



Handling of Code Components in IEC Standards Including Copyright Licensing

v8.0 - 2019-08-23

Table of content

| | | |
|-------|----------------------------------------------------------------------------|----|
| 1 | Introduction | 2 |
| 1.1 | General | 2 |
| 1.2 | Versioning information | 2 |
| 1.3 | Editing conventions | 2 |
| 2 | Terms and abbreviations | 2 |
| 2.1 | General | 2 |
| 2.2 | Main roles | 3 |
| 3 | Procedure for including code components in documents | 4 |
| 3.1 | General | 4 |
| 3.2 | Code component identification | 4 |
| 3.3 | Approval by IEC Central Office | 4 |
| 3.3.1 | Preparation and notification to IEC Central Office | 4 |
| 3.3.2 | Assessment by IEC Central Office | 5 |
| 3.4 | Modification of the deliverable to embed and publish code components | 5 |
| 3.5 | Access to the code component content during the development phase | 5 |
| 3.6 | Submission to National Committees | 5 |
| 3.7 | Publication Process of the code component | 5 |
| 3.8 | Posting code components | 6 |
| 3.8.1 | General | 6 |
| 3.8.2 | Free and public access | 6 |
| 3.8.3 | Protected access | 6 |
| 3.9 | Summary of the main step for publishing code components | 7 |
| 4 | Formal references associated with code component | 8 |
| 4.1 | General | 8 |
| 4.2 | Common data | 8 |
| 4.3 | Character string rules | 8 |
| 4.4 | IECCommitteeName | 9 |
| 4.5 | CodeComponentPackageName | 9 |
| 4.5.1 | General | 9 |
| 4.5.2 | RefStandard | 9 |
| 4.5.3 | CodeComponentName | 9 |
| 4.5.4 | VersionStateInfo | 9 |
| 4.5.5 | LightFull | 9 |
| 4.5.6 | PublicationStage | 10 |
| 5 | Handling of code components in the IEC deliverable | 10 |
| 6 | IEC code components packaging for distribution as separated file | 11 |
| 6.1 | Purpose | 11 |
| 6.2 | Generic notice | 11 |
| 6.2.1 | General | 11 |

DRAFT

| | | |
|---------|-------------------------------------------------------------------------------------------------------------------|----|
| 6.2.2 | Manifest file | 12 |
| 6.2.3 | Package | 13 |
| 6.3 | MIB notice | 14 |
| 6.3.1 | Copyright statement inclusion | 14 |
| 6.3.2 | Full MIB and light MIB | 14 |
| 6.4 | XML notice | 15 |
| 6.4.1 | General | 15 |
| 6.4.2 | Copyright notice included as a comment | 16 |
| 6.4.3 | Copyright notice included as part of the IEC XML grammar (checkable) | 16 |
| 6.5 | XSD notice | 16 |
| 6.5.1 | General | 16 |
| 6.6 | UML (XMI) notice | 17 |
| 6.6.1 | General | 17 |
| 6.6.2 | Copyright notice included as a UML note within a UML Diagram | 17 |
| 7 | Further updates of the guideline document | 17 |
| Annex A | Copyright Notice | 18 |
| Annex B | Software EULA | 19 |
| Annex C | Using draft code components | 20 |
| Annex D | Manifest XSD schemas | 21 |
| Annex E | XML schema to embed in order to get checkable copyright notice presence into IEC code component of XML type | 25 |

1 Introduction

1.1 General

This document sets out the process and rules to be used by IEC groups (TCs, SCs, PCs, WGs, SyCs, ...) and experts in charge of editing IEC documents to ensure a proper handling of copyright licensing of code components included in IEC deliverables. It also defines the technical and process requirements to consider to optionally offer a free access to certain code component(s) through the IEC web site.

At the current time this document only applies to IEC TC57.

1.2 Versioning information

Date: 2019-08-23
Document version: v8.0
Licensing conditions version: v0

1.3 Editing conventions

Any term of this document expressed within brackets { ...} refers to a specific term defined precisely in the clause 4 and should be replaced by its actual content at the time of implementation.

2 Terms and abbreviations

2.1 General

2.1.1

Code component

Any piece of information, intended to be directly processed by a computer and encoded in accordance with specific software code language rules (typically XML, XSD, Java, C, ...).

NOTE there may be multiple code components into one standard

DRAFT

2.1.2

Code component(s) package

Package of files used to distribute the code component(s), including the IEC copyright and licensing rules, the IEC manifest and the electronic file(s) reflecting the code component(s) content.

2.1.3

Electronic file

Digital container of serialised data, reflecting the code component content, directly processable by a computer.

2.1.4

IEC group

A team of experts in charge of producing IEC deliverables, following the IEC procedures as defined in IEC Directives, typically TC, SC, PC, SyC Committees, and further PT, MT or WG group.

2.1.5

IEC deliverable

Any document elaborated by an IEC group, and resulting from the application of IEC procedures, as defined in the IEC Directives, typically International Standards (IS), Technical Specifications (TR) or Technical Reports (TS), and any of the transitory stages of such document such as DC, DTR, CD, DTS, CDV, FDIS,

2.1.6

free and public access

A mean by which any IEC standard users will be offered to get access to the code component (typically through the IEC web site)

2.1.7

protected access

A mean by which only selected IEC standard users will be offered to get access to the code component (typically through a private area of the IEC web site which will require a specific login)

2.2 Main roles

2.2.1

editor

person/expert or group of persons/experts specifically in charge of producing the deliverable within an IEC group. It may be the IEC group leader by default.

2.2.2

IEC group leader

person in charge of ensuring that the ISO/IEC Directives are followed for the corresponding scope of work, typically the convenor for a WG, a MT, a PT or the secretary for a TC, SC, PC ...

2.2.3

IEC Central Office

IEC entity as defined in the IEC organisation which supervises within IEC the proper application of the Statutes, Rules of Procedure and Directives, and ensures project management, transmission of working documents and the publication of final texts of standards.

2.2.4

IEC Head of Sales and Business Development

person in the IEC Central Office heading Sales and Business Development.

2.2.5

IEC officers

The officers (chairman, secretary) of the TC, SC, PC, SyC ... to which the IEC group belongs.

2.2.6

web publisher

person/expert or group of persons/experts specifically in charge of publishing code components onto the IEC web site at {Code Components Repository URL} for a specific IEC group. By default, the IEC officers are granted write access to this page. They may possibly grant write access to other people.

DRAFT

It is recommended first that the IEC group leader(s) are granted with such permission, and in addition that the IEC group leader(s) designate within the group an expert specifically in charge of publishing such content. As a backup, the web content administrator may take the role.

2.2.7

web content administrator

person/expert or group of persons/experts specifically in charge of administrating all code components of a specific IEC group onto the IEC web site at {Code Components Repository URL}, as well as updating the web page (or web frame within a page) which would expose some news related to the availability of new code components onto the IEC web site, and possibly other web contents (such as the content of the restricted area related to subscribed users). By default, the IEC officers are granted write access to this page. But they may grant write access to other people. It is recommended that the IEC group designates within the group an expert specifically in charge of administrating such content.

