CADDY-*xml* Format Specification

Version 03.08.01

An XML-based Interchange Format

for Pesticides Registration Applications

Issued by the eSEG (electronic Submission Expert Group) within the

European Crop Protection Association (ECPA)

www.esubmission.ecpa.eu

Status: Final, Sept 17th, 2019

Copyright © 2019 ECPA

Table of Contents

1   About the CADDY-*xml* Specification

1.1   Background

This document describes CADDY-*xml* , an electronic dossier interchange and archiving format for modifications of plant protection products. CADDY stands for **C**omputer **A**ided **D**ossier and **D**ata suppl**Y** and was introduced in 1996. The CADDY-*xml* standard is an extension of the current CADDY 2.0 specification and is intended to support the demand for standardized data-exchange through the **[XML]** format between the applicant and the national competent authority. CADDY-*xml* is additionally designed to adequately support the product specific lifecycle.

Compared to CADDY 2.0, CADDY-*xml* is extended by a lifecycle management mechanism, helping notifiers and authorities to reduce costs and shorten the evaluation time.

1.2   Reading of this Specification

The specification has been written with various modes of presentation in mind. In case of a discrepancy, the online electronic version is considered the authoritative version of the document.

This document uses the terms **must**, **must not**, **required**, **shall**, **shall not**, **recommended**, **should**, **should not**, **may**, and **optional** in accord with **[RFC 2119]**.

1.3   How the Specification is Organized

The specification is organized into the following chapters:

Chapters 1 and 2

The introduction outlines the design principles and includes a brief tutorial on CADDY-*xml* . The second chapter gives an overview of general considerations to consider for the submission of a CADDY-*xml* dossier.

Chapters 3 to 5

CADDY-*xml* reference manual. These chapters describe the XML elements directory structure and more.

Appendices

The appendices contain normative references for CADDY-*xml* , information on references, and other useful information.

1.4 Documentation Conventions

The following typographical conventions are used to present technical material in this document.

Official terms are defined in the following manner: [Definition: You can find most **terms** in chapter 6.5 References]. Links to **terms** may be specially highlighted where necessary.

The XML representations of various elements within CADDY-*xml* are presented using the syntax for Abstract Modules in XHTML Modularization.

Examples are set off typographically:

Example: Example item

```
Example Item
```

References to external documents appear as follows: **[REF]** with links to the references section of this document.

Sample Reference

Reference - linked to from above.

The following typesetting convention is used for non-normative commentary:

Note:

A gentle explanation to readers.

1.5 Change history

This version of the CADDY-*xml* specification stands for its own. It will be as much compatible with former versions as possible but is it not kept downward compatible to keep it lean. Changes are only made where there are helpful for the overall usage but may result in conflicts. See 6.5 for details of the changes.

2 Introduction to CADDY-*xml*

2.1 Overview

CADDY-*xml* supports everything needed to submit PPP electronic dossiers to authorities. It maintains general information, a table of contents, confidential, and non-confidential documents, report information and hyperlink information. Additionally, a CADDY-*xml* based dossier can be instantly viewed with a common web-browser supported by Acrobat Reader.

This standard is flexible to accommodate to different dossier structures such as OECD and EU based standards.

A CADDY-*xml* submission consists out of a well-defined directory structure and an XML file that describes the meta-information of the submission. The XML-file can be thought of as the backbone of the dossier. It references every document that is included within the directory structure.

In a nutshell CADDY-*xml* is composed of product information, a table of content and a list of documents.
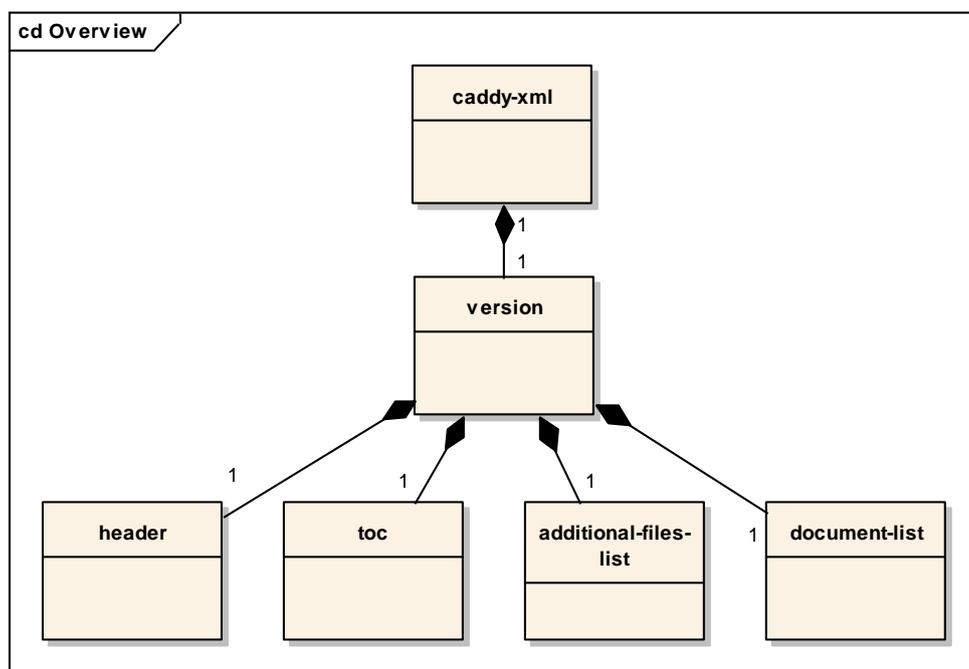
Figure 1: XML-Backbone Overview

The product information, called header, provides a summary describing the product. It consists of optional dossier-related attributes, product, active substance, concentration and company information. The table of contents lists the chapters hence the logical structure of the documents. Any table of contents item has its numbering-scheme and name. A version history list gives information about all the changes made compared to the former submission version. A list of documents and reports that

contains references to the files and to optional report data or attached files is an additional component of the CADDY-*xml* specification. See Figure 1: XML-Backbone Overview.
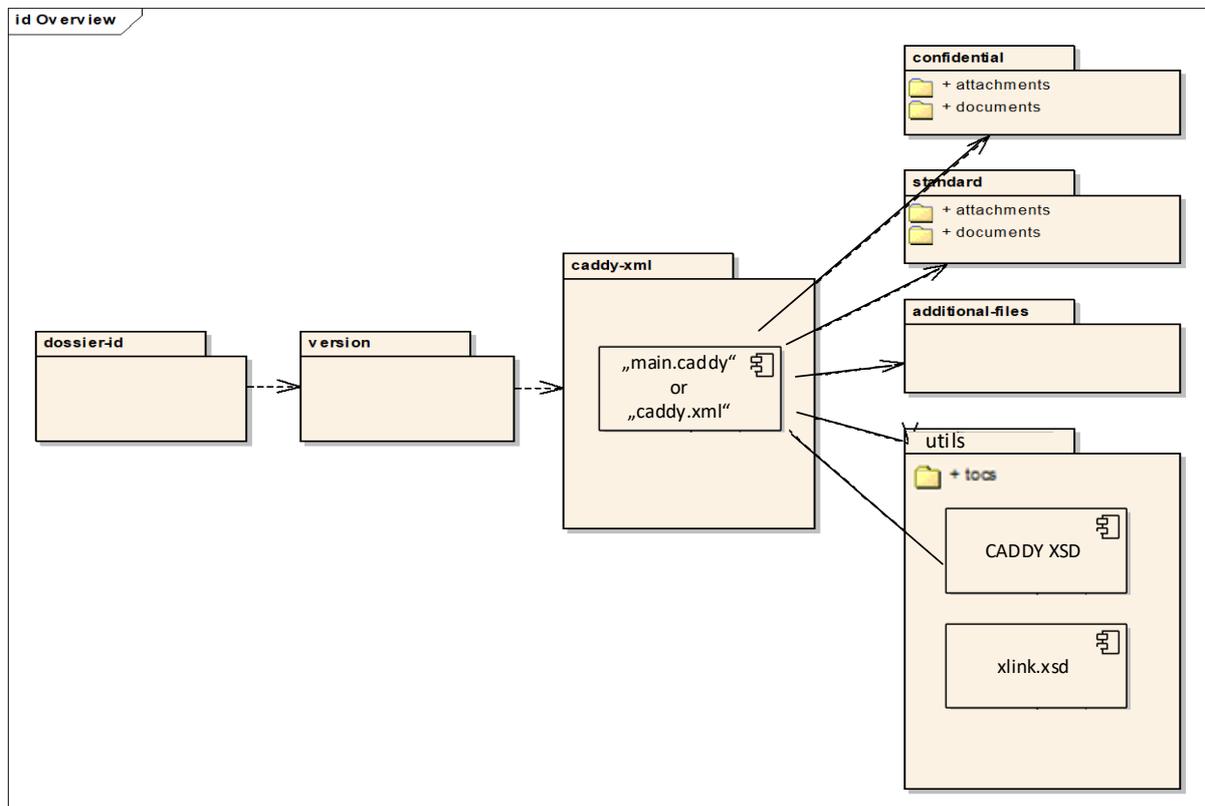
Figure 2: File-System Organization

In general, a dossier contains the XML backbone-file combining all information mentioned above and the submitted documents. The files are organized in a file-system directory structure allowing the differentiation between confidential and standard documents. The entry point for each submission is the CADDY XML backbone file, called **"main.caddy" or caddy.xml**. See Figure 2: File-System Organization

CADDY-*xml* is document-based where the preferred file format is PDF. On the other hand, replacing, deleting, or replacing single pages or appending pages cannot be done transparently. Each of these actions will result in a replaced document. Hence a reviewer may not be able to see the changes instantly. Therefore, in CADDY-*xml* each document has a status in the XML backbone file and when there is a change there is a comment representing the changes made to the document. Additionally, the version number where the document was introduced is also noted, so recent changes can be identified easily.

2.2 Overview and versioning of CADDY-xml specification artefacts

The CADDY-xml specification consists of a set of artefacts that can have a separate lifecycle and that are therefore versioned separately:

Name	Description	Versioning
CADDY-*xml* format specification	The document you are currently looking at	For the current format specification version please see the header page of this specification document, e.g. "03.08.00". The versioning scheme of the specification is specified in the type xmlVersionNumberType in chapter 4.18.1.
CADDY-*xml* Backbone XSD	XSD describes the structure of the XML-Backbone (see chapter 4)	See attribute "version" of the schema element in the XSD. Example: `<xs:schema xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xs="http://www.w3.org/2001/XMLSchema" version="03.06.00">`.
CADDY-*xml* TOC XSD	XSD that describes the structure of the Table of Contents, used for validation (see chapter 5)	See attribute "version" of the schema element in the XSD. Example: `<xs:schema xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xs="http://www.w3.org/2001/XMLSchema" version="03.06.00">`.

Table 1 – CADDY-xml specification artefacts and their versioning

Please note that there is separate versioning of the backbone XML file itself. The attribute xmlversion of the caddy-xml element (see 4.1.2) will denote the version of the CADDY-xml specification it refers to. This specification version is also displayed in the viewer module in the upper right corner to ease communication during support.

For more details please see the webpage http://esubmission.ecpa.eu

2.2.1 Information about matching versions

The information what version of one artefact is compliant with some other version of another artefact (e.g. what Backbone XSD version supports what specification version) is listed on the CADDY website starting with version 03.06.00 of the format specification. To ease communication all artefacts that are adapted to work with format specification 03.06.00 are also initially set to a version 03.06.00. From now on the artefacts may have different lifecycle and version numbers may develop independently.

2.2.2 Further related CADDY-xml artefacts

Next to the artefacts belonging to the CADDY-xml format specification itself there are other artefacts that further support the use of CADDY-xml. They are not part of the specification, but are listed here for completeness:

Name	Description	Versioning

CADDY-*xml* viewer module	Software that allows to display the CADDY-*xml* submission in the ECPA eSubmission Viewer	The CADDY-xml viewer module is a piece of software that is used in conjunction with the ECPA eSubmission viewer. It is not part of the submission package but bundled with the eSubmission Viewer. The CADDY-xml viewer module version is displayed in the upper right corner of the main screen. Prior to the specification version 03.08.00 the viewing capability was delivered with a viewer that was part of the submission package and displayed in the default local web browser. This viewer is referred to as "outdated embedded viewer" in this specification.
CADDY Converter	software that converts legacy CADDY v2 dossiers into CADDY-*xml* dossiers	The CADDY converter displays its version in the title bar. Please note that the generated into CADDY-*xml* dossier will contain the backbone XSD, the TOC XSD and the outdated embedded viewer. Those versions are currently bundled with the CADDY converter version.
eSubmission validator	software that checks a dossier for consistency with this specification	The eSubmission validator displays its version in the title bar and the about dialog. The eSubmission viewer can run different types of CADDY-xml consistency checks by choosing different validation rulesets. The eSubmission validator includes a check for compliance with the CADDY-xml PDF/A profile , but is not considered to be a full PDF/A validation.
CADDY-xml validation ruleset	Validation rule definition files used in conjunction with the eSubmission validator	The validation rules carry their own versioning. Different types of validation rules can be used in the eSubmission validator at the same time. The versions of available validation rules are displayed in the respective control of the eSubmission validator.

