>
  <xs:complexType name="Predicted_Orbit_File_Type">
    <xs:sequence>
      <xs:element name="Earth_Explorer_Header" type="Predicted_Orbit_Header_Type"></xs:element>
      <xs:element name="Data_Block" type="Predicted_Orbit_Data_Block_Type"></xs:element>
    </xs:sequence>
    <xs:attribute name="schemaVersion" type="xs:decimal" use="required"></xs:attribute>
  </xs:complexType>
  <xs:element name="Earth_Explorer_File" type="Predicted_Orbit_File_Type"></xs:element>
</xs:schema>
```

Fig. 2 Example of the EO_OPER_MPL_ORBPRES.xsd Schema associated to MPL_ORBPRES.EOF file (XML File)

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:dmx="http://www.deimos.com.pt/dmx/dmx-1.0"
  xmlns:dfdl="http://www.ogf.org/dfdl/dfdl-1.0/"
  version="1.1">
  <xs:annotation>
    <xs:documentation>$Revision: 532 $ $Date:: 2012-03-27 18:03:55#</xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="Sentinel12X-bandTMTFTypes.xsd"/>
  <xs:annotation>
    <xs:appinfo source="http://www.ogf.org/dfdl/dfdl/">
      <dfdl:format byteOrder="bigEndian" encoding="utf-8" />
    </xs:appinfo>
  </xs:annotation>
  <xs:element name="Transfer_Frame" type="TypeTF" />
</xs:schema>

```

Fig. 3 Example of a DFDL Schema for a binary file

The use of multiple Schema files is recommended, so that the data structures and the related data types can be described modularly in separate files allowing the reuse of common blocks (e.g. date structure or data types) and reducing the complexity of big, nested data definitions. In the examples above the use of `xs:include` statement allows reuse of lower level Schemas containing basic type definitions. The corresponding logical representations of the Schema hierarchy are shown in the 2 examples below.