3 Procedure for including code components in documents

3.1 General

This section describes the main steps to follow to include code component(s) into an IEC deliverable and to optionally produce and publish such code component(s) as a package directly accessible from the IEC web site.

Such an approach may happen at any development stage of the targeted IEC deliverable but it is highly recommended to proceed implementing such an approach as early as possible in the writing process.

3.2 Code component identification

As an IEC group develops a document according to IEC procedures, it should first identify the presence of code components.

At the earliest feasible stage in the development process (typically at the CD or DC stage), the editor of the document should screen the contents and identify all “code components” elements contained in the document, i.e. those contents which would otherwise have to be copied/modified from the IEC publication by any potential users of the standards, just for the sake of applying the standard.

Note: this means that the proposed code components will be reviewed by National Committees during the standard development process. However these draft versions will not be hosted on the public area, but only through the IEC Collaboration tool as explained in 3.5.

3.3 Approval by IEC Central Office

3.3.1 Preparation and notification to IEC Central Office

As soon as a code component is identified and prior to the next stage of submission to NCs, IEC group leader in charge of the document should notify the IEC Central Office that a document is being submitted which contains code component(s).

For that purpose, the IEC group should prepare a publication information statement to be sent to the IEC Head of Sales and Business Development describing:

- The concerned draft publication
- The expected usage of the code components by the purchaser, by intermediaries, and by other end-users ...

This publication information statement should reflect as much as possible the user's intended usage of the code component, to enable IEC Central Office to confirm that the licensing model in place for code components will meet users' needs for the particular code component.

One possibility to open the dissemination of the code component to the free and public access might be to reduce its exposed content for free publication compared to what is in the IEC full code

DRAFT

component, especially regarding the level of depth of textual descriptions (refer to clause 4.5.5). This might be complementary to the protected access.

Once the group agrees on the proposed publication statement content, The IEC group leader should post the proposed publication policy mentioned in clause 3.3.1 onto the collaboration tool and notify the IEC Head of Sales and Business Development.

NOTE When matured this process will go through the TC57 Technical Officer.

NOTE At current time the IEC Head of Sales and Business Development is Ms Guilaine Fournet (gnf@iec.ch)

3.3.2 Assessment by IEC Central Office

On a case-by-case basis, the IEC Central Office will assess the deliverable, the identified code component, and the proposed policy, and where appropriate will approve the application of the “specific” copyright licensing terms and conditions for code components in that standard, as described further in this document (refer to Annex A and Annex B).

Within one month the IEC will provide a first answer to the proposal. If there are any issues, both IEC central Office and the IEC Group will try to find the best compromise.

When an IEC deliverable with code components is approved by the IEC Central Office, i.e. is approved for code component licensing, such notice will be attached to the publication policy.

3.4 Modification of the deliverable to embed and publish code components

As soon as approval is received from the IEC Central Office, the deliverable shall be modified as detailed in clause 5 and clause 6.

In the case when the publication of the code component(s) is agreed through the IEC web site, the code component(s) needs to be extracted and associated to copyright and licensing rules as defined by the IEC. Clause 6 depicts how to proceed, depending on the code language.

3.5 Access to the code component content during the development phase

Draft code components are posted on the IEC collaboration tools as any draft work, prior to their submission to NCs, thus available to all experts from the concerned WG, for comments..

During the review phases of the IEC deliverable by NCs, code components will only be available for review to IEC experts and IEC National Committees (as any draft publication), and will not be downloadable by anybody on the public web area.

In addition, “draft” code components may support some very specific experimentation usages, especially for field testing. Such usage is conditioned by specific right and duties exposed in Annex C.

This is the reason why any draft code components shall be explicitly tagged as “draft” as detailed in clause 4, 5, 6. Of course this draft mention shall disappear from the published version. A code component is considered as draft at any development stages (typically DC; CD, CDV, DTR,DTS, FDIS, ...) except the official publication.

3.6 Submission to National Committees

The deliverable, together with the code components and the code component publication policy shall be submitted for review to National Committees as per IEC directives.

This means that draft code components (i.e before having reached the official publication stage, typically in DC, CD, DTR; DTS, CDV, FDIS stages) will circulate to NCs and national experts as any IEC draft publication.

In case of dual publication types (light and full) both versions will follow the exact same process at the same time.

3.7 Publication Process of the code component

Code components (if any) are part of the official IEC publication and thus are published by IEC together with the IEC publication on the web-store. In case light and full versions co-exist, only the full version is associated with the IEC official publication.

DRAFT

In case the publication of the code component through the IEC web site is agreed, it requests some specific care because it replicates content which is also available on the IEC webstore. So, synchronisation and alignment is required.

In the case when the publication of the “public” code component(s) is to be synchronised with the official publication of IEC on its webstore, the proposed process is described below:

- The IEC group leader sends an official request to the IEC Central Office.
- Upon readiness of publication, the IEC Central Office informs the web publisher and the IEC group leader that the code component can be posted on the web site.
- At reception of this acknowledgement, the web publisher posts the package containing the code component at {Code Components Repository URL} where the “draft” indication has been taken out, and informs the web administrator to update the web content he is in charge of.
- The web administrator updates the associated content such as the front page, and other related content (web-access content for example)
- The web publisher informs back the IEC Central Office that the web content is ready.

3.8 Posting code components

3.8.1 General

In case of dual publication types (light and full), i.e. serving both the free and public access, and the protected one, both versions will be posted at the same time.

3.8.2 Free and public access

All publicly available code components of a TC (whatever their versions) will be at current stage hosted on the same web page (the “SupportDocuments” web page of the TC - {Code Components Repository URL} as defined in 4.2), and therefore this requires some rules in order to be easily browsed by the users.

Globally, the information should be “just enough” to identify the code component.

Thus, the fields shall be filled this way by the web publisher:

| Field | Content | Comment |
|-----------------------|-----------------------------------------------------------|---------------------------------------------------------------|
| Title | {CodeComponentPackageName} | Refer to 4.2 |
| Description | Code Component Name with the main IP reference | |
| Download size | (automatically filled) | |
| Date of creation | (automatically filled) | |
| Publication reference | Select within the proposed list the concerned publication | Should have been entered previously by the IEC Central Office |

Example :

TC 57 Power systems management and associated information exchange

Scope Structure Projects / Publications Documents Votes Meetings Collaboration Platform

Working Documents Other Documents Supporting Documents Log in En Fr

Documents available in this area are supporting documents uploaded by the committee. The IEC is not liable for the content of these documents.

TC 57 Supporting Documents Table search: [] [] []

| Title, description | Downloads | Created | Publication reference |
|----------------------------------------------------------------------------------------------------------------------|-----------|------------|------------------------|
| IEC_61850-90-6.NSD.2018A.light IEC 61850-90-6 light code component - See IEC TR 61850-90-6 for full legal notices | 17 kB | 2018-10-05 | IEC TR 61850-90-6:2018 |

3.8.3 Protected access

The host and rules for “protected access” still needs to be defined.