Table 2 – Further CADDY-xml related artefacts (not part of this specification)

For more details please see the webpage http://esubmission.ecpa.eu

3 General Considerations

This chapter discusses the various aspects of the CADDY-*xml* in detail. The focus is set to the explanation of the CADDY-*xml* aspects. The following chapter is meant to be a reference showing all the details.

3.1 The XML-Backbone

As previously mentioned the XML-backbone represents a file that stores all relevant data of the dossier, except the documents and attached files. This file is placed in the root directory of each version of a dossier (See **[XML]**). The file format is XML, a widely used and well understood standard for data-exchange.

Each version of a dossier is represented with a separate XML-backbone file that covers all information of the version. This allows viewing the current version without reference to previous versions.

The XML-backbone file contains the following groups of information:

3.1.1 Definition

Any XML file has a definition file that defines its structure. This structure is defined using the **[XML Schema]** definition language. This file also defines the rules for CADDY-*xml* . Hence each XML file must reference the XML definition file. This reference is one of the first instructions.
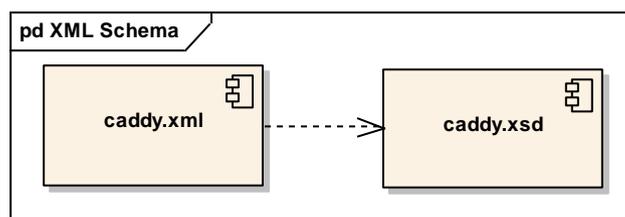
Figure 3: XML Schema

3.1.2 Specification Version

To allow the CADDY-*xml* standard to evolve a specification version stored in the topmost element allows a mapping of the correct XSD-version used for validation and structure enforcement independently from the delivered XSD-file.

This specification version is introduced with the CADDY-*xml* standard version. If no specification version is given the former CADDY-*xml* version 1.0.1 is assumed as the default.

Having a specification version allows that the different CADDY-*xml* versions may not be up or downward compatible to keep the standard lean.

3.1.3 Version

The central element in CADDY-*xml* is the version. The version groups all other information. A submission uses a version numbering scheme that allows distinguishing between major and minor versions. By definition a submission with a minor version equal to zero must contain all documents. A submission with a minor version above zero must contain at least all changed documents.

3.1.4 Header

A header groups the information for a dossier version. These are title, authority, company, product information, active substances and their concentration. Versioning of the header information allows determining changed values.
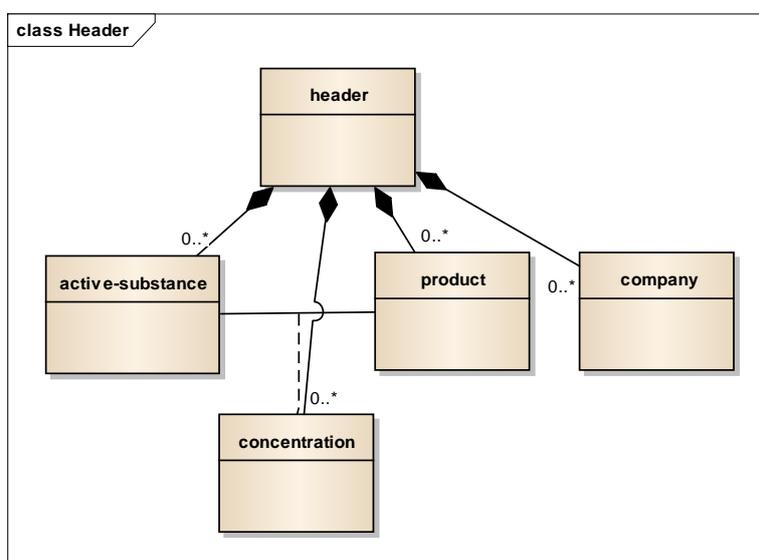
Figure 4: Header

The concentration is semantically the relation between the active substance and the product. Syntactically, it is a separate XML element.

3.1.5 Table of Contents

The table of contents describes the structure of the submitted dossier. Basically any hierarchical structure can be expressed within CADDY-*xml* .

CADDY-*xml* may reference a Standard TOC (See 3.2). Such Standard TOCs are defined in extra XML files which can be used by validators to check the structure of the submitted table of contents. A reference to a Standard TOC is optional but may be mandatory for some regulatory authorities.

A table of contents element consists of a unique ID, a structuring number (e.g. Annex point) and heading title (e.g. section title). A table of contents element may contain other table of contents elements as a child.
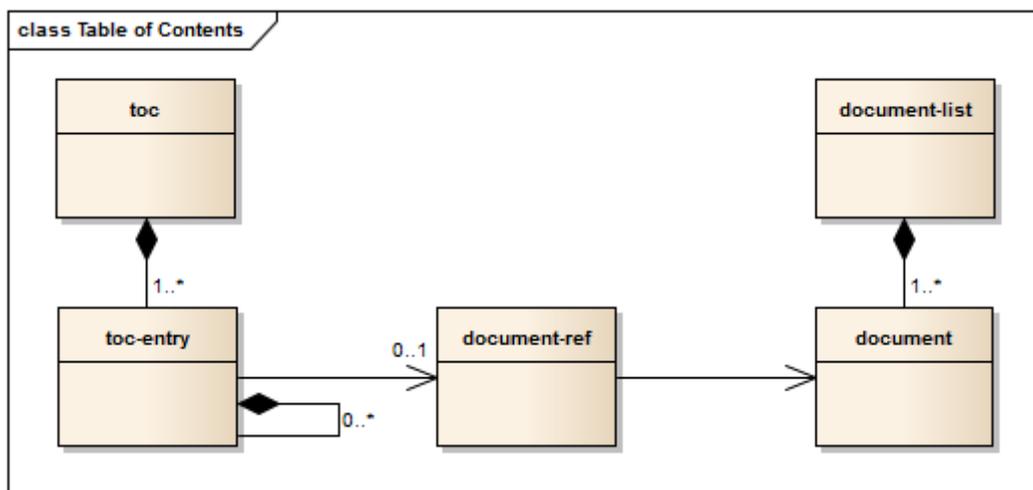
Figure 5: Table of Content

Each table of contents element may also have a reference to one document. From several table of contents elements CADDY-*xml* allows to reference to the same document.

To allow a better control about an empty document reference a table of content element can optionally flagged as 'intentionally left blank'. Such a TOC entry explicitly must not have a document reference and shall be treated as "not missing" by validators or reviewers. Instead these programs shall generate an empty "Intentionally left blank" page showing the optional comment.

3.1.6 Documents

The XML-Backbone maintains a list of documents. This list is separated from the table of content to support one document in different table of content sections. For archiving reasons the document must be a PDF or an XML document. Documents cannot be included in an XML file, so the location within the file system is also noted in the XML-backbone. See Figure 6: Document List
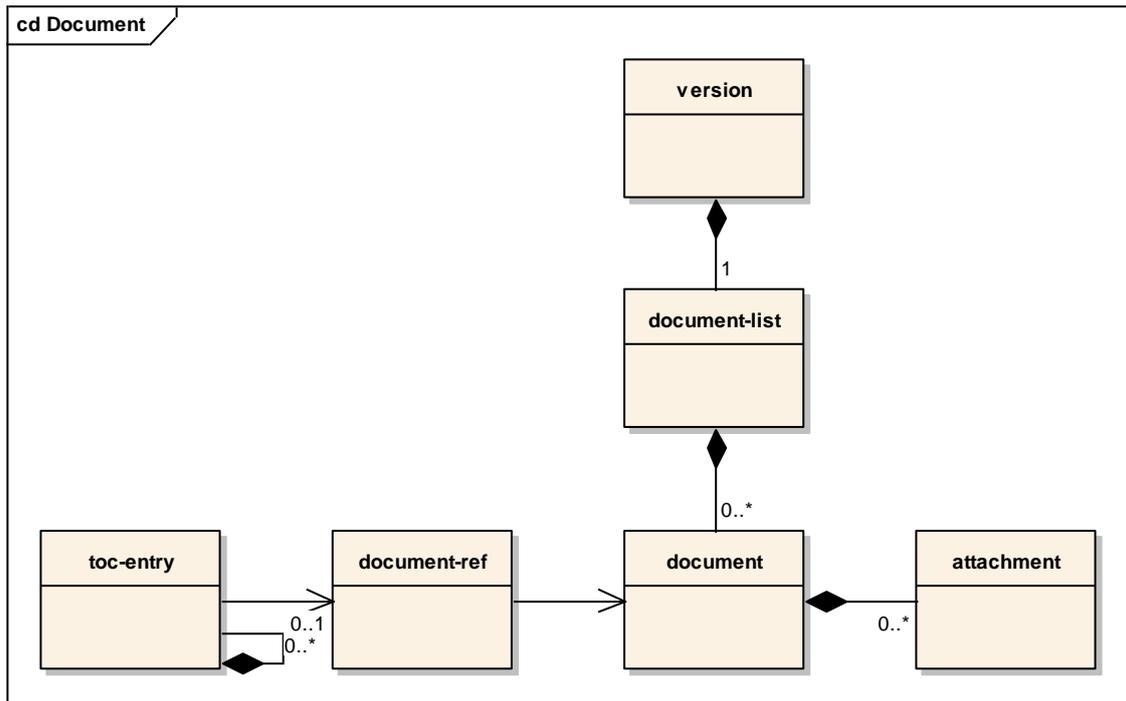
Figure 6: Document List

A document can be referenced by multiple ToC entries in the same version (also called multiple inclusion). In this case the document must be included multiple times, as part of the dossier metadata in the report-data element is considered to be ToC node specific (e.g. dossierFilenumber, see chapter 4.14.2). This means that

a) the XML-backbone contains multiple instances of the document and attachment element describing the same document with the same metadata, differentiated only by different unique IDs.

b) the referenced files are contained multiple times with different file names in the folders

For document lifecycle reasons and for review purposes the document has states and a version number representing the dossier version it was introduced.

CADDY-*xml* introduces a document operation. The document operation introduced by the **[eCTD]** standard gives the opportunity to support reviewers with an assignment of the intention of this document. Supported operations are **new**, **deleted** and **replaced**. To explicate the operation an optional comment should be filled out.

Access to documents can be controlled via the confidential attribute that must be either set to be true or false. Confidential files are stored separately from standard files. See 3.4

To ensure data integrity referred to in section 3.5, a checksum exists that gives a document a unique identifier. The checksum algorithm that must be used is the MD5 code. **[RFC 1321]** This will identify that the document is the same.

The directory structure for documents is defined in section 3.4. For file formats or names, refer to 3.8.

Optionally a document can be described in a formal way using report data, referencing several attributes like authors, document date, owner, and the flags GLP, published and data protection.

3.1.7 Attachments

Documents may refer to additional files that are called attachments in CADDY-*xml* (CADDY 2.0 refers to these files as CADDY controlled files). These files are meant for several purposes. An optional comment can be defined. The purpose values are defined in section 3.8.2.

Each attachment file can be uniquely associated to one document. The access to attachments is controlled via the confidential flag of the according document. Hence confidential documents imply confidential attachments.

The directory structure for attachments is defined in section 3.4. For file formats or names refer to 3.8.2.

3.1.8 Additional Files

Submissions based on CADDY 1.1 and CADDY 2.0 may reference additional files to the submission for review purposes. CADDY-*xml* may also contain a helper directory called "additional files". This directory can be seen as transfer container to submit files which do not belong to specific documents and must not be treated as attachments.