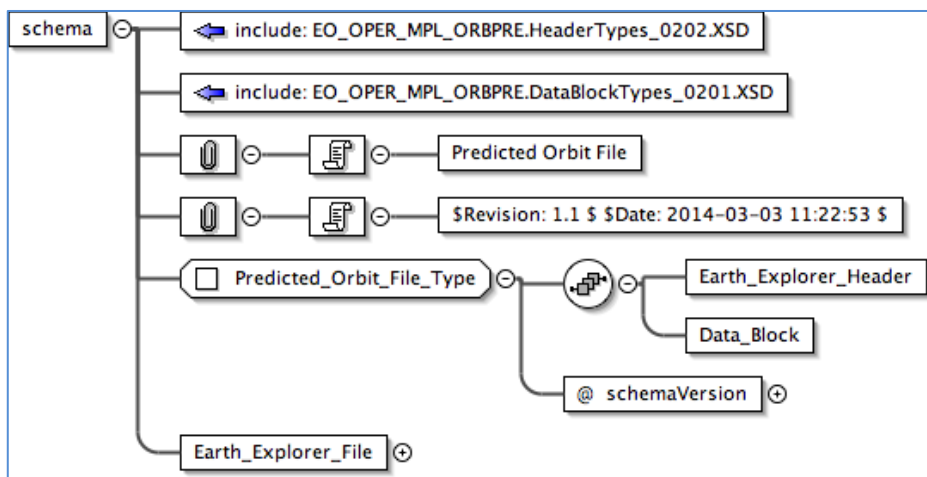


Fig. 4 Example of hierarchy in the Schema structure for MPL_ORBPRES.XSD file

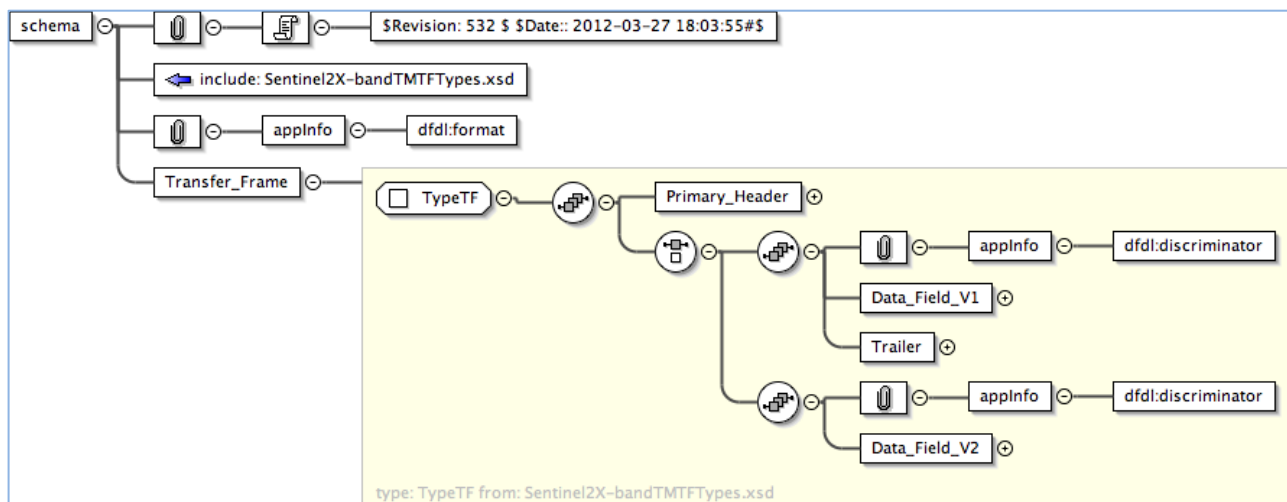


Fig. 5 Example of hierarchy in the DFDL Schema structure for a binary file with type included from a separate file (yellow box)

4.2.1 Schemas for XML data.

The XML Standards from W3C <http://www.w3.org/> fully cover Schemas for XML files (XSD files) in [W3C_SD].

Building on that this handbook recommends a standardised form of schema declaration, reference, namespace use and versioning to be used coherently for all Schemas.

4.2.2 Schemas for binary data

XML Standards from W3C <http://www.w3.org/> do not cater for binary data definitions, which are outside the scope of the XML.

Dictionaries (and associated libraries), which mimic, for binary data, the relationship between XML instances and Schemas (XSD files), have been developed within both proprietary and public domains. The recommended language to describe binary data for ESA EO mission is DFDL as described in [DFDL_SPEC] which is an open standard with available tool and a natural extension (to describe the encoding) of a XSD fully compatible with W3C standard schema which describes instead the logical data structure. This handbook assumes all the description of binary data is made using DFDL syntax.

DFDL allows the description of the structure/format of binary files/data by mean of an XSD standardised definition extended with a set of XML attributes which is a purely descriptive approach in the sense that the format of the binary data itself is left completely user defined and not constrained by the description language. Use of DFDL Schemas makes sense for any User-Defined binary data while it is optional for Commercial/Standard binary formats (e.g. jpg) as these formats are normally described and controlled via other mechanisms.

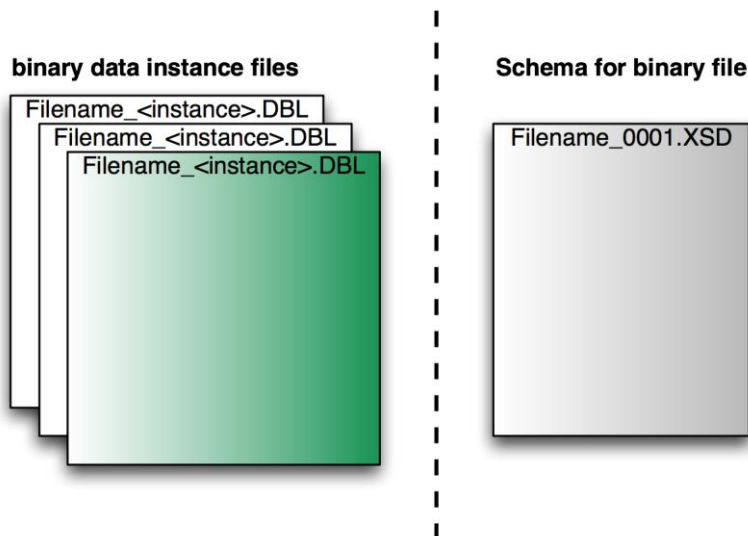


Fig. 6 Binary data entities (e.g. .DBL files) and associated schemas

DFDL and the corresponding software libraries (e.g. DFDL4S or DAFFODIL) allow the description of binary data structure, the verification of the values of each field and standardised serialisation in read and write mode.

Note that a standard XML parser is not able to process binary data or any corresponding “schema” definition and therefore there are dedicated libraries as mentioned above, playing a role similar to an XML parser like *Xerces* or *libxml*, allowing the user to access (read and write) data inside the binary data by mean of symbolic references contained in the XSD files which function as a schema.