3.9 Summary of the main step for publishing code components

This section sums up the clauses from 3.2 to 3.9, i.e. the main steps to follow to embed and publish code components associated to IEC deliverables.

Figure 1 describes the steps up to the publication, and Figure 2 depicts the publication steps.

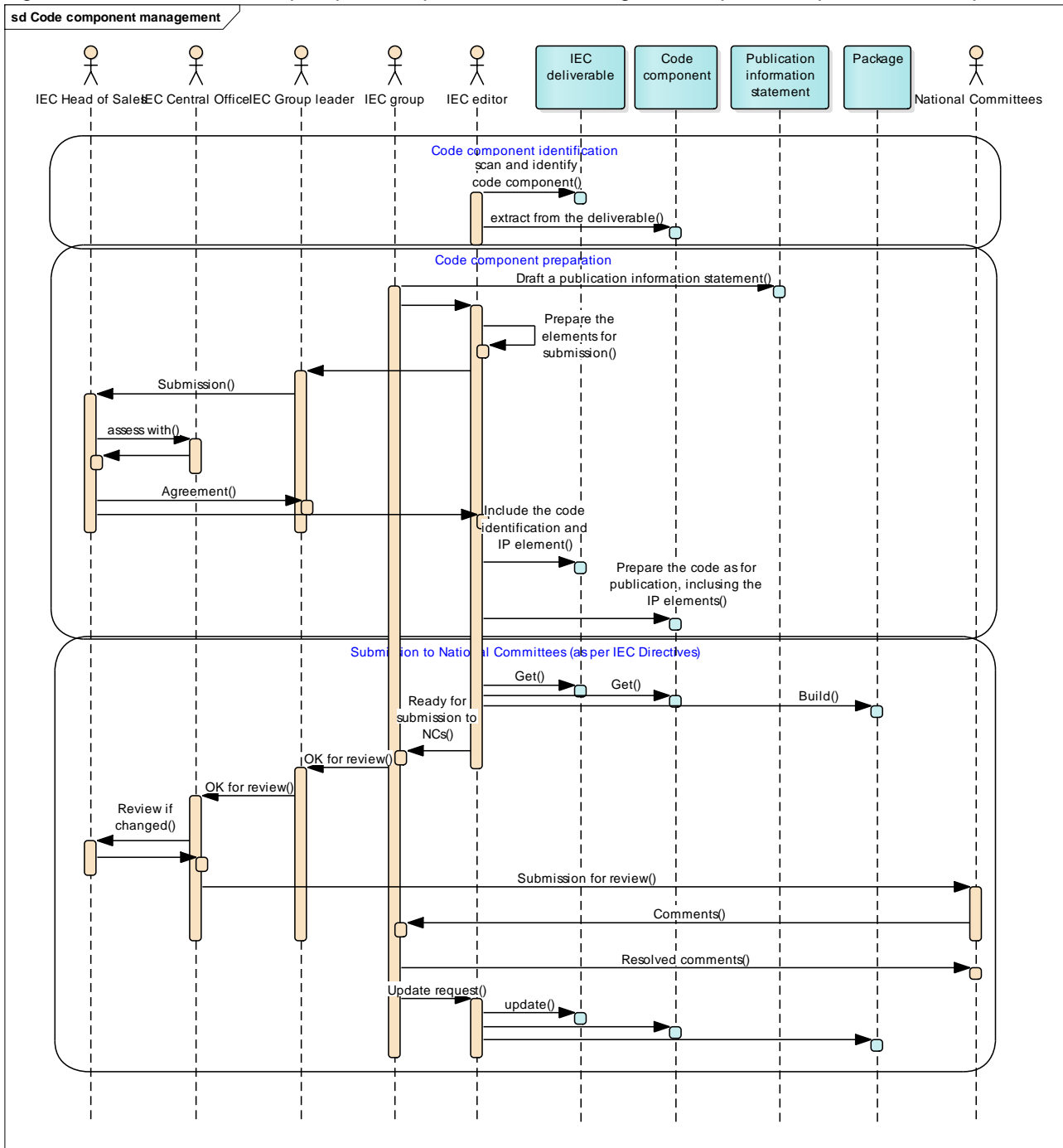


Figure 1 Main steps to follow to identify, prepare IEC deliverables to embed code components