3.1.9 Hyperlinks

Each table of content entry may contain hyperlinks guiding reviewers through the submission. A hyperlink is defined having a source and a destination. Compared to e.g. web-pages having the hyperlink within the document (implicit), CADDY-*xml* uses explicit hyperlinks. Due to this definition, CADDY-*xml* doesn't store hyperlinks within a document. Changes to the dossier table of contents do not enforce replacing of documents, like with implicit hyperlinks.
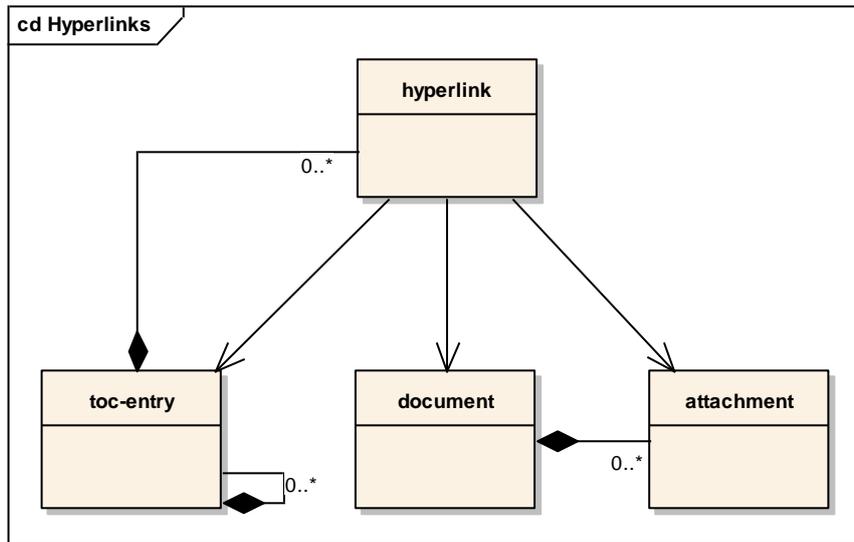
Figure 7: Hyperlinks

Explicitly representing hyperlinks implies a definition of the source. The source can be either the table of content entry itself, a specific document or an attachment. If the source is noted to be a document, the hyperlink must be described additionally with a page number and a rectangular region placed on that page. Each hyperlink must have a title describing the target. Often a section name in the bookmark structure is sufficient. See Figure 7: Hyperlinks.

The destination of a hyperlink can be another table of content entry, a document, or an attachment. Having defined a document as the destination, the precise position can be defined either by a named destination or by a page. Default is the first page of the document. See **[Adobe-Ref]**

Example:

Opens a document at the bookmark "named_destination".

prodlist.pdf#namedest=named_destination

Opens a document on page 12.

prodlist.pdf#page=12

It is up to a CADDY-xml viewer to implement the correct hyperlink navigation functionality.

3.2 Standard Table of Contents

Introduced with CADDY-*xml* specification a Standard Table of Contents (Standard TOC) allows the definition of official table of contents structures. If such an official table is available, the CADDY-*xml* files should reference such a Standard TOC and must then define a table of content which is compliant with the structure the Standard TOC defines. To ensure remaining flexibility the Standard TOC shall define table of content elements as extensible where appropriate and define restrictions for document references.

3.2.1 Definition

Like CADDY-*xml* a Standard TOC has a definition file that defines its structure. This structure is defined using the **[XML Schema]** definition language.
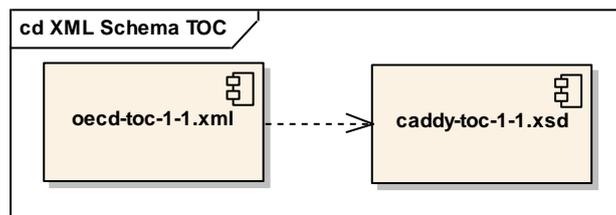
Figure 8: XML Schema

3.2.2 Specification Version

To allow the Standard TOC to evolve it must have a specification version stored in the topmost element. This allows a mapping of the correct XSD-version used for validation and structure enforcement.

Having a specification version the different CADDY-*xml* Standard TOC versions may not be upward or downward compatible to keep the standard lean.

3.2.3 Table of Contents

The table of contents elements describe a structure for a Standard TOC. It consists of a structuring number (e.g. Annex point) and heading title (e.g. section title) but has no Id as there is no relation to other elements. A table of contents element may contain other table of content elements as a child.
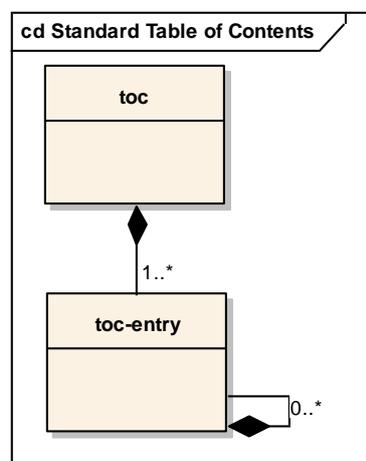
Figure 9: Standard Table of Content

To allow a definition of slightly flexible structures each table of content for a Standard TOC can be defined as extensible which allows additional table of content elements. Furthermore, the occurrence of documents in the dossier must be specified for each table of content entry. Document references shall be marked as **required** (mandatory), **allowed** (optional) or **prohibited** (not allowed at all).

3.2.4 Directories and Filenames

A Standard TOC may be distributed in the *CADDY-xml* directory structure with the directory 'tocs'. As validators shall use the official versions of Standard TOCs the file will usually not be referenced. Transmitting the used valid Standard TOC only allows a convenient inspection (via a compatible Web browser) of the used standard.

The filename of a Standard TOC should have the TOC version number within the filename e.g.: OECD-CADDY-xml-TOC-01-00.xml.

3.2.5 Validity and Distribution of Standard TOCs

It is planned to make standard TOCs available on the official ECPA website in the XML format defined in chapter 5. A Standard TOC version must be treated effective (i.e. allowed for usage) as long as the correct date is between the "validFrom" and "validTo" dates specified in the header. An attribute "substituteVersion" will allow a replacement of a previous effective version. Such a replacement should be recognized and the new Standard TOC should be used instead of the previous one. Only an effective Standard TOC version can replace a previous version. As long as the new version is not effective the replacement is only a hint for the future of the Standard TOC.

3.3 Version Handling of Submissions

The CADDY-*xml* standard supports two modes to submit a dossier: complete and incremental. Complete submissions must contain all documents, incremental only those that have been changed from the former version. To simplify the creation of review software and to deal with removed or modified hyperlinks the TOC structure must always be included completely.

Deleted files will not be part of the TOC structure any more. Hyperlinks defined in previous submissions of such effected toc entries shall be removed.

Due to processes that are well defined between submitter and authority, the first submission evolves with the help of a rapporteur country. During this evolvement only the minor part of the version number will be advanced and only the newly added or changed files will be transferred.

On the other hand it may be favourable to submit complete dossiers as one version instead of submitting all versions. With creation of a new major version all files will be placed in the same version

directory, regardless in with version they were originally submitted. Multiple times replaced files will only be submitted in their last version but deleted files will not be included in a submission. To allow a simple recognition of deleted files, the former document element shall be kept in the document-list with the filename of the last submission.

CADDY-*xml* defines to use incremental submissions for subversion unequal zero and complete submissions if the subversion is zero.

See Appendix (section 6.4) for a sample dossier which demonstrates the different aspects of versioning.

3.4 Directory Structure and Volume Handling

The submission of a CADDY-*xml* dossier may be transferred to the authorities on any kind of electronic media agreed upon with the agency. The authority is responsible to copy the submission into the file system or into an appropriate CADDY-*xml* review system. All confidential documents are placed in a separate folder and can be separated from the standard documents.

The root folder of a submission must be named with the unique dossier ID. See chapter 4.18.9. It is recommended to submit all documents of a CADDY-*xml* version with one media.

A submission may reference documents or attachments from previous versions. To support this all versions of a dossier must be placed within the same folder. The folder name must be the same as the version attribute of the submission.
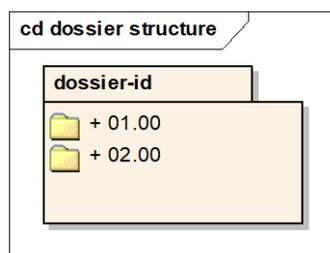
Figure 10: Dossier Structure

The folder for each version must contain the file system structure outlined in Figure 11: Version Structure. The 'standard' and 'confidential' folders host the reports and attachments. The 'additional-files' folder contains the files that are meant to support the whole dossier. The 'utils' folder contains the CADDY XML schema file and the referenced xlink schema file.

All filenames and relative paths are restricted to a maximum summarized length of 230 characters. It is highly recommended to not have a length of more than 200 characters to allow the storage of dossiers in an existing directory and still not override the total path length of 255 characters to allow an access in file systems which restrict the length of filenames and/or paths.

Next to the documents folders the following main files within the version folder and the utils subfolder are essential – please note that some are version-dependent in their file naming (see 2.2).

File	Path	Mandatory	Description
"main.caddy" or "caddy.xml" (Backbone XML)	version	Yes	The XML backbone file containing the information of the submission version. The file name must be either "main.caddy" or "caddy.xml" and must not be altered. Using "main.caddy" allows to open the respective CADDY-xml dossier in the eSubmission viewer by double-clicking on it.
"caddy.html"	version	Yes	Simple standalone display of the dossier content in human-readable format, generated from the XML backbone file by the supplied XSLT. See chapter 6.2.1 for more information. The file name must not be altered.
caddy_03-07-00.xsd (Example)	version/utils	Yes	The XML Schema definition file that validates the submission and its structure. The filename has to contain the version number as present in the version attribute of the XSD itself. The XSD is referenced in the attribute `xsi:noNamespaceSchemaLocation` in the element caddy-xml (see chapter 4.1.4). The name of the XSD file may vary according to the version of the XSD. Please see the CADDY website for the compatibility between specification document and XSD.
"xlink.xsd"	yersion/utils	No	standard xlink XSD referenced from the CADDY XSD

Some files from the above list are not mandatory for a valid CADDY-*xml* submission: The version folder must only contain the backbone XML file and the caddy.html file. The utils folder must contain the CADDY XSD file. All other files in the utils folder and the toc subfolder are optional.

For the filenames that must be used for documents, attachments or additional files refer to section 3.7 which describes the nomenclature in detail.

The following figure summarizes the folder structure within one submission version.
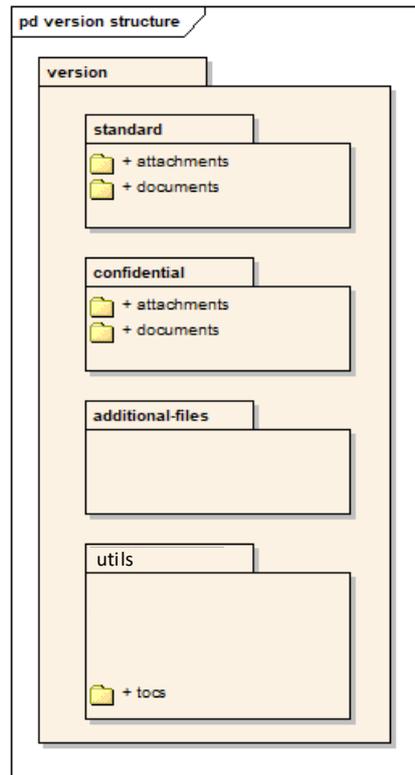
Figure 11: Version Structure

3.5 Document Integrity

Documents, attachments and additional files are protected with checksums. The intention of this checksum is to verify the uniqueness of a file. Checksums are mandatory for all files contained within the current submission. They are optional for files that are part of a preceding full submission and are referenced from the current incremental submission. The algorithm used for the checksum generation is described in **[RFC 1321]**. Its purpose is to identify the content of a file to be unique. Any change within a file will result in a different checksum. This allows the submitter and authorities to protect from unintentionally changed files.

3.6 Document Status

CADDY-*xml* defines states that allow identifying the changes that have been made to a document.

Operation	Description
new	If a document is newly added to a submission
deleted	If a document has been deleted
replaced	If a document has been replaced by another document

3.7 File References

The references to files within the submission is done with the xlink:href attribute, see **[XLink]**. The URL that is used for reference must contain a relative path including the submission version where the document was submitted last. This is independent to the attributes addedVersion or changedVersion. The relative path always starts from the submission version directory where the 'caddy.xml' file is placed.

The folder names below the folders "standard", "confidential" and "additional-files" have to follow the following regular expression:

`([a-zA-Z0-9_\-\.]+)`

The file names for the files in those folders have to follow the same following regular expression:

`([a-zA-Z0-9_\-\.]+)`

E.g.: 3.1.1_Laudis_WG_dRR_Part_C_16.pdf

The complete pattern for relative paths to be used in file references is

`<xs:pattern value="(((\.\.){1}/)*)?(([a-zA-Z0-9_\-\.]+)/)*([a-zA-Z0-9_\-\.]+)"/>`

The pattern will be checked as part of the validation ruleset checks. It is not part of the XSD definition. The complete length of a relative path as present in the XML file must not exceed 230 characters. Major versions are always submitted completely. It is not mandatory, but advisable that the file name of a specific unchanged document in the submission remains identical between major versions.

If a document content changes it is advisable to change the file name as well. This is also not mandatory, the only relevant information about lifecycle is maintained in the XML, not in the filename.

Example:

Document A (DocA.pdf) has been changed in version 1.1 but document B (DocB.pdf) is unchanged. Relative paths (without specification of access protocol i.e. file:// or http://) shall be used to allow that CADDY-*xml* dossiers can be viewed on web servers as well as on local storage.