The files we are dealing with are therefore:

- One (or multiple) binary data instance file(s) containing the data.
- One corresponding DFDL compliant XSD “schema” file (called *Schema for binary*)

Finally note that Schema files for binary data are not structure-validating schemas (since binary encoding is not fixed) and are only used by the ad-hoc software libraries to read and write compliant data structures within files and to verify data values (assuming the correct encoding)

4.2.2.1 The DFDL language

The Data Format and Description Language (DFDL) is a dictionary, standardised by the Open Grid Forum, to use within a W3C standard XSD schema XSD attributes to describe the formatting/encoding of binary data. It presents the considerable advantage that the binary schema is a fully W3C compliant schema which describes the encoding by means of dedicated XML attributes and can exploit the flexible and rich XSD constructs (e.g. choice constructs, XPath, etc).

DFDL is actively being maintained and evolving by the OGF community and it is the recommended descriptive binary schema language for future applications. An example of its usage is in the ESA’s S2G data visualisation tool (<http://eop-cfi.esa.int/S2G/s2g.html>), which uses a subset to describe CCSDS data file containing satellite telemetry.



DFDL is also the prescribed binary description language by SAFE version 2.0 [SAFE_SPEC], the archival format for ESA Earth Observation data.

See [DFDL_SPEC] [DFDL_SPEC] and <http://www.ggf.org/dfdl/>

5 SCHEMA FILE STRUCTURE

This section describes the conventions to be used when defining schemas for different type of data

5.1 Schemas for XML Files

Schema files for XML data shall be constituted by one or multiple XML ASCII files. (see Fig. 5) as per [W3C_SD].

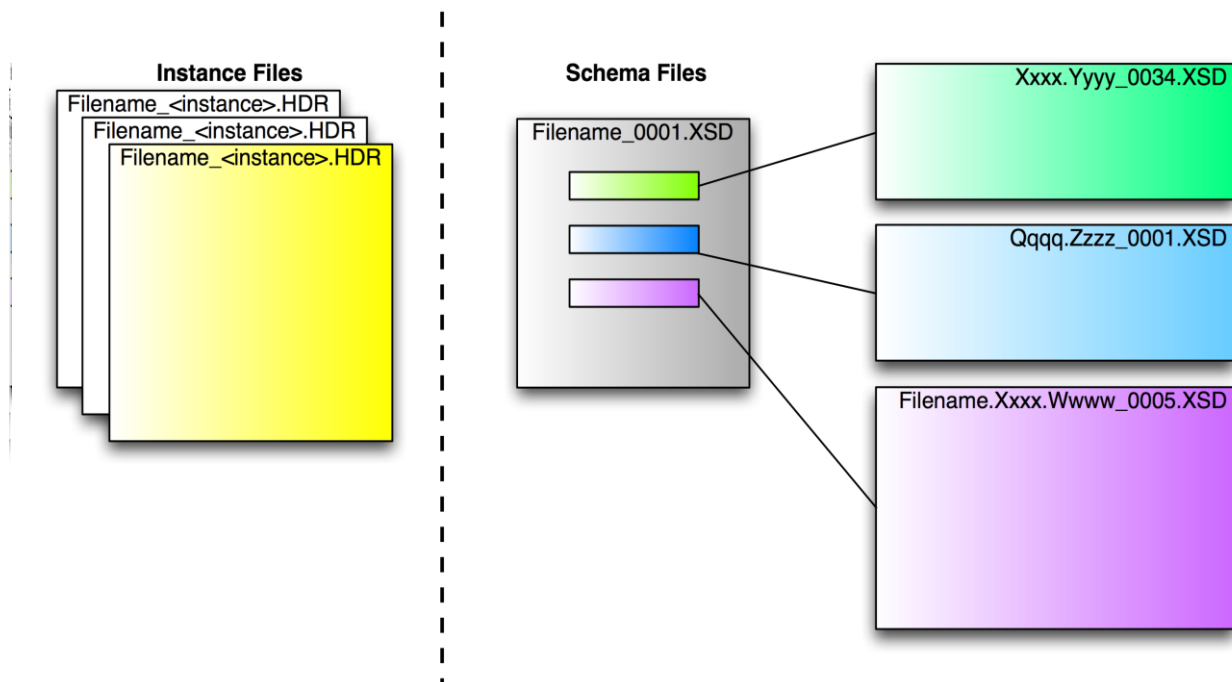


Fig. 7 Example of Schemas for pure XML files , e.g. for .HDR

[W3C_SD] mandates that only one schema reference is present in each instance XML file and only in the root element; for this reason Schema for XML files should be composed as described in the following subsections in the cases of ESA's Earth Observation Files.