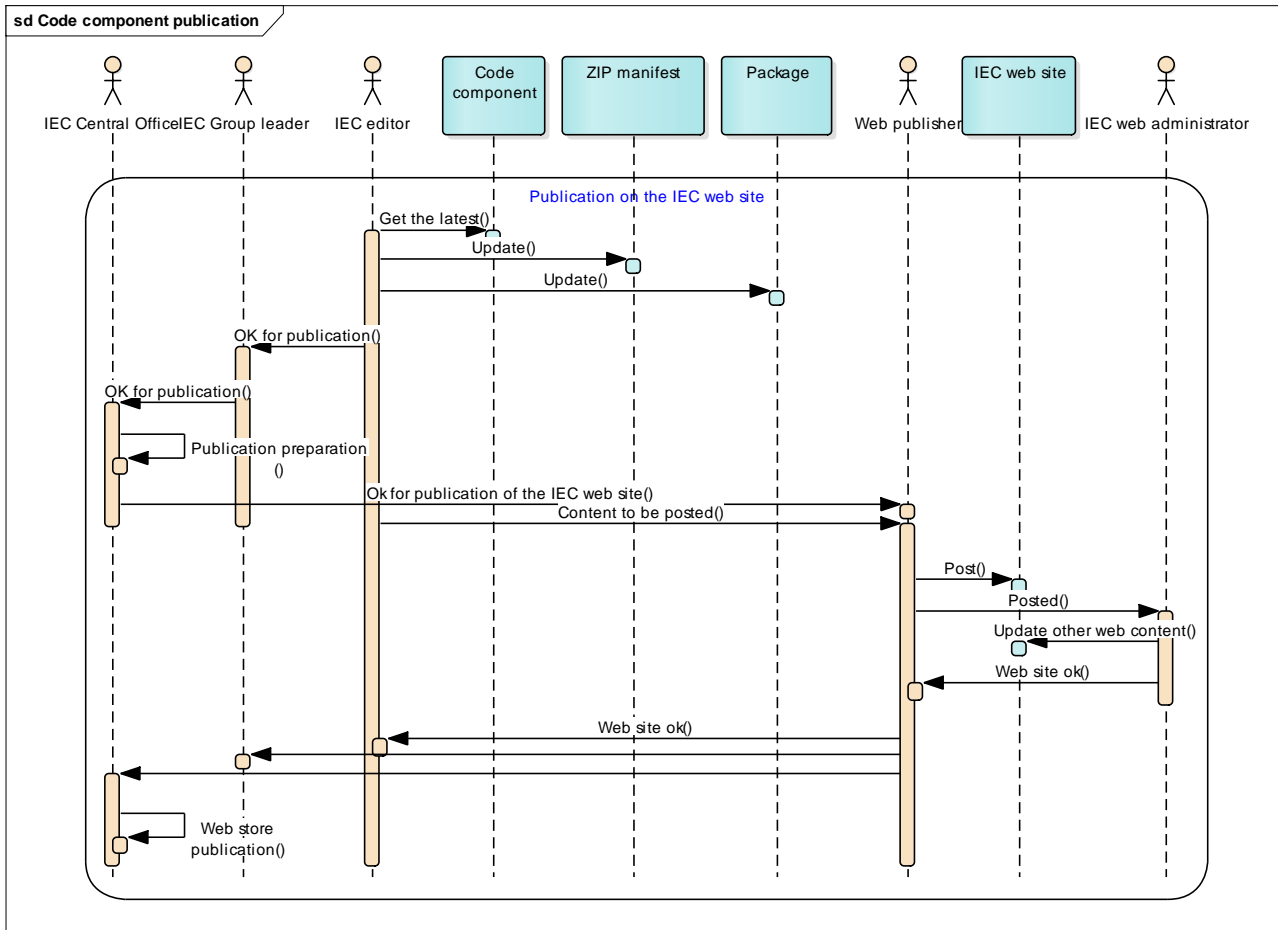


Figure 2 Steps to follow to publish code components

4 Formal references associated with code component

4.1 General

Here are the main general rules to apply when forming a filename used for the package including the code component(s) file(s). The objective of these common rules is to ensure that across all involved IEC groups, code component naming will be harmonised, and will also be easily supported by different operating systems.

For this reason, the rules rely on generic names such as “IECCommitteeName”, “CodeComponentPackageName”, ... whose details are given below.

4.2 Common data

License URI: www.iec.ch/CCv1

Code components repository URL: <http://www.iec.ch/{IECCommitteeName}/supportdocuments>

Code component URL: {Code components repository URL}\{CodeComponentPackageName}

The 3 fields IECCommitteeName and CodeComponentPackageName are described in the clauses 4.4 and 4.5.

4.3 Character string rules

Any character string used in the rest of clause 4 shall follow the rules listed below:

- No blank – if text separation is needed, substitute blank by “_”

DRAFT

- No special characters, no accents, no signs except “-“, “.” and “_”.

4.4 IECCommitteeName

{IECCommitteeName} is used to form the URL on the IEC web site. Lowercase letters shall be used.

It designates the IEC Committee (TC or SC or PC or SyC ...) which hosts the IEC group in charge of producing the IEC deliverable including the code components. There shouldn't be any blank in the name, example: “tc57” or “sc121a”

4.5 CodeComponentPackageName

4.5.1 General

{CodeComponentPackageName} designates the full name of a ZIP file containing the Code component(s), as it is stored on the IEC server.

The expected syntax is:

{RefStandard}.{CodeComponentName}.{VersionStateInfo}.{LightFull}{PublicationStage}.zip

Each of the fields are described below.

4.5.2 RefStandard

{RefStandard} designates the parent IEC deliverable (standard document) and version which includes the code component. The expected syntax is:

{IEC{StandardType}_{StandardNumber}}

With:

- StandardType = {_TR|_TS} or empty if IS
- StandardNumber = 5 digits IEC numbering

Examples :

- “IEC_TS_62357”,
- “IEC_TR_61850-90-4”,
- “IEC_61850-6”

4.5.3 CodeComponentName

{CodeComponentName} designates a part of the parent IEC deliverable which is of type Code Component. The expected syntax is just a string like:

- “Enumeration_List”,
- “XSD_Schema”,

4.5.4 VersionStateInfo

VersionStateInfo designates the version state information of the code component itself, expressed as a String (typically may include version, revision, release number ...). Should be expressed in a way it can be sorted, and where the highest value designates the latest version.

The format depends on conventions defined by the related IEC deliverable.

4.5.5 LightFull

LightFull indicates whether the content included in the considered file, fully reflects the Code component content (full) or reflects just a part of it (light).

LightFull = {light|full}

NOTE if the file contains a list of code components with different attribute light or full, as soon as one of them is light, the ZIP file should be indicated as light

4.5.6 PublicationStage

PublicationStage reflects the IEC stage of the considered code component. Typically

PublicationStage = {Draft} or nothing if this is an official publication

5 Handling of code components in the IEC deliverable

As soon as at least one code component is present in an IEC deliverable, the editor shall:

- include within the foreword of the document the specific license conditions which will apply to code components included in the document, namely:

“This IEC standard includes Code Components i.e. components that are intended to be directly processed by a computer. Such content is any text found between the markers <CODE BEGINS> and <CODE ENDS>, or otherwise is clearly labelled in this standard as a Code Component.

The purchase of this IEC standard carries a copyright license for the purchaser to sell software containing Code Components from this standard to end users either directly or via distributors, subject to IEC software licensing conditions, which can be found at: www.iec.ch/CCv1.”

- introduce exactly before the beginning and after the end of each set of Code Components respectively the tag <CODE BEGINS> and the tag <CODE ENDS>, or otherwise a clear label identifying this specific content as a Code Component.
- Explicitly state the “draft” status of the concerned code components, within the code component content itself as described in clause 6. This draft status will be taken out only at the stage of the official publication of the code component (whatever the channel of publication)

A table shall indicate the one-to-one relationship between all previous IEC editions/ amendments/ corrigenda with their associated code components name.

In the specific case of Code Components which have to be distributed by IEC as a separate file or package of files (as described in the publication policy), some additional measures are expected to be followed:

- Within the document with code components, a sentence in the clause 1 of the document should state where the latest code component file can be found. Typically, this sentence would state:

“The Code Components included in this IEC standard are also available as electronic machine readable file at:

*[where:](http://www.iec.ch/{IECCommitteeName}/supportdocuments/{CodeComponentPackageName}”</i></p>
</div>
<div data-bbox=)*

- “[*“The Code Component\(s\) included in this IEC standard are potentially subject to maintenance works and user shall select the latest release in the repository located at:*](http://www.iec.ch/{IECCommitteeName}/supportdocuments/{CodeComponentPackageName}” represents the URL of the IEC file hosting the considered “final” IEC code components. “final” means that it must not include any PublicationStage information
If further versions/revisions may apply, due to some maintenance works, a sentence shall also indicate how to find the latest version. Typically, this sentence would state:

</div>
<div data-bbox=)

*[*The latest version/release of the document will be found by selecting the file {RefStandard}.{CodeComponentName}.{VersionStateInfo}.{LightFull}{PublicationStage}.zip with the filed VersionStateInfo of the highest value.*](http://www.iec.ch/{IECCommitteeName}/supportdocuments”.</i></p>
</div>
<div data-bbox=)*

where:

- “

DRAFT

- *VersionStateInfo* should just remain textual as it and not be replaced by the actual version and revision state information
- Because IEC de facto publishes the same content through two channels, an additional sentence shall also indicate which publication is of highest priority in case of differences/discrepancies:

If the downloadable code is of highest priority (typically because of maintenance work):

“In case of any differences between the here-below code and the IEC pdf published content, the here-attached code(s) is(are) the valid one; it may be subject to updates. See history files”.