Version 1.0:

`<... xlink:href="../`**`01.00`**`/standard/documents/DocA.pdf" .../>`

`<... xlink:href="../`**`01.00`**`/standard/documents/DocB.pdf" .../>`

Version 1.1:

`<... xlink:href="../`**`01.01`**`/standard/documents/DocA_V2.pdf" .../>`

`<... xlink:href="../`**`01.00`**`/standard/documents/DocB.pdf" .../>`

Version 1.2:

`<... xlink:href="../`**`01.01`**`/standard/documents/DocA_V2.pdf" .../>`

`<... xlink:href="../`**`01.00`**`/standard/documents/DocB.pdf" .../>`

Version 2.0: (Keep in mind that major versions are always submitted completely)

```
<... xlink:href="../02.00/standard/documents/DocA_V2.pdf" .../>
<... xlink:href="../02.00/standard/documents/DocB.pdf" .../>
```

3.8 Common File Formats

This chapter defines the constraints that submitted CADDY-*xml* files must conform to. CADDY-*xml* distinguishes between three kinds of documents or files. The first are the original documents that are referenced directly from the table of content. The second are attachments that come with an original document for several reasons. The third are the dossier related additional files which do not belong a single original document.

3.8.1 Original Documents

The original document that is submitted must follow certain rules to improve readability and compatibility across different platforms. These files must be either in PDF format or in XML format.

PDF Documents:

To achieve a normative, readable, reproducible and usable form for PDF documents several features of the PDF standard must be used in a well-defined manner. Additionally the graphics resolution of the contained graphics must be restricted to a feasible size to reduce the dossier size and to enforce fast access to the documents.

The file of an original document in PDF format must be placed into the 'documents' folder or a subfolder of this folder. The filenames does not need to contain the ID in its filename.

PDF files in this section must be conform to the "CADDY-xml PDF profile" described below, which is based on the PDF/A-1b standard described in **[ISO-19005-1]**, which restricts PDF in a way that it is optimized for exchange and reproducibility of the content.

The profile defines additional constraints on top of the PDF/A-1b standard and partly relieves constraints from the base standard. The profile is not a complete definition of the file format that allows strict format verification, but a set of recommendations.

The following table defines the CADDY-xml PDF profile in detail.

	CADDY-xml PDF profile
Allowed	External references Restriction: Only hyperlinks to computer files that are part of the current submission package in the content folder (see chapter 3.4) are to be used. The hyperlinks in the PDF files should adhere to the relative path definition in chapter 3.7. Hyperlinks that refer to individual chapters or pages of a target PDF document are not recommended, as this functionality is not stable.
Recommended	Embedding of all fonts Colour spaces specified in a device-independent manner Inclusion of document structure Language specification Use of metadata
Not recommended	Transparency / overlay Audio / video content Encryption and password protection Digital signatures Embedding of arbitrary file formats Usage of dynamic content (JavaScript) and executable file launches

Table 3 – CADDY-xml PDF profile

XML Documents:

XML files that are referenced directly from the table of contents must be restricted in a way that the content is displayed with the help of an XML-Style sheet. Submitted XML files must have a style sheet that transforms the XML content to **[XHTML]** which will be viewed by the reviewer.

Often these style sheets use supporting files like images. Hence, restrictions for all the files are defined. The XML file itself, the style sheet and all additional files must be placed in a subfolder that may be named with the ID attribute of the document, but other names for the folder are also permitted. Within this folder subfolders may be used, if necessary (See 3.4 for length restrictions). All used folder names and filenames should prevent the use of special characters outside of the valid characters for file references (see 3.7).

E.g.: Document with ID 'id012130123' with its style sheet that references two picture files.

01.00/standard/documents/id012130123/report.xml

01.00/standard/documents/id012130123/stylesheet.xsl

01.00/standard/documents/id012130123/picture1.jpeg

01.00/standard/documents/id012130123/picture2.jpeg

3.8.2 Attachments

Attachments within CADDY-*xml* are stored in the attachments subfolder either of the confidential or standard folder. Attachments must follow the above mentioned restrictions of the filename. Attachments of one document may be grouped within one single subfolder. For ease of understanding it is recommended to use the filename of the document as folder name for the attachments of this same document. Within this folder the attachment files do not have limitations for their filenames unless they do not exceed the allowed total path length (See 3.4). Filenames should prevent the use of special characters outside of the valid characters for Original Documents (see 3.7). Mutual agreements between submitter and receiver should be used to define less restrictive rules if necessary.

Example: Document A (idd0001.pdf) has three attachments

01.00/standard/documents/idd0001.pdf

01.00/standard/attachments/idd0001/attachment_number_1.xls

01.00/standard/attachments/idd0001/attachment_number_1.doc

01.00/standard/attachments/idd0001/attachment_number_1.xpt

The attachment must be classified with a type which can be one of the following:

Type	Description
rendition	This attachment is a rendition of the document
appendix	This attachment is an appendix
figure	This attachment is a descriptive figure
photo	This attachment is a descriptive photo
sas-table	This attachment is a SAS file containing the original data
oecd-data	This attachment is a collection of OECD conform XML report description files (e.g. OHT – OECD Harmonized Templates)
zip-file	This attachment is a general ZIP file
other	This attachment does not conform to other types. The comment attribute must be used to describe the attachment type.

3.8.3 Additional Files

Additional files that are dossier related are stored in a folder within the root directory of a submission version. This folder must be named 'additional-files'. Within this folder the files do not have limitations for their filenames unless they do not exceed the allowed total path length (see 3.4). Filenames should prevent the use of special characters outside of the valid characters for Original Documents (see 3.7). Mutual agreements between submitter and receiver shall be used to less restrictive rules if necessary.

3.9 Display of CADDY-*xml*

A CADDY-*xml* submission can be viewed in two ways:

- Using the eSubmission viewer and the associated CADDY-xml viewer module, by opening the file "main.caddy" or "caddy.xml" in the root directory of the submission in the eSubmission viewer. For details please see the website http://esubmission.ecpa.eu. This viewing option provides advanced functionality like searching and filtering.
  If a dossier with specification version 03.07.00 and earlier is opened with the eSubmission viewer, the user can also decide to display the dossier using the outdated embedded CADDY viewer.

- Clicking the file "caddy.html" in the root directory of the submission. This will display a simple static overview of the dossier content in static HTML format (no JavaScript) in the default web browser on the PC.
  This option serves to quickly browse the dossier version when no advanced dossier viewer is at hand.

The XML backbone file is encoded in UTF-8. This means that practically all characters can be used within the XML as attribute or element values, provided their usage is not restricted (e.g. in folder and filenames).

4   The CADDY-*xml* Core Elements

This chapter describes all elements and their associations in detail. Figure 12 gives an overview of the hierarchical structure of the CADDY-*xml* structure. All elements within this section are XML-elements that are placed within one file.
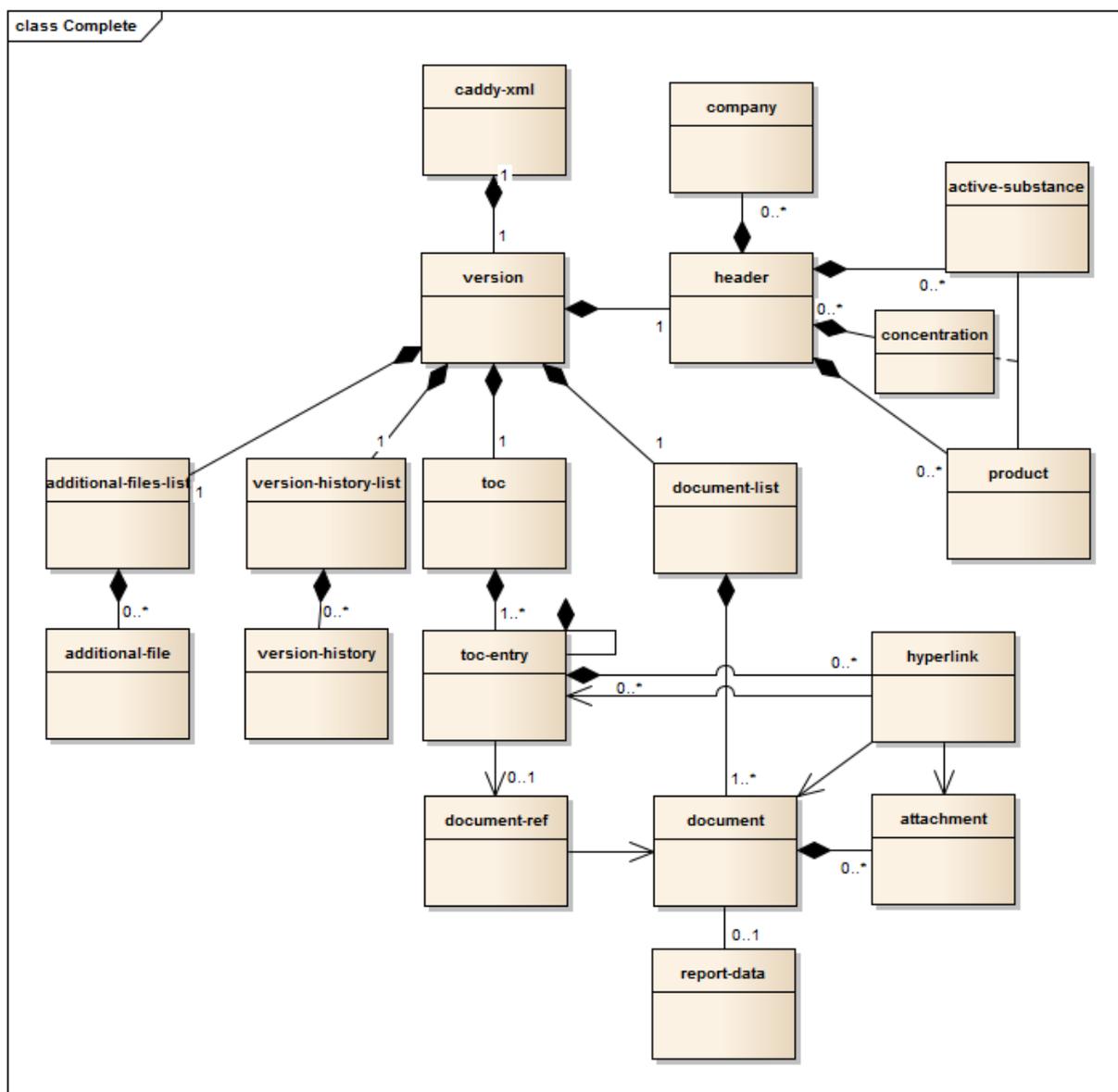
Figure 12: Overview CADDY-xml Elements

4.1 The caddy-xml Element

This element is the root element of all caddy.xml files. The caddy-xml element must have exactly one child, the version element. It does not have attributes.

4.1.1 Child Elements

Elementname	Mandatory	Min	Max	Definition	Description
version	yes	1	1	4.2	Must include this element that describes the submission in detail.