5.1.1 Separated Header file (.HDR) and Data Block file (.<DBL_Ext>¹)

If the Header and Data Block data are located in separate files then this handbook recommends one schema (.XSD) for the Header and separately one Schema for each Data Block files. These schemas are referenced within the Header and in the Data Block files root element as described in [EO_FFS].

5.1.2 Single .EOF file composed by an Header and a Data Block enclosed in <Earth_Observation_File> tag

If the data is a single file comprising the Header and (one) Data Block, then this handbook recommends one single schema (.XSD) for the whole file which internally reference (includes) the 2 schemas, one for the Header part and one for the Data Block part (see Fig 8 below) and to be referenced within the root element of the .EOF data file.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://eop-cfi.esa.int/CFI" xmlns="http://eop-cfi.esa.int/CFI"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.1">
  <xsd:include schemaLocation="EO_OPER_MPL_ORBSCT.HeaderTypes_0100.XSD"></xsd:include>
  <xsd:include schemaLocation="EO_OPER_MPL_ORBSCT.DataBlockTypes_0101.XSD"></xsd:include>
  <xsd:annotation>
    <xsd:documentation>Orbit Scenario File</xsd:documentation>
  </xsd:annotation>
  <xsd:annotation>
    <xsd:documentation>$Revision: 1.1 $ $Date: 2007/07/31 17:08:42 $</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType name="Orbit_Scenario_File_Type">
    <xsd:sequence>
      <xsd:element name="Earth_Explorer_Header"
        type="Orbit_Scenario_Header_Type"></xsd:element>
      <xsd:element name="Data_Block"
        type="Orbit_Scenario_Data_Block_Type"></xsd:element>
    </xsd:sequence>
    <xsd:attribute name="schemaVersion" type="Decimal_Type" use="required"></xsd:attribute>
  </xsd:complexType>
  <xsd:element name="Earth_Explorer_File" type="Orbit_Scenario_File_Type"></xsd:element>

```

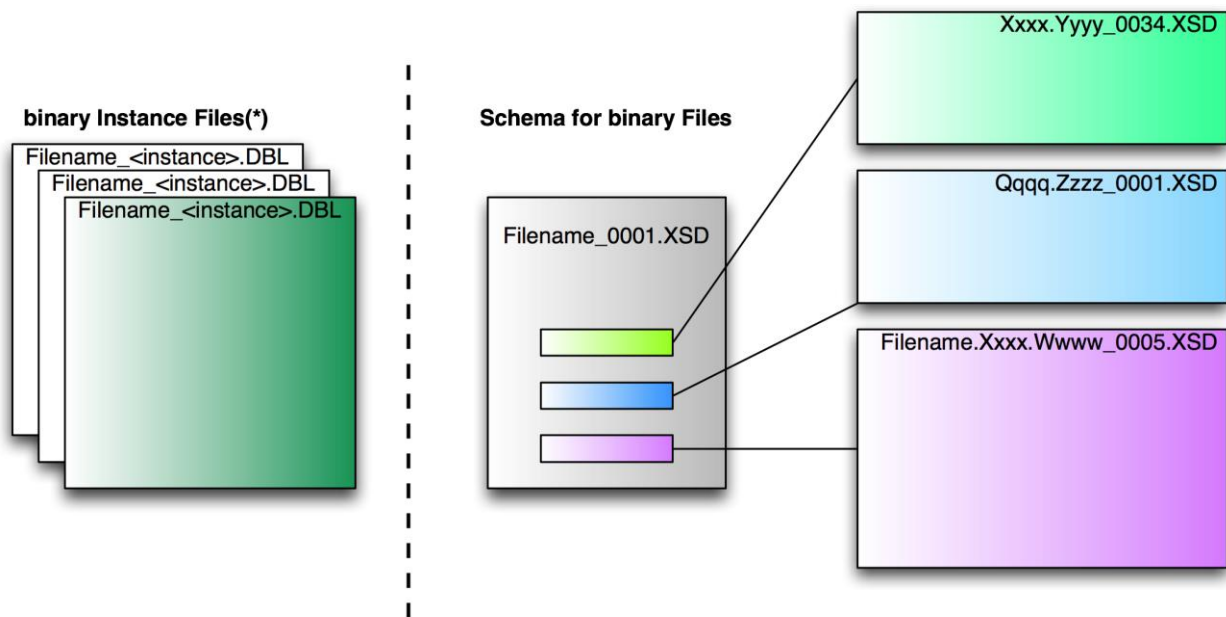
Fig. 8 Example of Schemas for compound Header and Data Block file

¹ Since Data Block files can have multiple extensions according to [EO_FFS], the text <DBL_Ext> is used within this document as generic identifier of the extension.

5.2 Schemas for Binary Files

If the data to be described is a binary file (e.g. a binary Data Block) then this handbook recommends that the Schema file is constituted by one or multiple XSD files. (see Fig .9).

- It is recommended that the definition for binary data are included in a single top level schema file. A hierarchic approach like shown in Fig. 9 is recommended to maximise re-use of common types.



(*) NB Schema for binary can be defined also for non .DBL Data Blocks

Fig. 9 Schema for binary file (generic)

6 FILE NAMING

This section describes the conventions to be used for naming schema files.

Naming convention for Schema files shall be modelled based on the filename of the related data file type as described in [EO_FFS].

NB: [EO_FFS] version 3 introduced multiple data blocks, the present handbook defines a naming convention to avoid ambiguous or duplicated schema filenames.

6.1 Schema for XML files

6.1.1 Top level schema file in the hierarchy

(box shown in gray in Fig. 5)

As defined in [EO_FFS], physical files have dedicated extensions representative to their content

- MMM_CCCC_TTTTTTTTTT_<instance_id>.EOF : Single XML files including a Header and a single XML Data Block
- MMM_CCCC_TTTTTTTTTT_<instance_id>.HDR :XML Header File
- MMM_CCCC_TTTTTTTTTT_<instance_id>.<DBL_Ext> : for separate (multiple) Data Block Files. Note: In this case <DBL_Ext> are XML Data Blocks. For the case of Binary Data Block, please refer to Section 1.1.

Schema File describing structure and format of a .EOF XML file named:

MMM_CCCC_TTTTTTTTTT_<instance_id>.EOF

shall be named :

MMM_CCCC_TTTTTTTTTT_<version_id>.XSD

Schema File describing structure and format of a .HDR XML file named

MMM_CCCC_TTTTTTTTTT_<instance_id>.HDR

shall be named :

MMM_CCCC_TTTTTTTTTT_<version_id>.HDR.XSD

Schema File describing structure and format of a .<DBL_Ext> Data block XML file named

MMM_CCCC_TTTTTTTTTT_<instance_id>.<DBL_Ext>

shall be named :

MMM_CCCC_TTTTTTTTTT_<version_id>.<DBL_Ext>.XSD

where <version_id> is a series of 4 digits (VVVV) following the convention

VV: first version number (release)

vv: second version number (enhancement and update)

(e.g. 0100 corresponds to 1.0, 0102 corresponds to 1.2, 1031 corresponds to 10.31).

6.1.2 Sub component in hierarchy

1. Schema files which are generic component and referenced in a top level Schema file shall be named:

`<SubElementName>_<version_id>.XSD`

where `<SubElementName>` is a variable length character string of the type:

`Xxxxx.Yyyyy.Zzzzz`

And where `<version_id>` is a series of 4 digits (VVVV) following the convention

VV: first version number (release)

vv: second version number (enhancement and update)

(e.g. 0100 corresponds to 1.0, 0102 corresponds to 1.2, 1031 corresponds to 10.31).

2. Schema Files which are specific components (like for example the DataBlock definition for a specific File type) and referenced in a Top Level Schema file `MMM_CCCC_TTTTTTTTTT_<version_id>.XSD` shall be named:

`MMM_CCCC_TTTTTTTTTT.<SubElementName>_<version_id>.XSD`

where `<SubElementName>` is a variable length character string of the type:

`Xxxxx.Yyyyy.Zzzzz`

and where `<version_id>` is a series of 4 digits (VVVV) following the convention

VV: first version number (release)

vv: second version number (enhancement and update)

(e.g. 0100 corresponds to 1.0, 0102 corresponds to 1.2, 1031 corresponds to 10.31).

6.2 Schema for Binary Files

6.2.1 Top level schema file in the hierarchy

(Shown in gray in Fig. 6)

As defined in [EO_FFS], Data Block physical files have dedicated extensions (hereafter referred as `.<DBL_Ext>`) representative of their content and multiple Data Block files are allowed.

Schema File describing structure and format of a Binary Data block (`.<DBL_Ext>`) file named:

`MMM_CCCC_TTTTTTTTTT_<instance_id>.<DBL_Ext>`.

`MMM_CCCC_TTTTTTTTTT_<version_id>.<DBL_Ext>.XSD`

Where `<version_id>` is a series of 4 digits (e.g. 0001, 0002, 0127, 9999)



Since [EO_FFS] allows also the case of multiple physical Data Block files with identical format extensions whereby Data Block Files are differentiated only by an intermediate fixed sized extension, two cases are then possible:

- In the case all these files have identical format structure then only a single Schema file is needed:
The Schema file describing structure and format of binary Data Block files named e.g.:
MMM_CCCC_TTTTTTTTTT_<instance_id>.R001.NC
MMM_CCCC_TTTTTTTTTT_<instance_id>.R002.NC
MMM_CCCC_TTTTTTTTTT_<instance_id>.R003.NC
shall be named :
MMM_CCCC_TTTTTTTTTT_<version_id>.NC.XSD
- In the case all these files have different format structure a dedicated Schema file is needed for each file:
The Schema files describing structure and format of binary Data Block files named:
MMM_CCCC_TTTTTTTTTT_<instance_id>.R001.NC
MMM_CCCC_TTTTTTTTTT_<instance_id>.R002.NC
MMM_CCCC_TTTTTTTTTT_<instance_id>.R003.NC
shall be named, respectively:
MMM_CCCC_TTTTTTTTTT_<version_id>.R001.NC.XSD
MMM_CCCC_TTTTTTTTTT_<version_id>.R002.NC.XSD
MMM_CCCC_TTTTTTTTTT_<version_id>.R003.NC.XSD

6.2.2 Sub component in hierarchy

1. Schema files, which are generic component and referenced in a top level Schema file shall be named:

<SubElementName>_<version_id>.XML (or .XSD depending on the binary dictionary chosen)