If the pdf published version is of highest priority:

“In case of any differences between the here-below code and the IEC pdf published content, the IEC pdf published content is the valid one”.

- Within the Code Component file itself, the document’s editor shall include a reference to the IEC copyright and licensing conditions. This reference should not be removed by end users, and should comply with the grammar of the considered code. Details on the way to do it are provided further in clause 6.
- A code component package shall be created as described in clause 6, which will be the file potentially downloaded by users of the IEC deliverable.

6 IEC code components packaging for distribution as separated file

6.1 Purpose

This clause intends to present how IEC code components shall be packaged and include the copyright notice in order to be distributed as separated file.

There are two levels:

- Generic notice applicable for all code components based on a package
- Specific notice dependent of the code component technology (MIB, XML, XSD, ...)

6.2 Generic notice

6.2.1 General

The IEC delivery consists of a package, using ZIP technology, which includes:

- a (machine processable) manifest describing the package content,
- the code component(s) extracted from the IEC publication(s) (typically XSD file, XML file, SNMP MIB file ...).
- (optionally) history files, describing the changes which have been considered in the associated package, since the last IEC publication (at least)

The manifest file is a concept used in the IT world to organize electronic files in a package. However, it appears that there is no standard defining a common format for such a manifest, Therefore, a specific manifest format is defined below for IEC code components distribution.

The package shall follow the ZIP format, as specified by PKWARE® Inc. APPNOTE.PX (<http://www.pkware.com/documents/APPNOTE/APPNOTE-6.3.3.TXT>).

The **name of the ZIP package** should apply the referencing naming rules as defined as {CodeComponentPackageName} in 4.5, i.e. should reflect the name of the IEC publication from which the code component(s) are extracted, as well as the part which is represented by this code component(s) itself, including versioning information.

Note: The standard [ISO/IEC 21320-1:2015 - Information technology - Document Container File - Part 1: Core] also defines ZIP as the format for document containers.

DRAFT

In case the concerned code component is still in draft mode, then this shall be indicated explicitly within the code themselves. Many formats are possible depending on the language. Please refer to each section from 6.3 to 6.5 to find some recommended ways to tag code components as draft.

6.2.2 Manifest file

6.2.2.1 General

The manifest file is an XML file called "IECManifest.xml" with the root IECManifest containing two sections:

- The Copyright section includes the copyright license attached to this package.
- The CodeComponent section lists the elements composing the code component itself, including as well a History section including the list of changes which have occurred since a specified date and/or version of an official IEC publication

Here is a typical structure of such Manifest file:

```
<IECManifest xmlns="http://www.iec.ch/CC/2017/IECManifest" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.iec.ch/CC/2017/IECManifest IECManifest.xsd">
  <Copyright>
</Copyright>
  <CodeComponent>
    <Publication/>
    <File/>
    <HistoryFile/>
  </CodeComponent>
</IECManifest>
```

The structure of such Manifest XML file can be checked using the XSD file specified in Annex C.

6.2.2.2 Copyright section

The Copyright section defines the notice and references the full license file. The license file could be included in the package or be a reference available at IEC web site.

This section is defined as per the XML copyright notice description (see Annex A) by referencing Copyright schema defined in D.2.

6.2.2.3 CodeComponent section

The CodeComponent section identifies the code component with:

- an ID (id attribute) – to be defined by the IEC entity. This ID identifies uniquely the code component content including its version (usually machine processable)
- a name (name attribute), – to be defined by the IEC entity. This name depicts uniquely the code component content including its version (usually for human)
- the type of content (content attribute LightFull, with value full or light)
- the date corresponding to the publication of the content (date attribute with format yyyy-mm-dd)

Then it lists the publication information, needed to identify the parent standard publication. This provided by at least one, and potentially multiple "publication" elements (such as the base standard and the amendments). Each publication contains:

- a name (name attribute as string, reflecting {RefStandard})
- (optionally) a comment (comment attribute as string)

Then it lists all the files in the package and the kind of contents, namely if it is a full or reduced set of code components. Each file is described by one specific XML "File" element containing:

- a name (name attribute as string, reflecting the name of the file in the package)
- the category indicating the nature of publication, whether it is "normative" or "informative": category = {informative|normative}

DRAFT

- the type of content LightFull, whether the attached file reflects the “full” or just a “light” version of the code component:
content = {light|full}
- (optionally) a comment (comment attribute as string)

Finally, it lists the history files, containing the change tracks since a defined former IEC publication. History may be split into different files, therefore each HistoryFile is described by one specific XML HistoryFile element containing:

- a name (name attribute as string, reflecting the name of the history file in the package)
- a starting date (startingDate attribute as date with format yyyy-mm-dd)
- an ending date (endingDate attribute as date with format yyyy-mm-dd)
- a starting code component version state (startingVersion attribute as string)
- an ending code component version state (endingVersion attribute as string)
- (optionally) a comment (comment attribute as string)

6.2.2.4 Manifest example

```
<IECManifest xmlns="http://www.iec.ch/CC/2017/IECManifest" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.iec.ch/CC/2017/IECManifest IECManifest.xsd">
<Copyright>
  <Notice>
    COPYRIGHT (c) IEC, 2016. This version of this code component is part of IEC 61850-6:2009/AMD1:2016;
    see the IEC 61850-6:2009/AMD1:2016 for full legal notices. In case of any differences between the here-below code and the
    IEC published content, the here-below definition supersedes the IEC publication; it may contain updates. See history files. The whole
    document has to be taken into account to have a full description of this code component.
    See www.iec.ch/CCv1 for copyright details.
  </Notice>
  <License uri="www.iec.ch/CCv1">IEC License</License>
</Copyright>
<CodeComponent id="IEC_61850-6.2007B2.SCL.XSD" name="IEC 61850-6 schema 2007B2" date="2016-11-30" >
  <Publication name="IEC_61850-6.2015_ed2.1" comment="xxx"/>
  <File name="SCL.xsd" category="normative" content="full"/>
  <File name="SCL_BaseSimpleTypes.xsd" category="normative" content="full"/>
  <File name="SCL_BaseTypes.xsd" category="normative" content="full"/>
  <File name="SCL_Communication.xsd" category="normative" content="full" comment="yyy"/>
  <File name="SCL_DataTypeTemplates.xsd" category="normative" content="full"/>
  <File name="SCL_Enums.xsd" category="normative" content="full"/>
  <File name="SCL_IED.xsd" category="normative" content="full"/>
  <File name="SCL_Substation.xsd" category="normative" content="full"/>
  <HistoryFile name="history.2007B1.txt" endingDate="2014-12-10" startingDate="2013-12-11"
  endingVersion="pub1 v2" startingVersion ="pub1 v1" comment="history of modifications from v1 to v2
  publication"/>
  <HistoryFile name="history.2007B2.txt" endingDate="2016-12-10" startingDate="2014-12-11"
  endingVersion="pub1 final" startingVersion =" pub1 v2" comment="history of modifications from v2 to final
  publication"/>
</CodeComponent>
</IECManifest>
```

6.2.3 Package

The package itself will be a simple ZIP file with all files described in the Manifest, and the IECManifest.xml file. Additionally, the license file itself could be included.