4.1.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
xmlVersion	No	xmlVersionNumber	4.18.1	The version number of the referenced CADDY-*xml* specification and to which the submission adheres. It is composed of a two digit major, a two digit minor and a two digit patch level version number separated by single dots. (e.g. 01.00.01 or 03.00.00). If this attribute is missing the default used is CADDY-*xml* version 1.0.1

4.1.3 Definition

```
<xs:element name="caddy-xml" type="caddy-xml-type"/>
<xs:complexType name="caddy-xml-type">
    <xs:sequence>
        <xs:element ref="version"/>
    </xs:sequence>
    <xs:attribute name="xmlVersion" type="xmlVersionNumber"
    use="optional"/>
</xs:complexType>
```

4.1.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<caddy-xml xmlVersion="03.06.00"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="utils/caddy_03-06-00.xsd"
 xmlns:xlink="http://www.w3.org/1999/xlink">
 <version version="01.00">
 </version>
</caddy-xml>
```

4.2 The version Element

This element is the root of a submission. It is defined to have a header element, a toc element, a document-list element and an additional-files-list as children. It has a version attribute that has to be

unique over the whole lifecycle of the submission. The version is composed of a two digit mayor and a two digit minor version number separated with a dot. E.g. 01.00 or 01.02

4.2.1 Child Elements

Elementname	Mandatory	Min	Max	Definition	Description
header	yes	1	1	4.3	Must include this element that describes the submission in detail.
toc	yes	1	1	4.8	Must include this element that manages the table of content of the submission.
document-list	yes	1	1	4.12	Must include this element that maintains a list of documents/reports.
additional-files-list	yes	1	1	4.16	Must include this element which maintains a list of additional files.

4.2.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
version	yes	versionNumber	4.18.2	The version number of this submission. It is composed of a two digit major and a two digit minor version number separated with a dot. E.g. 01.00 or 01.02
masterDate	yes	xs:date	**[XML Schema]**	Day of mastering the electronic CADDY-*xml* version of the dossier (CADDY compilation date)
issueDate	no	xs:date	**[XML Schema]**	Issue date of the original dossier.

4.2.3 Definition

```
<xs:element name="version" type="version-type"/>
<xs:complexType name="version-type">
     <xs:sequence>
          <xs:element ref="header"/>
          <xs:element ref="toc"/>
          <xs:element ref="document-list"/>
          <xs:element ref="additional-files-list"/>
     </xs:sequence>
     <xs:attribute name="version" type="versionNumber" use="required"/>
     <xs:attribute name="masterDate" type="xs:date" use="required"/>
     <xs:attribute name="issueDate" type="xs:date" use="optional"/>
</xs:complexType>
```

4.2.4 Example

```
<version version="01.00" masterDate="2001-02-01">
 <header .../>
 <toc/>
 <document-list/>
```

```
 <additional-files-list/>
</version>
```

4.3 The header Element

This element describes the global attributes of the version. The header may also have elements for

company, product, active-substance, and concentration.

4.3.1 Child Elements

Element Name	Mandatory	Min	Max	Definition	Description
company	no	0	*	4.4	May include elements that describe the submitting company in detail.
product	no	0	*	4.5	May include elements that describe the submission's product in detail.
active-substance	no	0	*	4.6	May include elements that describe the submission's active substance in detail.
concentration	no	0	*	4.7	May include elements that describe the concentration of an active substance within a product.

4.3.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier unchanged over the whole lifecycle of the dossier. If a new version introduces changes in header attributes it must have a new id.
dossierTitle	yes	string250	4.18.3	Dossier title
dossierSubtitle	no	string250	4.18.3	Dossier subtitle
uniqueDossierID	yes	dossierID	4.18.9	Companies unique dossier id. See 4.18.9
authority	yes	string100	4.18.3	Authority (e.g. EEC)
guideline	yes	string100	4.18.3	Requirement / official guideline
regulation	yes	string100	4.18.3	Regulation (e.g. 3600/92)
rapporteur	yes	IndISO2	4.18.4	European rapporteur member state (alpha-2 code **[ISO 3166-1-alpha-2]**)
changedVersion	no	versionNumber	4.18.2	Represents the version number of the last change to this element.

4.3.3 Definition

```
<xs:element name="header" type="header-type"/>
<xs:complexType name="header-type">
    <xs:sequence>
```

```
            <xs:element ref="company" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="product" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="active-substance" minOccurs="0"
            maxOccurs="unbounded"/>
            <xs:element ref="concentration" minOccurs="0"
            maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:ID" use="required"/>
        <xs:attribute name="dossierTitle" type="string250" use="required"/>
        <xs:attribute name="dossierSubtitle" type="string250"
        use="optional"/>
        <xs:attribute name="uniqueDossierID" type="dossierID"
        use="required"/>
        <xs:attribute name="authority" type="string100" use="required"/>
        <xs:attribute name="guideline" type="string100" use="required"/>
        <xs:attribute name="regulation" type="string100" use="required"/>
        <xs:attribute name="rapporteur" type="lndISO2" use="required"/>
        <xs:attribute name="changedVersion" type="versionNumber"
        use="optional"/>
</xs:complexType>
```

4.3.4 Example

```
<header id="IDH00001"
 dossierTitle="CADDY Version 2 Demonstration Dossier"
 dossierSubtitle="Test dossier for CADDY-xml"
 uniqueDossierID="DOEGB001"
 authority="EU Commission"
 guideline="1663/VI/94, rev. 8"
 regulation="91/414/EEC"
 rapporteur="CH">
<company .../>
<product .../>
<active-substance .../>
<concentration .../>
</header>
```

4.4 The company Element

This element describes the company making the submission.

4.4.1 Child Elements

There are no child elements for this element.

4.4.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier unchanged over the whole lifecycle of the dossier. If a new version introduces changes in company attributes it must have a new id.
name	yes	string100	4.18.3	Company name

code	yes	companyCode	4.18.5	3-digit EU company code or 6-letter US company ID
country	yes	string100	4.18.3	Country name of the company's location
countryCode	yes	lndISO2	4.18.4	Companie's state (alpha-2 code **[ISO 3166-1-alpha-2]**)
changedVersion	no	versionNumber	4.18.2	Represents the version number of the last change to this element.

4.4.3 Definition

```
<xs:element name="company" type="company-type"/>
<xs:complexType name="company-type">
      <xs:attribute name="id" type="xs:ID" use="required"/>
      <xs:attribute name="name" type="string100" use="required"/>
      <xs:attribute name="code" type="companyCode" use="required"/>
      <xs:attribute name="country" type="string100" use="required"/>
      <xs:attribute name="countryCode" type="lndISO2" use="required"/>
      <xs:attribute name="changedVersion" type="versionNumber"
      use="optional"/>
</xs:complexType>
```

4.4.4 Example

```
<company id="IDC0001"
 name="Company-Name"
 code="123456"
 country="Germany"
 countryCode="DE"/>
```

4.5 The product Element

This element describes each product of the submission.

4.5.1 Child Elements

There are no child elements for this element.

4.5.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier unchanged over the whole lifecycle of the dossier. If a new version introduces changes in product attributes it must have a new id.
name	yes	string250	4.18.3	Product name
formulation	yes	string10	4.18.3	Formulation type according to GIFAP rules. The current formulation codes are managed by CropLife International and are all of length 2. To support legacy identifiers the validation rulesets accepts strings up to length 3.

| annex | yes | xs:boolean | **[XML Schema]** | True, if covered by European annex III dossier |
| changedVersion | no | versionNumber | 4.18.2 | Represents the version number of the last change to this element. |

4.5.3 Definition

```
<xs:element name="product" type="product-type"/>
<xs:complexType name="product-type">
      <xs:attribute name="id" type="xs:ID" use="required"/>
      <xs:attribute name="name" type="string250" use="required"/>
      <xs:attribute name="formulation" type="string10" use="required"/>
      <xs:attribute name="annex" type="xs:boolean" use="required"/>
      <xs:attribute name="changedVersion" type="versionNumber"
      use="optional"/>
</xs:complexType>
```

4.5.4 Example

```
<product id="IDP0001"
 name="Product-Name"
 formulation="FOR"
 annex="true"/>
```

4.6 The active-substance Element

This element describes each active substance of the submission.

4.6.1 Child Elements

There are no child elements for this element.

4.6.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier unchanged over the whole lifecycle of the dossier. If a new version introduces changes in active substance attributes it must have a new id.
name	yes	string250	4.18.3	Substance name
cipac	no	string10	4.18.3	Cipac number
cas	no	string100	4.18.3	CAS code
annex	yes	xs:boolean	**[XML Schema]**	True, if covered by European annex II dossier
changedVersion	no	versionNumber	4.18.2	Represents the version number of the last change to this element.

4.6.3 Definition

```
<xs:element name="active-substance" type="active-substance-type"/>
```

```
<xs:complexType name="active-substance-type">
     <xs:attribute name="id" type="xs:ID" use="required"/>
     <xs:attribute name="name" type="string250" use="required"/>
     <xs:attribute name="cipac" type="string10" use="optional"/>
     <xs:attribute name="cas" type="string100" use="optional"/>
     <xs:attribute name="annex" type="xs:boolean" use="required"/>
     <xs:attribute name="changedVersion" type="versionNumber"
     use="optional"/>
</xs:complexType>
```

4.6.4 Example

```
<active-substance id="IDS0001"
 name="Substance"
 cipac="CIPA"
 cas="CAS-Code"
 annex="true"/>
```

4.7 The concentration Element

This element describes the concentrations of active substances within a product.

4.7.1 Child Elements

There are no child elements for this element.

4.7.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier over the whole lifecycle of the submission. If a new submission introduces changes in concentration attributes it must have a new id.
concentration	yes	string100	4.18.3	Total concentration containing units (e.g. 15 mg/l)
productId	yes	xs.IDREF	**[XML Schema]**	Reference to a product id
substanceId	yes	xs:IDREF	**[XML Schema]**	Reference to a substance id
changedVersion	no	versionNumber	4.18.2	Represents the version number of the last change to this element.

4.7.3 Definition

```
<xs:element name="concentration" type="concentration-type"/>
<xs:complexType name="concentration-type">
     <xs:attribute name="id" type="xs:ID" use="required"/>
     <xs:attribute name="concentration" type="string100" use="required"/>
     <xs:attribute name="productId" type="xs:IDREF" use="required"/>
     <xs:attribute name="substanceId" type="xs:IDREF" use="required"/>
     <xs:attribute name="changedVersion" type="versionNumber"
     use="optional"/>
```

```
</xs:complexType>
```

4.7.4 Example

```
<concentration id="IDR001"
 concentration="15mg/l"
 productId="IDP0001"
 substanceId="IDS0001"/>
```

4.8 The toc Element

A toc element groups all table of content relevant elements. It must have a least one toc-entry element as child. An optional reference to a Standard TOC allowes validators to check the structure of the toc children (See 3.2 for details).

4.8.1 Child Elements

Elementname	Mandatory	Min	Max	Definition	Description
toc-entry	yes	1	*	4.9	Must include at least one toc-entry element

4.8.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
standardTocID	no	xs:ID	**[XML Schema]**	A reference to a Standard TOC from ECPA to allow validators to check the nested structure.
xlink:href	no	xlink:href	**[XLink]**	References the Standard TOC within the 'toc' subdirectory.

4.8.3 Definition

```
<xs:element name="toc" type="toc-type"/>
<xs:complexType name="toc-type">
     <xs:sequence>
          <xs:element ref="toc-entry" maxOccurs="unbounded"/>
     </xs:sequence>
     <xs:attribute name=" standardTocID" type="xs:ID" use="optional"/>
     <xs:attribute ref="xlink:href" use="required"/>

</xs:complexType>
```

4.8.4 Example

```
<toc standardTocID="Sample-OECD-CADDY-xml-TOC"
 xlink:href="utils/tocs/sample-oecd-toc.xml">
 <toc-entry .../>
</toc>
```

4.9 The toc-entry Element

Each entry within a table of content is represented by a toc-entry element. A table of content entry may be a section or a document. In case of a section, it may contain additional table of content elements. In case of a document, a table of content entry must contain a document-ref element.

4.9.1 Child Elements

Element Name	Mandatory	Min	Max	Definition	Description
document-ref	no	0	1	4.10	If this element describes a document this element must be used to reference the document.
toc-entry	no	0	*	4.9	If this element represents a section it may have these elements representing subsections.
hyperlink	no	0	*	4.11	Hyperlinks that are associated with this element.

4.9.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier over the whole lifecycle of the submission. If a new submission introduces changes in header attributes it must have a new id.
number	yes	string100	4.18.3	Hierarchical number of the section or document
title	yes	string250	4.18.3	title of the document or section
intentionallyLeftBlank	no	xs:boolean	**[XML Schema]**	If this is set to 'true' the document-ref is explicitly not defined for this node. Default is 'false'
intentionallyLeftBlankComment	no	string250	**[XML Schema]**	A comment available for the reviewers in case of the flag 'intentionallyLeftBlank' is 'true'. Comments in case of intentionallyLeftBlank is 'false' are not allowed.

4.9.3 Definition

```
<xs:element name="toc-entry" type="toc-entry-type"/>
<xs:complexType name="toc-entry-type">
    <xs:sequence>
        <xs:choice>
            <xs:element ref="document-ref" minOccurs="0"
            maxOccurs="1"/>
            <xs:element ref="toc-entry" minOccurs="0"
            maxOccurs="unbounded"/>
```

```
        </xs:choice>
        <xs:element ref="hyperlink" minOccurs="0"
        maxOccurs="unbounded"/>
     </xs:sequence>
     <xs:attribute name="id" type="xs:ID" use="required"/>
     <xs:attribute name="number" type="string100" use="required"/>
     <xs:attribute name="title" type="string100" use="required"/>
     <xs:attribute name="intentionallyLeftBlank" type="xs:boolean"
     use="optional"/>
     <xs:attribute name="intentionallyLeftBlankComment" type="string250"
     use="optional"/>
</xs:complexType>
```

4.9.4 Example

The following table of content:

1 Section
1.1 Document
2 Section
2.1 Section
2.1.1 Document
2.2 Document

is represented by:

```
<toc>
 <toc-entry id="ID00001" number="1" title="Section">
  <toc-entry id="ID00002" number="1.1" title="Document">
   <document-ref docId="IDD00001"/>
   <hyperlink id="IDH0001" sourceType="toc-entry" targetId="ID00006"/>
  </toc-entry>
 </toc-entry>
 <toc-entry id="ID00003" number="2" title="Section">
  <toc-entry id="ID00004" number="2.1" title="Section">
   <toc-entry id="ID00005" number="2.1.1" title="Document">
    <document-ref docId="IDD00001"/>
   </toc-entry>
  </toc-entry>
  <toc-entry id="ID00006" number="2.2" title="Document">
   <document-ref docId="IDD00001"/>
  </toc-entry>
 </toc-entry>
</toc>
```

4.10 The document-ref Element

This element describes a reference from a table of content entry to a document (4.13) that is placed within the document list (4.12).

4.10.1　　　　Child Elements

There are no child elements for this element.

4.10.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
docId	yes	xs:IDREF	**[XML Schema]**	A unique identifier over the whole lifecycle of the submission that refers to a document Id within the document list.
targetPage	no	xs:Integer	**[XML Schema]**	This attribute allows to open the document on the specified page.

4.10.3 Definition

```
<xs:element name="document-ref" type="document-ref-type"/>
<xs:complexType name="document-ref-type">
      <xs:attribute name="docId" type="xs:IDREF" use="required"/>
      <xs:attribute name="targetPage" type="xs:integer" use="optional"/>
</xs:complexType>
```

4.10.4 Example

In the following example the document-ref element refers to the document named "Document Title 1" through the Id (IDD00001).