where <SubElementName> is a variable length character string of the type:

Xxxxx.Yyyyy.Zzzzz

And where <version_id> is a series of 4 digits (e.g. 0001, 0002, 0127, 9999).

2. Schema Files which are specific components (like for example the DataBlock definition for a specific File type) and referenced in a Top Level Schema file MMM_CCCC_TTTTTTTTTT_<version_id>.XML shall be named:

MMM_CCCC_TTTTTTTTTT.<SubElementName>_<version_id>.XSD

where <SubElementName> is a variable length character string of the type:

Xxxxx.Yyyyy.Zzzzz

and where <version_id> is a series of 4 digits (e.g. 0001, 0002, 0127, 9999).

7 FILE SYNTAX

This section describes the prescribed form and conventions of the syntax of the attribute relating to schema definition, namespace, versioning as well as the reference within the schema and, when applicable, within the instance file.

7.1 Schema for XML Syntax

XML Schemas are formatted as standard .XSD files as per W3C recommendations [W3C_SD].

The following attributes **shall** be present in the root element of the schema file (i.e. the <xsd:schema> node):

Attribute	Comment	Value	Examples
xmlns	Define what the default namespace URI within this schema declaration is.	Project defined URI	<code>xmlns="http://eop-cfi.esa.int/SWARM"</code>
xmlns:xsd	Define the "basic" namespace URI for definitions and specifies xsd as the prefix.	<code>http://www.w3.org/2001/XMLSchema</code>	<code>xmlns:xsd="http://www.w3.org/2001/XMLSchema"</code>
targetNamespace	Define the the namespace URI of the elements that are being defined in the schema. It is recommended to define a specific one for every domain or project, see examples.	Project defined URI	<code>targetNamespace="http://eop-cfi.esa.int/XBS"</code> <code>targetNamespace="http://eop-cfi.esa.int/S1"</code> <code>targetNamespace="http://www.my-server.com/ZZ/Projectname"</code>
elementFormDefault		<code>qualified</code>	<code>elementFormDefault="qualified"</code>
attributeFormDefault		<code>unqualified</code>	<code>attributeFormDefault="unqualified"</code>
version	Textual representation of the version of the schema, see section 8 of this document	String in X.Y format	<code>version="1.0"</code> <code>version="2.3"</code>

NOTE: this recommendation prescribes the use of qualified XMLSchema and default targetNamespace.

7.2 Schema for Binary Syntax

The format and syntax of schema files for binary data are dependent on the binary description language.

In case they are formatted as .XSD files (e.g. DFDL language) they should comply with the same syntax as section 7.1 above.



7.2.1 DFDL case

DFDL schemas are an extension of .XSD ones and shall follow [DFDL_SPEC].

The root element attributes shall follow the same rules as section 7.1 above.

7.3 Reference to Schemas

This section described a standardised way to include a schema reference within each .XML file.

The following attributes **shall** be present in the root element of each XML file (i.e. the root element node, for EO files typically within the <Earth_Observation_File> tag.

NOTE: This guideline RECOMMENDS to use XML schemas defining a namespace. In case the applicable schema does not have a namespace the syntax below (*xsi: noNamespaceSchemaLocation*) shall be instead used.

Attribute	Comment	Value	Examples
xmlns	Define the default namespace URI, i.e. the namespace for the XML elements without prefix. It is recommended to define one for each project. Must match the one of the element defined in the Schema.	Project defined URI	xmlns="http://eop-cfi.esa.int/XBS" xmlns="http://www.mysrv.int/myproject"