The naming of the package will be {CodeComponentPackageName}.

| Nom | Taille | Compressé | Modifié le | Créé le | Accédé le | Attributs | Chiffre |
|---------------------------|--------|-----------|------------------|------------------|------------------|-----------|---------|
| EULA.txt | 18 | 18 | 2016-11-09 17:15 | 2016-11-09 17:40 | 2016-11-09 17:40 | A | - |
| history.2007B.txt | 34 737 | 9 726 | 2014-01-22 11:11 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |
| IECManifest.xml | 1 337 | 530 | 2016-11-10 08:42 | 2016-11-09 18:21 | 2016-11-09 18:21 | A | - |
| SCL.xsd | 4 111 | 1 223 | 2016-11-09 17:37 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |
| SCL_BaseSimpleTypes.xsd | 6 258 | 1 028 | 2016-11-09 17:37 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |
| SCL_BaseTypes.xsd | 6 629 | 1 215 | 2016-11-09 17:37 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |
| SCL_Communication.xsd | 12 761 | 1 719 | 2016-11-09 17:38 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |
| SCL_DataTypeTemplates.xsd | 7 894 | 1 447 | 2016-11-09 17:38 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |
| SCL_Enums.xsd | 26 829 | 2 799 | 2016-11-09 17:38 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |
| SCL_IED.xsd | 39 157 | 4 132 | 2016-11-09 17:38 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |
| SCL_Substation.xsd | 28 988 | 2 121 | 2016-11-09 17:40 | 2016-11-09 17:16 | 2016-11-09 17:16 | A | - |

6.3 MIB notice

6.3.1 Copyright statement inclusion

The MIB is a file format used by SNMP to describe devices.

The IETF defines a copyright statement for MIBs (<https://www.ietf.org/iesg/statement/mib-copyright.html>) to:

“The RFC-Editor, during the editing process of an RFC that contains a MIB or a PIB Module, copies a short copyright statement into the DESCRIPTION clause of the MODULE-IDENTITY macro of each MIB or PIB Module in the RFC-to-be.”

The proposed copyright statement for IEC MIBs should be therefore included within the DESCRIPTION of MODULE-IDENTITY macro as well.

In case of draft code component, the « draft » aspect of the code component shall be explicit, as shown in the example below (in the example : part of the DESCRIPTION by inclusion of the “FDIS” publication stage). This mention will be deleted at the time of official publication.

The inclusion of the sentence:

“Copyright (C) IEC. This version of this MIB module is part of {RefStandard}; see the {RefStandard} for full legal notices”

Is requested and results as follows when included in each MIB MODULE IDENTITY:

| | | |
|-------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| clock-agent | MODULE-IDENTITY | |
| | LAST-UPDATED | "201611091400Z" |
| | ORGANIZATION | "IEC" |
| | CONTACT-INFO | "IEC TC57 WG15" |
| | DESCRIPTION | "57-62351-7-Ed1-FDIS Copyright (C) IEC. This version of this MIB module is part of 57-62351-7-Ed1-FDIS; see the 57-62351-7-Ed1-FDIS for full legal notices." |
| | REVISION | "201611091400Z" |
| | DESCRIPTION | "57-62351-7-Ed1-FDIS" |

Note: Please note that the DESCRIPTION field will also include the reference to the related standard (and version and publication stage when applicable).

6.3.2 Full MIB and light MIB

Two set of MIBs may be provided:

- A set of MIBs that includes in the DESCRIPTION field of each object the copy of the description provided in the standard. This set of MIBs is more suitable for the inclusion on

DRAFT

NSM central tools because it provides the direct description of each object semantics to the operator.

- A set of “light” MIBs that provides a reference to the UML object in the object description without including the object description. This approach allows public access to the MIBs while still ensuring that the IEC standard is needed to understand what the MIBs mean.

The MIB files are included in a package together with a general license (EULA). This package contains the following files:

- The standard document in PDF format
- A directory containing MIBs
- A directory containing light MIBs
- The End User License Agreement (EULA) text file.

For example, the UML object Clocks Agent::ClockEntry.TimeTraceable, mapped to the MIB object cLKETimeTraceable, is provided in the following ways:

Inside light MIB:

```
cLKETimeTraceable OBJECT-TYPE
                  SYNTAX
                  MAX-ACCESS
                  STATUS
                  DESCRIPTION
                  ::= { cLKClocksEntry 7 }

                  TruthValue
                  read-only
                  current
                  "Clocks Agent::ClockEntry.TimeTraceable"
```

Inside full MIB:

```
cLKETimeTraceable OBJECT-TYPE
                  SYNTAX
                  MAX-ACCESS
                  STATUS
                  DESCRIPTION
                  ::= { cLKClocksEntry 7 }

                  TruthValue
                  read-only
                  current
                  "The Time Traceable flag shall be set False
                  before the IED has received and qualified a signal from
                  a recognized standard time source. Once it has
                  locked to the recognized standard time source and
                  stabilized, the Time Traceable flag shall be set True. The
                  Time Traceable flag shall not be set False again, so
                  long as the IED can estimate its holdover
                  uncertainty. If at some point, perhaps due to an extended
                  holdover interval, the IED no longer can estimate its
                  holdover uncertainty, then the Time Traceable flag
                  shall be set False."
```

Note: the only difference is within the DESCRIPTION field. The OIDs will be the same both for light MIBs and MIBs.

6.4 XML notice

6.4.1 General

The XML format is used for many purposes when the exchange of data between different tools has to be performed in a standard way.

No specific statement exists about the definition of a copyright attached to an XML as it is mainly used to exchange information in a transitional way.

Two cases are considered, depending whether the copyright notice is defined by IEC as part of the XML grammar (and then checkable) or not.

DRAFT

6.4.2 Copyright notice included as a comment

A common agreement in XML user community is to use a file header comment containing the copyright; this is the simplest way of doing.