```
<toc>
 ...
 <toc-entry id="ID00001" number="1" title="Document">
  <document-ref docId="IDD00001"/>
 </toc-entry>
 ...
</toc>
<document-list>
 <document id="IDD00001"
  title="Document Title 1" .../>
 ...
</document-list>
```

4.11 The hyperlink Element

This element describes a hyperlink. Hyperlinks are defined through a source and target description. They may either point to and from table of content entries, documents, or attachments. This is represented by the source and target type, respectively.

For the coordinate system the following definition is made: The positive x axis extends horizontally to the right and the positive y axis vertically upward, as in standard mathematical practice. This is in accordance to the former CADDY 1.1 and CADDY 2.0 standard.

4.11.1 Child Elements

There are no child elements for this element.

4.11.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier unchangedover the whole lifecycle of the dossier. If a new version introduces changes in hyperlink attributes it must have a new id.
title	yes	string100	**[XLink]**	Description of the hyperlink. E.g. a section name.
targetType	yes	hyperlinkType	4.18.6	Describes the destination type of a hyperlink.
sourceType	yes	hyperlinkType	4.18.6	Describes the source type of a hyperlink.
targetId	yes	xs:IDREF	**[XML Schema]**	References the table of content id of the target or the attachment. It has to be valid for any targetType, not only for the type "table of content". For the types document and toc-entry it references the table of content entry containing the target. For attachments it references the attachment id.
sourcePage	no	xs:integer	**[XML Schema]**	Source page. Only valid if sourceType is document.
sourceX	no	xs:decimal	**[XML Schema]**	x-coordinate of source on source page [mm]. Only valid if sourceType is document.
sourceY	no	xs:decimal	**[XML Schema]**	y-coordinate of source on source page [mm]. Only valid if sourceType is document.
sourceW	no	xs:decimal	**[XML Schema]**	width of source on source page [mm]. Only valid if sourceType is document.
sourceH	no	xs:decimal	**[XML Schema]**	height of source on source page [mm]. Only valid if sourceType is document.
targetPage	no	xs:integer	**[XML Schema]**	Target page. Only valid if targetType is document. Definition will be overridden if targetDestination is given.
targetX	no	xs:decimal	**[XML Schema]**	x-coordinate of target on target page [mm]. Only valid if targetType is document. Definition will be overridden if targetDestination is given.
targetY	no	xs:decimal	**[XML Schema]**	x-coordinate of target on target page [mm]. Only valid if targetType is document. Definition will be overridden if targetDestination is given.
targetDestination	no	String250	4.18.3	named destination. Only valid if targetType is document. Definition overrides targetPage, targetX and targetY.

4.11.3 Definition

```
<xs:element name="hyperlink" type="hyperlink-type"/>
<xs:complexType name="hyperlink-type">
     <xs:attribute name="id" type="xs:ID" use="required"/>
```

```
        <xs:attribute name="title" type="string100" use="required"/>
        <xs:attribute name="sourceType" type="hyperlinkType" use="required"/>
        <xs:attribute name="targetType" type="hyperlinkType" use="required"/>
        <xs:attribute name="targetId" type="xs:IDREF" use="required"/>
        <xs:attribute name="sourcePage" type="xs:integer" use="optional"/>
        <xs:attribute name="sourceX" type="xs:decimal" use="optional"/>
        <xs:attribute name="sourceY" type="xs:decimal" use="optional"/>
        <xs:attribute name="sourceW" type="xs:decimal" use="optional"/>
        <xs:attribute name="sourceH" type="xs:decimal" use="optional"/>
        <xs:attribute name="targetPage" type="xs:integer" use="optional"/>
        <xs:attribute name="targetX" type="xs:decimal" use="optional"/>
        <xs:attribute name="targetY" type="xs:decimal" use="optional"/>
        <xs:attribute name="targetDestination" type="string250"
        use="optional"/>
</xs:complexType>
```

4.11.4   Example

The following example implements two hyperlinks. The first (IDHL0001) points from this table of content entry to the table of content entry with the id ID000001. The second hyperlink is placed on page 2 in the lower left corner (10mm,10mm,10mmx10mm) and points to the named destination 'Summary' of a document with IDD00002.

```
<toc-entry id="ID000005" number="D 1" title="Intended uses …">
 <document-ref docId="IDD00001"/>
 <hyperlink id="IDHL0001"
  title="hyperlink1"
  sourceType="toc-entry"
  targetType="toc-entry"
  targetId="ID000001"/>
 <hyperlink id="IDHL0002"
  title="hyperlink2"
  sourceType="document"
  targetType="document"
  targetId="IDD00002"
  sourcePage="2"
  sourceX="10.0"
  sourceY="10.0"
  sourceW="10.0"
  sourceH="10.0"
  targetDestination="Summary"/>
</toc-entry>
```

4.12 The document-list Element

The document-list element groups all documents within this submission. It must have at least one document element as child.

4.12.1   Child Elements

Element name	Mandatory	Min	Max	Definition	Description
document	yes	1	*	4.13	Must include at least one document element

4.12.2 Attributes

There are no attributes for this element.

4.12.3 Definition

```
<xs:element name="document-list" type="document-list-type"/>
<xs:complexType name="document-list-type">
     <xs:sequence>
          <xs:element ref="document" maxOccurs="unbounded"/>
     </xs:sequence>
</xs:complexType>
```

4.12.4 Example

```
<document-list>
 <document id="IDD00001" title="Document Title 1" .../>
 <document id="IDD00002" title="Document Title 2" .../>
 <document id="IDD00003" title="Document Title 3" .../>
</document-list>
```

4.13 The document Element

This element represents a document. A document must be referenced by at least one table of content entry. A document itself may contain a report data attribute set and/or attachments.

4.13.1 Child Elements

Element Name	Mandatory	Min	Max	Definition	Description
report-data	no	0	1	4.14	May contain a more descriptive set of attributes.
Attachment	no	0	*	4.15	May contain attachment elements.

4.13.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	Yes	xs:ID	**[XML Schema]**	A unique identifier unchanged over the whole lifecycle of the dossier. One id identifies one document over its whole lifecycle.
title	Yes	string250	**[XML Schema]**	Title of the document / report
xlink:href	Yes	PathString	Xlink.xsd	Reference to the file location of the document.
confidential	Yes	xs:boolean	**[XML Schema]**	If true it marks the document and its attachments as confidential.
operation	Yes	changeOperation	4.18.7	The operation valid for this version of the document. See also 3.6

addedVersion	Yes	versionNumber	4.18.2	This version number represents the number where this document was initially added to the submission.
changedVersion	No	versionNumber	4.18.2	Represents the version number of the last change to this element.
checksum	No	md5	4.18.8	Hold a md5 checksum that uniquely identifies the submitted core PDF file for this document. See 3.5. For complete submission the value is mandatory for all referenced content files. For incremental submission, the checksum has to be provided for referenced content files that have changed and part of the current submission, not for previously submitted content that is also referenced from the XML backbone.

4.13.3 Definition

```
<xs:element name="document" type="document-type"/>
<xs:complexType name="document-type">
    <xs:sequence>
        <xs:element ref="report-data" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="attachment" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attribute name="title" type="string250" use="required"/>
    <xs:attribute ref="xlink:href" use="required"/>
    <xs:attribute name="confidential" type="xs:boolean" use="required"/>
    <xs:attribute name="operation" type="changeOperation"
    use="required"/>
    <xs:attribute name="addedVersion" type="versionNumber"
    use="required"/>
    <xs:attribute name="changedVersion" type="versionNumber"
    use="optional"/>
    <xs:attribute name="checksum" type="md5" use="optional"/>
</xs:complexType>
```

4.13.4 Example

A document newly added to the submission version 1.0. It is not confidential and is stored in the file system in file `./standard/documents/idd00001.pdf`:

```
<document id="IDD00001"
 title="Document Title 1"
 confidential="false"
 operation="new"
 xlink:href="../01.00/standard/documents/idd00001.pdf"
 addedVersion="01.00"
 checksum="2d9238e6dffd389a80b208d68e4a1ef0">
<report-data .../>
<attachment .../>
</document>
```

A replaced document changed in version 2.0. It is not confidential and is stored in the file system in file

`'./standard/documents/idd00001.pdf'`:

```
<document id="IDD00001"
 title="Document Title 1"
 confidential="false"
 operation="replaced"
 xlink:href="../02.00/standard/documents/idd00001.pdf"
 addedVersion="01.00"
 changedVersion="02.00"
 checksum="2d9238e6dffd389a80b208d68e4a1ef1">
<report-data .../>
<attachment .../>
</document>
```

4.14 The report-data Element

The report data element stores a set of attributes describing a document.

4.14.1 Child Elements

There are no child elements for this element.

4.14.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier unchanged over the whole lifecycle of the dossier. If a new version introduces changes in the report data attributes it must have a new id.
dossierFilenumber	no	string100	4.18.3	Dossier file number (for EU) / MRID (for US and Canada) Note: This field does not define a report's position within a dossier. The position must be retrieved from the ToC and not from any of the fields in report data.
companyFilenumber	no	string100	4.18.3	Company file number

date	yes	xs:date	**[XML Schema]**	date of document. If validMonth and validDay is true, then date contains the effective date of the report. If validDay = false and validMonth = true, then the day of the month is unknown and date should only be displayed by the software without a day. For sorting and consistency reasons the date field will contain the first day of the month. If validDay and validMonth = false, then the month of the year is unknown and date should only be displayed by the software as a year. For sorting and consistency reasons the date field will contain the first January of the year. The combination validDay = true and validMonth = false is ignored.
validMonth	yes	xs:boolean	**[XML Schema]**	Is day of report date applicable? for explanation see field date
validDay	yes	xs:boolean	**[XML Schema]**	Is month of report date applicable? for explanation see field date
authors	no	string250	4.18.3	Authors of document
source	no	string250	4.18.3	Source of document
owners	no	string250	4.18.3	Owners of report at submission date
testFacility	no	string250	4.18.3	Test facility
glp	yes	xs:boolean	**[XML Schema]**	Is report GLP or GEP conform ?
published	yes	xs:boolean	**[XML Schema]**	Was report published ?
vertebrates	yes	xs:boolean	**[XML Schema]**	Test on vertebrates ?
protect	yes	xs:boolean	**[XML Schema]**	Data protection claimed ?
changedVersion	no	versionNumber	4.18.2	Represents the version number of the last change to this element.

4.14.3 Definition

```
<xs:element name="report-data" type="report-data-type"/>
<xs:complexType name="report-data-type">
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attribute name="dossierFilenumber" type="string100"
    use="optional"/>
    <xs:attribute name="companyFilenumber" type="string100"
    use="optional"/>
    <xs:attribute name="date" type="xs:date" use="required"/>
    <xs:attribute name="validMonth" type="xs:boolean" use="required"/>
    <xs:attribute name="validDay" type="xs:boolean" use="required"/>
    <xs:attribute name="authors" type=" string250" use="optional"/>
    <xs:attribute name="source" type=" string250" use="optional"/>
    <xs:attribute name="owners" type=" string250" use="optional"/>
    <xs:attribute name="testFacility" type=" string250" use="optional"/>
```

```
        <xs:attribute name="glp" type="xs:boolean" use="required"/>
        <xs:attribute name="published" type="xs:boolean" use="required"/>
        <xs:attribute name="vertebrates" type="xs:boolean" use="required"/>
        <xs:attribute name="protect" type="xs:boolean" use="required"/>
        <xs:attribute name="changedVersion" type="versionNumber"
     use="optional"/>
</xs:complexType>
```

4.14.4   Example