xmlns:xsi	Define the "basic" schema instance namespace URI and specifies xsi as prefix	http://www.w3.org/2001/XMLSchema-instance	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation RECOMMENDED: SEE NOTE ABOVE	Define the namespace URI and the physical location of the schema to be used for validation in format of 2 strings separated by a blank space. While the location of the schema can be changed at any time to allow, for instance, remote access to a central repository it is recommended that the first and "baseline" location be local (i.e. same folder as the file)	Project defined URI	a) xsi:schemaLocation="http://eop-cfi.esa.int/CFI schemafilename.XSD" b) xsi:schemaLocation="http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/mydirectory/schefilename.XSD" Example a) is a local schema, b) a remote schema
xsi:noNamespaceSchemaLocation OPTIONAL: SEE NOTE ABOVE	References an XML Schema that does not have a target namespace. In form of a string.	Location and name of the applicable XML schema	a) xsi:noNameSpaceSchemaLocation="schemafilename.XSD" b) xsi:noNameSpaceSchemaLocatio=



	<p>While the location of the schema can be changed at any time to allow, for instance, remote access to a central repository it is recommended that the first and “baseline” location be local (i.e. same folder as the file)</p>		<p><code>"cfi.esa.int/CFI/mydirectory/sche filename.XSD"</code></p> <p>Example a) is a local schema, b) a remote schema</p>
<p>schemaVersion</p>	<p>Textual representation of the version of the schema, see section 8 of this document</p>	<p>String in X.Y format</p>	<p><code>schemaVersion = "0.3"</code> <code>schemaVersion = "5.1"</code></p>

8 SCHEMAS VERSIONING

8.1 XML Files Schema Versioning

XML schemas evolution is handled by use of versioning mechanisms. This section describes how to record the schema's version in a standardised way within and across different missions and ensure coherency between schemas and instance files themselves.

8.1.1 Schema Version Identification

The schema version shall be indicated in the optional version attribute at the beginning of the XML schema. This internal schema version number shall be consistent with the version number given in the schema filename.

In addition, an “schemaVersion” attribute shall be created on the root element of the instance file. The “schemaVersion” attribute is used in the instance file to declare the version of the schema with which the instance is compatible.

With this approach, an application can compare the schema version (captured in the schema file) with the version to which the instance file reports that it is compatible.

In both the XML schema and the instance file, the schema version attribute shall be written as “X.Y”, “X.YY” or “XX.YY”, depending on the number of digits needed.

See examples below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by name (company) -->
<xsd:schema xmlns="http://eop-cfi.esa.int/CFI" targetNamespace="http://eop-cfi.esa.int/CFI"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.1">
  <xsd:include schemaLocation="EO_OPER_MPL_ORBPRED.HeaderTypes_0201.XSD"/>
  <xsd:include schemaLocation="EO_OPER_MPL_ORBPRED.DataBlockTypes_0200.XSD"/>
  <xsd:annotation>
    <xsd:documentation>Predicted Orbit File</xsd:documentation>
  </xsd:annotation>
  <xsd:annotation>
    <xsd:documentation>$Revision: 1.1 $ $Date: 2011/10/07 08:01:47 $</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType name="Predicted_Orbit_File_Type">
    <xsd:sequence>
      <xsd:element name="Earth_Explorer_Header" type="Predicted_Orbit_Header_Type"/>
      <xsd:element name="Data_Block" type="Predicted_Orbit_Data_Block_Type"/>
    </xsd:sequence>
    <xsd:attribute name="schemaVersion" type="xsd:decimal" use="required"/>
  </xsd:complexType>
  <xsd:element name="Earth_Explorer_File" type="Predicted_Orbit_File_Type"/>
</xsd:schema>
```