Here is a typical inclusion of the copyright notice into an XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    COPYRIGHT (c) IEC, 2016. This version of this NSD is part of IEC 61850-7-4:2007B, and this whole document as
to be taken into account to have a full description of this code component.
    See IEC 61850-7-4:2007B for full legal notice.
    Draft
-->
[XML code component content]
```

In case of draft code component, the « draft » aspect of the code component shall be explicit, as shown in the example above. This mention will be deleted at the time of official publication.

6.4.3 Copyright notice included as part of the IEC XML grammar (checkable)

The copyright notice used for the manifest file (see clause 6) could be reused by including IECCopyright schema to any schema used to verify a code component XML to allow adding same level of copyright notice than in manifest.

In case of draft code component, the « draft » aspect of the code component shall be explicit. This mention will be deleted at the time of official publication.

It is highly recommended to have the “draft” attribute explicitly attached to the namespace header element as shown below – here reflecting directly the IEC publication stage CDV.

Here is an example applied to XML file format:

```
<NS xmlns="http://www.iec.ch/61850/2016/NSD" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.iec.ch/61850/2016/NSD NSD.xsd" id="IEC 61850-7-4" version="2007" revision="B"
umlVersion="WG10UML02v18" publicationStage="CDV" >
  <Copyright>
    <Notice>
      COPYRIGHT (c) IEC, 2016. This version of this NSD is part of IEC 61850-7-4:2007B; see the IEC 61850-
7-4:2007B for full legal notices. In case of any differences between the here-below code and the IEC published content, the
here-below definition supersedes the IEC publication; it may contain updates. See history files. The whole document has to be taken
into account to have a full description of this code component.
      See www.iec.ch/CCv1 for copyright details
    </Notice>
    <License uri="www.iec.ch/CCv1" kind="Standard">IEC License</License>
  </Copyright>
[XML code component content]
</NS>
```

In this latest case, the XML grammar checker (re-using the XSD schema proposed in Annex E has the possibility to warn if this section is missing or not properly set (only the XML structure is checked, not the content)

6.5 XSD notice

6.5.1 General

The XSD format is the schema definition used to verify XML files.

No specific statement exists about copyright notice in XSD. Different ways could be used, the preferred method would be one that could be interpreted automatically if needed.

The way to express copyright notice in XSD uses the annotation elements which are used to integrate comments in an XSD. The first annotation of the schema defines a first documentation dedicated to copyright and if needed a second documentation section related to the XSD itself.

In case of draft code component, the « draft » aspect of the code component shall be explicit, as shown in the example below. This mention will be deleted at the time of official publication.

Here is an example of copyright for an XSD schema:

```
<xs:schema>
  <xs:annotation>
    <xs:documentation xml:lang="en">
```


DRAFT

COPYRIGHT (c) IEC, 2016. This version of this XSD is part of IEC 61850-6:2009; see the IEC 61850-6:2009 for full legal notices. In case of any differences between the here-below code and the IEC published content, the here-below definition supersedes the IEC publication; it may contain updates. See history files. The whole document has to be taken into account to have a full description of this code component.

See www.iec.ch/CCv1 for copyright details

```
</xs:documentation>
```

```
<xs:documentation xml:lang="en">
```

SCL schema version "2007" revision "B" release 2014-01-22, for IEC 61850-6 Ed. 2.1.

Draft

```
</xs:documentation>
```

```
</xs:annotation>
```

```
</xs:schema>
```

6.6 UML (XMI) notice

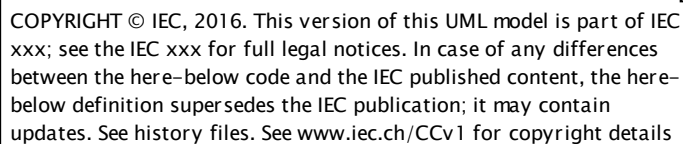
6.6.1 General

The UML (or XMI) format may as well be used for exchanging data between different tools in a standard way.

No specific statement exists about the definition of a copyright attached to an UML as it is mainly used to exchange information in a transitional way, thus we propose to host such statement as a UML note.

6.6.2 Copyright notice included as a UML note within a UML Diagram

Here is a typical inclusion of the copyright notice into a UML note, placed within a UML Diagram at the root package level:



COPYRIGHT © IEC, 2016. This version of this UML model is part of IEC xxx; see the IEC xxx for full legal notices. In case of any differences between the here-below code and the IEC published content, the here-below definition super sedes the IEC publication; it may contain updates. See history files. See www.iec.ch/CCv1 for copyright details

In case of XMI transposition, the XMI file shall contain the transposition of this UML Note.

In case of draft code component, the « draft » aspect of the code component shall be explicit, under a format to be defined case by case (for example associated to a specific attribute defining the maturity status of the concerned element, This mention will be deleted at the time of official publication.

7 Further updates of the guideline document

The availability of code components opens the possibility for IEC to offer new services to users such as:

- Ability to get access to full content
- Automatic notification in case of change/update
- Browsing capabilities of the whole content of code components from the different parts of the series (so called web-access).
- Change tracking
- ...

These new services may lead to the need to update the current guide

DRAFT

Annex A Copyright Notice

“The purchase of this IEC standard carries a copyright license for the purchaser to sell software containing Code Components from this standard directly to end users and to end users via distributors, subject to IEC software licensing conditions, which can be found here: www.iec.ch/CCv1”

DRAFT

Annex B Software EULA

Code Components in IEC standards (International Standards, Technical Specifications or Technical Reports) which have been identified and approved for licensing, are licensed subject to the following conditions:

- Redistributions of software must retain the Copyright Notice, this list of conditions and the disclaimer below (“Disclaimer”).
- The software license extends to modifications permitted under the relevant IEC standard.
- The software license extends to clarifications and corrections approved by IEC.
- Neither the name of IEC, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission. The relevant IEC standard may be referenced when claiming compliance with the relevant IEC standard.
- The user of Code Components shall attribute each such Code Component to IEC and identify the IEC standard from which it is taken. Such attribution (e.g., “This code was derived from IEC [insert {RefStandard}] within modifications permitted in the relevant IEC standard. Please reproduce this note if possible.”), may be placed in the code itself or any other reasonable location.

Code Components means components included in IEC standards that are intended to be directly processed by a computer and also includes any text found between the markers <CODE BEGINS> and <CODE ENDS>, or otherwise clearly labelled in this standard as a Code Component. **A list of common Code Components is at <source to be included>**

The Disclaimer is:

EACH OF THE CODE COMPONENTS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE CODE COMPONENTS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DRAFT

Annex C Using draft code components

C.1 General

This annex defines the rights and duties attached to the usage of draft code components, which have not reached the publication stage yet (typically associated to DC, CD, CDV, DTR, DTS, FDIS stages).

C.2 Allowed usage of draft code components

Draft code components shall not be distributed outside the IEC, to avoid market confusion of having multiple “unofficial” versions. Thus when submitted to National Committees for review, they shall not get outside of the IEC experts review.

Thus they can't be used nor integrated in commercial products.

However it is recognised that there is a high value to test the code component in situations close to real, before being standardised, especially to eliminate possible bugs and gaps.

In cases where it is necessary to use draft code components for testing, this is subject to the following conditions:

- Having access to the draft code components shall not create any commercial advantage. The usage of the draft component is restricted to experimentation situations.
- The user is required to provide feedback to the TC on the use of the draft code components, including any bugs and issues encountered.
- The draft code components shall not be distributed beyond the test environment.
- The draft code components shall be removed when the final publication is released with its final code components.

In this situation the IEC is not liable in any matter in the usage of such code component and doesn't guarantee any types of filiation between the used draft code components and the officially published one (neither backward compatibility nor forward compatibility). The official publication of the code component may even never happen (due to consensus reaching issues).

C.3 Identifying code component as “draft”

All draft code components shall be explicitly identified as draft code components.

This is done at two levels:

- Within the code component name as expressed in 3.5
- Within the code component content itself under many possible forms – refer to Clause 6

DRAFT

Annex DManifest XSD schemas

D.1 General

These schemas are structured in a way the IEC copyright XML definition may be re-used in other XML files than the Manifest.xml file

D.2 IECCopyright.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="0.1">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      COPYRIGHT (c) IEC, 2017. This version of this XSD is part of IEC copyright definition.
    </xs:documentation>
    <xs:documentation xml:lang="en">
      Definition of elements used for copyright declaration in a code component XML file
      Draft 2016-11-23 (version 0.1).

      To be imported in other code component XSD to enable the usage of a copyright.
    </xs:documentation>
  </xs:annotation>
  <!-- ===== -->
  <xs:simpleType name="tLicenseKind">
    <xs:annotation>
      <xs:documentation>Kind of license</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:Name">
      <xs:enumeration value="Standard"/>
      <xs:enumeration value="Private"/>
      <xs:enumeration value="None"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- ===== -->
  <xs:complexType name="tCopyrighted">
    <xs:annotation>
      <xs:documentation>Add description for a copyrighted XML.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Copyright" type="tCopyrightNotice" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>The copyright notice attached to the
XML</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tCopyrightNotice">
    <xs:annotation>
      <xs:documentation>Copyright notice definition.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Notice" type="tNotice" minOccurs="1" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>The textual copyright notice.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="License" type="tLicense" minOccurs="1" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>The license definition.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tNotice" mixed="true">
    <xs:annotation>
      <xs:documentation>Textual notice</xs:documentation>
    </xs:annotation>
  </xs:complexType>
  <xs:complexType name="tLicense" mixed="true">
    <xs:annotation>
      <xs:documentation>Describe the license definition.</xs:documentation>
    </xs:annotation>
  </xs:complexType>
```