```
<report-data id="IDR00001"
 dossierFilenumber="DOC/00001"
 companyFilenumber="I/00001"
 date="2001-02-01"
 validMonth="true"
 validDay="true"
 authors="Smith et. al."
 source="Source"
 owners="Owner"
 testFacility="Testfacility"
 glp="true"
 published="true"
 vertebrates="true"
 protect="true"/>
```

4.15 The attachment Element

The attachment element stores a reference to an attachment file in the file system. The attachmentType describes the function in respect of the document of this file.

4.15.1   Child Elements

There are no child elements for this element.

4.15.2   Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier over the whole lifecycle of the submission. If a new submission introduces changes in the attachment attributes it must have a new id.
attachmentType	yes	attachmentType	4.18.10	Classifies the attachment file.
title	yes	string250	4.18.3	Describes the attachment file.
xlink:href	yes	xlink:href	**[XLink]**	References the attachment file in the file system.
addedVersion	yes	versionNumber	4.18.2	The version number where this attachment was added.

checksum	No	md5	4.18.8	An unique md5 identifier for the attachment file. For incremental submission, the checksum has to be provided for referenced content files that have changed and part of the current submission, not for previously submitted content that is also referenced from the XML backbone.
changedVersion	No	versionNumber	4.18.2	Represents the version number of the last change to this element.
comment	No	string250	4.18.3	Comment describing the function of this attachment.

4.15.3 Definition

```
<xs:element name="attachment" type="attachment-type"/>
<xs:complexType name="attachment-type">
      <xs:attribute name="id" type="xs:ID" use="required"/>
      <xs:attribute name="attachmentType" type="attachmentType"
      use="required"/>
      <xs:attribute name="title" type="string250" use="required"/>
      <xs:attribute ref="xlink:href" use="required"/>
      <xs:attribute name="addedVersion" type="versionNumber"
      use="required"/>
      <xs:attribute name="checksum" type="md5" use="optional"/>
      <xs:attribute name="changedVersion" type="versionNumber"
      use="optional"/>
      <xs:attribute name="comment" type="string250" use="optional"/>
</xs:complexType>
```

4.15.4 Example

```
<attachment id="IDA00001"
 attachmentType="rendition"
 title="Rendition of the document"
 xlink:href="../01.00/standard/attachments/idd00001/rendition.pdf"
 addedVersion="01.00"
 checksum="2d9238e6dffd389a80b208d68e4a1ef0"/>
```

4.16 The additional-files-list Element

An additional file list element groups all additional file elements. For details refer to section 3.1.8.

4.16.1 Child Elements

Elementname	Mandatory	Min	Max	Definition	Description
additional-files	no	0	*	4.17	May include additional file elements

4.16.2 Attributes

There are no attributes for this element.

4.16.3   Definition

```
<xs:element name="additional-files-list"
type="additional-files-list-type"/>
<xs:complexType name="additional-files-list-type">
      <xs:sequence>
            <xs:element ref="additional-file" minOccurs="0"
            maxOccurs="unbounded"/>
      </xs:sequence>
</xs:complexType>
```

4.16.4   Example

```
<additional-files-list>
 <additional-file id="IDAV0001"
  comment="Standard CADDY Logo"
  xlink:href="../01.00/additional-files/CADDY-Logo.gif"
  addedVersion="01.00"
  checksum="6e93076f8b0b79938eb8ab7019cf2249"/>
</additional-files-list>
```

4.17 The additional-file Element

This element represents an additional file that does not belong to a specific table of content element.

For details refer to section 3.1.8.

4.17.1   Child Elements

This element has no child elements.

4.17.2   Attributes

Fieldname	Mandatory	Type	Definition	Description
id	yes	xs:ID	**[XML Schema]**	A unique identifier over the whole lifecycle of the submission. One id always belongs to one additional file and its whole lifecycle.
comment	yes	string250	4.18.3	Any text describing the additional file.
xlink:href	yes	xlink:href	**[XLink]**	Reference to the file location of the document.
addedVersion	yes	versionNumber	4.18.2	This version number represents the number where this document was initially added to the submission.
changedVersion	no	versionNumber	4.18.2	Represents the version number of the last change to this element.

checksum	no	md5	4.18.8	Hold a md5 checksum that uniquely identifies this document. See chapter 3.5. For incremental submission, the checksum has to be provided for referenced content files that have changed and part of the current submission, not for previously submitted content that is also referenced from the XML backbone.

4.17.3   Definition

```
<xs:element name="additional-file" type="additional-file-type"/>
<xs:complexType name="additional-file-type">
      <xs:attribute name="id" type="xs:ID" use="required"/>
      <xs:attribute name="comment" type="string250" use="required"/>
      <xs:attribute ref="xlink:href" use="required"/>
      <xs:attribute name="addedVersion" type="versionNumber"
      use="required"/>
      <xs:attribute name="changedVersion" type="versionNumber"
      use="optional"/>
      <xs:attribute name="checksum" type="md5" use="optional"/>
</xs:complexType>
```

4.17.4   Example

```
<additional-files-list>
 <additional-file id="IDAV0001"
  comment="Standard CADDY Logo"
  xlink:href="../01.00/additional-files/CADDY-Logo.gif"
  addedVersion="01.00"
  checksum="6e93076f8b0b79938eb8ab7019cf2249"/>
</additional-files-list>
```

4.18 CADDY-*xml* Types

CADDY-*xml* introduces a list of CADDY-*xml* specific types.

4.18.1   The xmlVersionNumber Type

This type represents a CADDY-*xml* version number of a valid and effective specification version. It consists of two digit numbers for version (major version), subversion (minor version) and patch level.. These version components are separated by single dots. E.g. "03.00.00".

4.18.1.1 Definition

```
<xs:simpleType name='xmlVersionNumber'>
      <xs:annotation>Version number: six digits</xs:annotation>
      <xs:restriction base='xs:string'>
            <xs:pattern value='\d{2}.\d{2}.\d{2}'/>
```

```
        </xs:restriction>
</xs:simpleType>
```

4.18.2   The versionNumber Type

This type represents a CADDY-*xml* submission version number that consists of a two digit number for version (major version) and subversion (minor version). These versions are separated by a dot. E.g. "01.00" for version 1.0, or "02.01" for version 2.1. Major versions start with 01 and minor versions start with 00. Both must be consecutive.

4.18.2.1 Definition

```
<xs:simpleType name='versionNumber'>
      <xs:annotation>Version number: four digits</xs:annotation>
      <xs:restriction base='xs:string'>
           <xs:pattern value='\d{2}.\d{2}'/>
      </xs:restriction>
</xs:simpleType>
```

4.18.3   The Restricted String Types

There are several string types defined within CADDY-*xml*. These are aimed to restrict the length of a string. E.g. string250 restrict a string to the length of 250 characters.

CADDY-*xml* defines the following string types: string10, string100, and string250.

4.18.3.1 Definition

Real types use the actual length of the string instead of XX:

```
<xs:simpleType name='stringXX'>
      <xs:restriction base='xs:string'>
           <xs:minLength value='1'/>
           <xs:maxLength value='XX'/>
      </xs:restriction>
</xs:simpleType>
```

4.18.4   The IndISO2 Type

This type is used for the **[ISO 3166-1-alpha-2]** two alpha country code.

4.18.4.1 Definition

```
<xs:simpleType name='lndISO2'>
      <xs:restriction base='xs:string'>
           <xs:pattern value='[A-Z][A-Z]'/>
           <xs:maxLength value='2'/>
      </xs:restriction>
</xs:simpleType>
```

4.18.5 The companyCode Type

This type represents the company code. It must either be the 3-digit EU company code or a 6-letter US company id.

4.18.5.1 Definition

```
<xs:simpleType name='companyCode'>
      <xs:restriction base='xs:string'>
            <xs:minLength value='3'/>
            <xs:maxLength value='6'/>
      </xs:restriction>
</xs:simpleType>
```

4.18.6 The hyperlinkType Type

This type is used to represent the source or target type of a hyperlink. It is either a toc-entry, a document or an attachment.

4.18.6.1 Definition

```
<xs:simpleType name='hyperlinkType'>
      <xs:restriction base='xs:string'>
            <xs:enumeration value="toc-entry"/>
            <xs:enumeration value="document"/>
            <xs:enumeration value="attachment"/>
      </xs:restriction>
</xs:simpleType>
```

4.18.7 The changeOperation Type

This type represents the operations that can be done for documents.

Operation	Description
new	If a document is newly added to a submission
deleted	If a document has been deleted
replaced	If a document has been replaced by another document

4.18.7.1 Definition

```
<xs:simpleType name='changeOperation'>
      <xs:restriction base='xs:string'>
            <xs:enumeration value="new"/>
            <xs:enumeration value="deleted"/>
            <xs:enumeration value="replaced"/>
      </xs:restriction>
</xs:simpleType>
```

4.18.8 The md5 Type

This type represents the md5 checksum specified in **[RFC 1321]**. A md5 checksum is a string that contains 32 hexadecimal characters.

4.18.8.1 Definition

```
<xs:simpleType name='md5'>
     <xs:annotation>MD5 Checksum</xs:annotation>
     <xs:restriction base='xs:string'>
          <xs:pattern value='[a-f,A-F,0-9]{32}'/>
     </xs:restriction>
</xs:simpleType>
```

4.18.9 The dossierID Type

To identify a dossier, every dossier will get an unique dossier ID, which is built using the following three subparts:

[dossierID] ::= [notifying company shortcode] [country shortcode] [current dossier number]

[notifying company shortcode] ::=   Notifying company or in case of a task force the leading company
                                     (3-digit EU company code or 6-letter US company ID, only capital
                                     letters are allowed.)

[country shortcode] ::=   Country code of the notifying company (short form with 2 characters)
                          (ISO 3166 - Alpha-2-code, only capital letters are allowed.)

[current dossier number] ::=   Current number of dossier generated by the notifying company
                               (padded with leading zeros to 3 to 5 characters)

The complete dossier ID string is between 8 and 13 characters long.

This ensures, that the notifying company is able to guarantee the uniqueness of the dossierID (formally known as "Volume Set IDs") of its submissions.Example: DOEGB00001

4.18.9.1 Definition

```
<xs:simpleType name='dossierID'>
     <xs:restriction base='xs:string'>
          <xs:pattern value='[A-Z,0-9]{8,13}'/>
     </xs:restriction>
</xs:simpleType>
```

4.18.10 The attachmentType Type

This type classifies an attachment. See an overview of the various attachment categories in chapter 3.1.7.

4.18.10.1   Definition

```
<xs:simpleType name='attachmentType'>
 <xs:restriction base='xs:string'>
            <xs:enumeration value="rendition"/>
            <xs:enumeration value="appendix"/>
            <xs:enumeration value="figure"/>
            <xs:enumeration value="photo"/>
            <xs:enumeration value="sas-table"/>
            <xs:enumeration value="oecd-data"/>
            <xs:enumeration value="zip-file"/>
            <xs:enumeration value="other"/>
      </xs:restriction>
</xs:simpleType>
```

5 The CADDY-*xml* Standard TOC Definitions

This chapter describes all elements and their associations of a CADDY-*xml* Standard TOC in detail. The following figure gives an overview of the hierarchical structure of the Standard TOC structure. All elements within this section are XML-elements that are placed within one file.
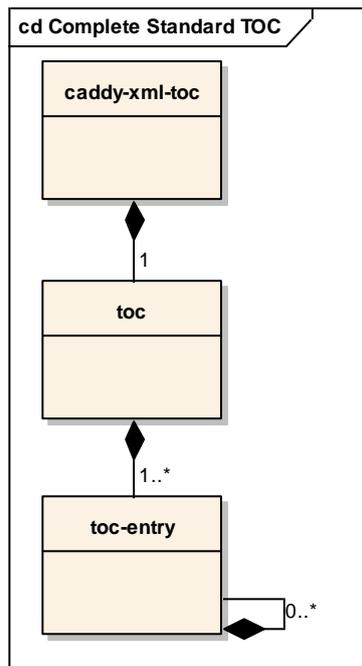

Figure 13: Overview CADDY-xml Standard TOC Elements

Note: All types of the following attributes definitions references the CADDY-*xml* types wherever appropriate. Only attribute types not already defined earlier are introduced in section 5.4.

5.1 The caddy-xml-toc Element

This element is the root element of all CADDY-*xml* Standard TOC files. The caddy-xml-toc element must have exactly one child, the toc element.

5.1.1 Child Elements

Element name	Mandatory	Min	Max	Definition	Description
toc	yes	1	1	5.2	Must include this element that describes the toc in detail.

5.1.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
xmlVersion	yes	xmlVersionNumber	4.18.1	The version number of the used CADDY-*xml* specification. It is composed of a two digit major, a two digit minor and a two digit patch level version number separated by single dots (e.g. 03.08.00).
id	yes	xs:ID	[XML Schema]	A unique identifier unchanged over the whole lifecycle of the TOC and its different versions.
title	yes	string250	4.18.3	A title for the TOC.
tocVersion	yes	versionNumber	4.18.2	The version number of this TOC. It is composed of a two digit major and a two digit minor version number separated with a dot. E.g. 01.00 or 01.02
originator	yes	string100	4.18.3	Organisation which created this TOC.
author	no	string100	4.18.3	Responsible person for this TOC.
issueDate	yes	xs:date	[XML Schema]	Issue data of the TOC
validFrom	yes	xs:date	[XML Schema]	The earliest date from which this TOC can be treated as valid.
validTo	no	xs:date	[XML Schema]	The latest date till this TOC can be treated as valid.
substitutesVersion	no	versionNumber	4.18.2	A previous version which will be substituted by this version. New Versions shall always set this field even if it is defined optional.

5.1.3 Definition