Fig. 10 Example of Schema for XML files: EO_OPER_MPL_ORBPRED_0210.XSD



```

<?xml version="1.0"?>
<Earth_Explorer_File xmlns="http://eop-cfi.esa.int/CFI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://eop-cfi.esa.int/CFI http://eop-cfi.esa.int/CFI/EE_CFI_SCHEMAS/EO_OPER_MPL_ORBPRES_0200.XSD"
  schemaVersion="2.0">
  <Earth_Explorer_Header>
    <Fixed_Header>
      <File_Name>S1A_TEST_MPL_ORBPRES_20111231T102653_20120107T115724_0001</File_Name>
      <File_Description>FOS Predicted Orbit File</File_Description>
      <Notes></Notes>
      <Mission>Sentinella</Mission>
      <File_Class>TEST</File_Class>
      <File_Type>MPL_ORBPRES</File_Type>
      <Validity_Period>
        <Validity_Start>UTC=2011-12-31T10:26:53</Validity_Start>
        <Validity_Stop>UTC=2012-01-07T11:57:24</Validity_Stop>
      </Validity_Period>
      <File_Version>0001</File_Version>
      <Source>
        <System></System>
        <Creator>EO_ORBIT:xo_gen_pof</Creator>
        <Creator_Version>4.2</Creator_Version>
        <Creation_Date>UTC=2011-06-01T09:18:28</Creation_Date>
      </Source>
    </Fixed_Header>
    <Variable_Header>
      <Ref_Frame>EARTH_FIXED</Ref_Frame>
      <Time_Reference>UTC</Time_Reference>
    </Variable_Header>
  </Earth_Explorer_Header>
  <Data_Block type="xml">
    <List_of_OSVs count="2">
      <OSV>
        <TAI>TAI=2011-12-31T10:27:26.906682</TAI>
        <UTC>UTC=2011-12-31T10:26:52.906682</UTC>
        <UT1>UT1=2011-12-31T10:26:52.906684</UT1>
        <Absolute_Orbit>+01761</Absolute_Orbit>
        <X unit="m">-2797510.395</X>
        <Y unit="m">+6500475.042</Y>
        <Z unit="m">+0000000.000</Z>
        <VX unit="m/s">+1458.377481</VX>
        <VY unit="m/s">+0617.969409</VY>
        <VZ unit="m/s">+7430.334805</VZ>
        <Quality>000000000000</Quality>
      </OSV>
      <OSV>
        <TAI>TAI=2011-12-31T12:06:11.478111</TAI>
        <UTC>UTC=2011-12-31T12:05:37.478111</UTC>
        <UT1>UT1=2011-12-31T12:05:37.478113</UT1>
        <Absolute_Orbit>+01762</Absolute_Orbit>
        <X unit="m">+0173009.448</X>
        <Y unit="m">+7074765.573</Y>
        <Z unit="m">+0000000.000</Z>
        <VX unit="m/s">+1583.188849</VX>
        <VY unit="m/s">-0047.582930</VY>
        <VZ unit="m/s">+7430.334805</VZ>
        <Quality>000000000000</Quality>
      </OSV>
    </List_of_OSVs>
  </Data_Block>
</Earth_Explorer_File>

```

Fig. 11 Example of XML instance file: Predicted Orbit file



Old EE XML files may not include the reference to the schema location and version.

8.1.2 Convention for Schema Versioning

Consider two cases for changes to XML schemas:

CASE 1. The new schema changes the interpretation of some element (e.g., a construct that was valid and meaningful for the previous schema does not validate against the new schema).

In this case, a version number shall be incremented by one (e.g., v1.0 to v2.0).

CASE 2. The new schema is only extended (e.g., new elements, attributes, extensions to an enumerated list, etc) but does not invalidate previously valid instances.

In this case, a version number shall be incremented by less than one (e.g., v1.2 to v1.3).

In both cases, once a schema is modified, the approach would be to do the following:

- concerning the schema file, it would be required to change the schema version number within the schema and in the schema filename.
- with respect to the instance files that use the updated schema, it would be necessary to modify the “schemaVersion” attribute and the version in the schema filename if schema location is specified in the root element.