DRAFT

```
<xs:attribute name="uri" type="xs:normalizedString">
  <xs:annotation>
    <xs:documentation>path to the license file</xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="kind" type="tLicenseKind">
  <xs:annotation>
    <xs:documentation>kind of license</xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>
<!-- ===== -->
<xs:element name="Copyright" type="tCopyrightNotice">
  <xs:annotation>
    <xs:documentation>The copyright notice attached to the XML</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:schema>
```

D.3 Manifest.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.iec.ch/CC/2017/IECManifest" xmlns:cc="http://www.iec.ch/CC/2017/IECManifest"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.iec.ch/CC/2017/IECManifest"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="0.2">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      COPYRIGHT (c) IEC, 2017. This version of this XSD is part of IEC copyright definition.
    </xs:documentation>
    <xs:documentation xml:lang="en">
      IEC code components will be delivered as a package including a manifest file. This Schema will describe
      what a package will contains
      Draft 2017-01-18 (version 0.2).
    </xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="IECCopyright.xsd"/>
  <!-- ===== -->
  <xs:simpleType name="tContentKind">
    <xs:annotation>
      <xs:documentation>Kind of content</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:Name">
      <xs:enumeration value="full"/>
      <xs:enumeration value="light"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tCategoryKind">
    <xs:annotation>
      <xs:documentation>Kind of category</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:Name">
      <xs:enumeration value="informative"/>
      <xs:enumeration value="normative"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- ===== -->
  <xs:complexType name="tNamedElement">
    <xs:annotation>
      <xs:documentation>description of one element of the code component</xs:documentation>
    </xs:annotation>
    <xs:attribute name="name" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation>Name of the element</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="comment" type="xs:string">
      <xs:annotation>
        <xs:documentation>Free comment to describe the element</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="tFile">
    <xs:annotation>
      <xs:documentation>description of one file of the code component package</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="tNamedElement">
```

DRAFT

```
<xs:attribute name="content" type="tContentKind" use="required">
  <xs:annotation>
    <xs:documentation>Type of content (full descriptive, or light for
distribution)</xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="category" type="tCategoryKind" use="required">
  <xs:annotation>
    <xs:documentation>Category of the element, if it is normative or
informative</xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="tPublication">
  <xs:annotation>
    <xs:documentation>description of the publication where the code component apply</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="tNamedElement"/>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tHistoryFile">
  <xs:annotation>
    <xs:documentation>description of one history file of the code component</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="tNamedElement">
      <xs:attribute name="startingDate" type="xs:date" use="required">
        <xs:annotation>
          <xs:documentation>date when the history starts</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="endingDate" type="xs:date" use="required">
        <xs:annotation>
          <xs:documentation>date when the history ends</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="startingVersion" type="xs:string" use="required">
        <xs:annotation>
          <xs:documentation>identification of the version when the history
starts</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="endingVersion" type="xs:string" use="required">
        <xs:annotation>
          <xs:documentation>identification of the version when the history
ends</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tCodeComponent">
  <xs:annotation>
    <xs:documentation>Listing of the files composing the code component of the
package</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Publication" type="tPublication" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="File" type="tFile" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="HistoryFile" type="tHistoryFile" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation>Name of the code component</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="id" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation>Identifier of the code component</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="content" type="tContentKind" >
    <xs:annotation>
      <xs:documentation>Type of content (full descriptive, or light for
distribution)</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

DRAFT

```
</xs:attribute>
<xs:attribute name="date" type="xs:date" use="required">
  <xs:annotation>
    <xs:documentation>date of the code component</xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>
<xs:complexType name="tManifest">
  <xs:annotation>
    <xs:documentation>Description the content of a package.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="tCopyrighted">
      <xs:sequence>
        <xs:element name="CodeComponent" type="tCodeComponent" minOccurs="1"
maxOccurs="1">
          <xs:annotation>
            <xs:documentation>The definition of the code components
included in the package</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!-- ===== -->
<xs:element name="IECManifest" type="tManifest">
  <xs:annotation>
    <xs:documentation>The manifest of a package</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:schema>
```


DRAFT

Annex E XML schema to embed in order to get checkable copyright notice presence into IEC code component of XML type

As presented in clause 6 it is possible to add a rule to request XML code component defined by IEC to host a placeholder for copyright details.

This will be done by including the copyright schema file IECCopyright.xsd as defined in D.1 and extend the "tCopyrighted" element to the original XSD schema of the concerned XML code component.

Typically, this would have this form, as for the NSD schema used for previous example:

```
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified" version="0.8">
  <xs:include schemaLocation="IECCopyright.xsd"/>
  <xs:complexType name="tNS">
    <xs:complexContent>
      <xs:extension base="tCopyrighted">
        <xs:sequence>
          [XSD definition]
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```