```
<xs:element name="caddy-xml-toc" type="caddy-xml-toc-type"/>
<xs:complexType name="caddy-xml-toc-type">
     <xs:sequence>
          <xs:element ref="toc"/>
     </xs:sequence>
     <xs:attribute name="xmlVersion" type="xmlVersionNumber"
     use="required"/>
     <xs:attribute name="id" type="xs:ID" use="required"/>
     <xs:attribute name="title" type="string250" use="required"/>
     <xs:attribute name="tocVersion" type="versionNumber"
     use="required"/>
     <xs:attribute name="originator" type="string100" use="required"/>
     <xs:attribute name="author" type="string100" use="optional"/>
     <xs:attribute name="issueDate" type="xs:date" use="required"/>
     <xs:attribute name="validFrom" type="xs:date" use="required"/>
     <xs:attribute name="validTo" type="xs:date" use="optional"/>
     <xs:attribute name="subsitutesVersion" type="versionNumber"
     use="optional"/>
</xs:complexType>
```

5.1.4 Example

```
<caddy-xml-toc
 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:noNamespaceSchemaLocation="caddy-toc-3-0.xsd"
 xmlns:xlink=http://www.w3.org/1999/xlink
 xmlVersion="03.00.00"
 id="Sample-OECD-CADDY-xml-TOC"
 title="Sample OECD CADDY.xml TOC"
 tocVersion="01.00"
 originator="ECPA"
 author="Andreas Wastl"
 issueDate="2007-07-06"
 validFrom="2007-07-06">
```

5.2   The toc Element

A toc element groups all Standard TOC relevant elements. It exists with one instance and groups all toc-entry elements that describe the table of contents of the dossier. It must have at least one toc-entry element as child.

5.2.1 Child Elements

Element name	Mandatory	Min	Max	Definition	Description
toc-entry	yes	1	*	5.3	Must include at least one toc-entry element

5.2.2 Attributes

There are no attributes for this element.

5.2.3 Definition

```
<xs:element name="toc" type="toc-type"/>
<xs:complexType name="toc-type">
      <xs:sequence>
            <xs:element ref="toc-entry" maxOccurs="unbounded"/>
      </xs:sequence>
</xs:complexType>
```

5.2.4 Example

```
<toc>
 <toc-entry .../>
</toc>
```

5.3 The toc-entry Element

Each entry within a Standard TOC is represented by a toc-entry element. A table of content entry may contain additional table of content elements to build up a tree structure.

5.3.1 Child Elements

Element Name	Mandatory	Min	Max	Definition	Description
toc-entry	no	0	*	4.10	Further subsections of this toc-entry or a placeholder for a document.

5.3.2 Attributes

Fieldname	Mandatory	Type	Definition	Description
number	yes	string100	4.18.3	Hierarchical number of the section or document
title	yes	string100	4.18.3	Title of the document or section
extensible	yes	xs:boolean	**[XML Schema]**	If this is set to either 'true' the toc may have extra sub levels starting from this entry.
documentRef	yes	documentRefType	**[XML Schema]**	Guideline for the occurrence of a document reference in the submission toc. Either 'allowed', 'required' or 'prohibited'.

5.3.3 Definition

```
<xs:element name="toc-entry" type="toc-entry-type"/>
<xs:complexType name="toc-entry-type">
     <xs:sequence>
          <xs:element ref="toc-entry" minOccurs="0"
          maxOccurs="unbounded"/>
     </xs:sequence>
     <xs:attribute name="number" type="string100" use="required"/>
     <xs:attribute name="title" type="string100" use="required"/>
     <xs:attribute name="extensible" type="xs:boolean" use="required"/>
     <xs:attribute name="documentRef" type="documentRefType"
     use="required"/>
</xs:complexType>
```

5.3.4 Example

The following Standard TOC :

1 Section
2 Section   ***(no documents allowed, not extensible)***
2.1 Section
2.2 Section
2.2.1 Document   **(document required, not extensible)**

is represented by:

```
<toc>
 <toc-entry number="1" title="Section"
  extensible="true" document-ref="allowed">
 </toc-entry>
 <toc-entry number="2" title="Section"
  extensible="false" document-ref="prohibited">
  <toc-entry number="2.1" title="Section"
   extensible="true" document-ref="allowed">
  </toc-entry>
  <toc-entry number="2.2" title="Section"
   extensible="true" document-ref="allowed">
   <toc-entry number="2.2.1" title="Document"
    extensible="false" document-ref="required">
   </toc-entry>
  </toc-entry>
 </toc-entry>
</toc>
```

5.4 CADDY-*xml* Standard TOC Types

CADDY-*xml* Standard TOCs introduces an extra list of CADDY-*xml* specific types.

5.4.1 The documentRef Type

This type represents the different document reference enforcements within a Standard TOC.

5.4.1.1   Definition

```
<xs:simpleType name='documentRefType'>
      <xs:annotation>Gives an enumeration of the valid document references
      </xs:annotation>
      <xs:restriction base='xs:string'>
            <xs:enumeration value="allowed"/>
            <xs:enumeration value="required"/>
            <xs:enumeration value="prohibited"/>
       </xs:restriction>
</xs:simpleType>
```

6 Appendices

This chapter contains the appendices

6.1 XML Definition

The XML definition in the form of the CADDY-xml XSD is not part of this document and only available online.

Please refer to the ECPA CADDY-*xml* internet page: http://esubmission.ecpa.eu

6.2 CADDY-xml viewer module

The CADDY-xml viewer module is a component that is used in conjunction with the ECPA eSubmission viewer. Both the eSubmission viewer and the CADDY-xml viewer are downloaded and installed once on the Windows PC of the reviewer and are used for viewing all CADDY-xml dossiers. No additional installation is required per dossier.

The description of the CADDY-xml viewer module is not part of this document and only available online. The CADDY-xml viewer module is a separate optional CADDY artefact and can be versioned separately from this specification document. This means that the list of files for the viewer module is independent on this specification.

The submission package does not include a viewer. Prior to version 03.08.00 of this specification, the outdated embedded viewer has been part of the submission folder, in the subdirectory utils/viewer. This viewer was replaced with the current viewer module due to issues when displaying the dossier within the latest web browser versions.

As a consequence the following files present in submission of version 03.07.00 and earlier are no longer required and must not be present starting with specification version 03.08.00:

- in the root directory
    - o index.html
- in the utils subfolder:
    - o caddy-3-0.xsl and all other *.xsl
    - o All *.js files
    - o All subfolders below utils except optionally for "tocs"
    - o The utils subfolders will contain two files only: The current CADDY XSD plus (optionally) the xlink.xsd file.

6.2.1 XSLT for static HTML representation

In order to support the quick display of a dossier when no eSubmission Viewer and/or a CADDY-xml module is available, the root directory of the submission has to contain a file called "caddy.html" that contains a human-readable overview of the information present in the CADDY XML backbone file. It is displayed in the default web browser. The file is to be generated using the XSLT file from the CADDY-xml backbone XML file during the generation of the CADDY-xml submission package.

The XSLT is supplied by ECPA and can be downloaded on the eSubmission website
http://esubmission.ecpa.eu

6.3 Stylesheet reference

The CADDY-xml backbone file must contain a reference to the stylesheet file with fixed name "caddy-3-0.xsl":

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<?xml-stylesheet type="text/xsl" href="utils/caddy-3-0.xsl"?>
<caddy-xml xsi:noNamespaceSchemaLocation="utils/caddy_03-07-00.xsd"
xmlVersion="03.07.00" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink">
```

This stylesheet reference was not explicitly mentioned in previous specifications but was already needed for the outdated embedded viewer. It is mandatory in exact the syntax as listed above also in the specifications with versions 03.08.00 and later: It is required for the functioning of the new CADDY-xml viewer module in the eSubmission viewer, even if the reference in the submission package points to a non-existent XSL file.

6.4 Versioning Example

6.4.1 Introduction

This section illustrates versioning within CADDY-*xml* with a small sample dossier consisting of three submissions 01.00, 01.01 and 02.00. Each submission has different document operations which cover the different actions during the dossier lifecycle.

6.4.2 Table of Content

The following outline of a table of content shows the locations where documents are located:

1 Section   *(initially added in submission 01.00)*
 1.1 Document A*
 1.2 Document B*
 1.3 Document C*
 1.4 Document D*
2 Section   *(initially added in submission 01.01)*
 2.1 Document KK*

Due to incremental build-up and delete operations the above structure will not really exist in any of the submissions completely. It is only an outline for all documents which are part of the Table of Content over all submissions. The different submissions will only have subsets of this structure.

6.4.3 Submission 01.00

The initial submission is always a major submission and can only have documents with the operation "new".

6.4.3.1 Operations

The following documents will be newly added:

- Document A

- Document B (with report-data)

- Document C

- Document D

The following hyperlinks exist:

- Document-Link from Document A (page 1) to Document B (page 1)

- Document-Link from Document B (page 1) to Document C (page 1)

- Document-Link from Document C (page 1) to Document D (page 1)

6.4.3.2 XML Snippet

The following XML fragment shows the relevant parts concerning versioning and does not contain all mandatory attributes and elements:

```
<caddy-xml ...>
 ...
 <toc>
  <!-- Submission 01.00 TOC -->
  <toc-entry id="IDT001" number="1" title="Section">
   <toc-entry id="IDT011" number="1.1" title="Document A*">
    <document-ref docId="IDD001"/>
    <hyperlink targetType="document" targetPage="1" title="A to B"
     id="IDHL001" sourceType="document" sourcePage="1" targetId="IDT012"/>
   </toc-entry>
```

```
    <toc-entry id=" IDT012" number="1.2" title="Document B*">
     <document-ref docId="IDD002"/>
     <hyperlink targetType="document" targetPage="1" title="B to C"
      id="IDHL002" sourceType="document" sourcePage="1" targetId="IDT013"/>
    </toc-entry>
    <toc-entry id=" IDT013" number="1.3" title="Document C*">
     <document-ref docId="IDD003"/>
     <hyperlink targetType="document" targetPage="1" title="C to D"
      id="IDHL003" sourceType="document" sourcePage="1" targetId="IDT014"/>
    </toc-entry>
    <toc-entry id=" IDT014" number="1.4" title="Document D*">
     <document-ref docId="IDD004"/>
    </toc-entry>
   </toc-entry>
  </toc>
  ...
  </toc>
  <document-list>
   <document id="IDD001" title="Document A" operation="new"
    xlink:href="../01.00/standard/documents/idd001/idd001.pdf"
    addedVersion="01.00" .../>
   <document id="IDD002" title="Document B" operation="new"
    xlink:href="../01.00/standard/documents/idd002/idd002.pdf"
    addedVersion="01.00" ...>
     <report-data id="IDR001" validDay="true" protect="true"
       date="2005-03-15" published="true" vertebrates="true"
       validMonth="true" glp="true" title="Report Data Title Original"/>
   </document>
   <document id="IDD003" title="Document C" operation="new"
    xlink:href="../01.00/standard/documents/idd003/idd003.pdf"
    addedVersion="01.00" .../>
   <document id="IDD004" title="Document D" operation="new"
    xlink:href="../01.00/standard/documents/idd004/idd004.pdf"
    addedVersion="01.00" .../>
  </document-list>
  ...
</caddy-xml>
```

6.4.4 Submission 01.01

The first follow-up submission contains some variation in the files and hyperlinks:

6.4.4.1 Operations

The following documents will be newly added:

- Document KK

- Document LL

- Document MM

The following documents will be replaced:

- Document A will become Document AA

- Document C will become Document CC

- The report-data of Document B

The following documents will be deleted:

- Document D

The following hyperlinks will be added to the already existing ones

- Document KK to Document LL

- Document LL to Document MM

Note: Hyperlinks which point to not longer existing documents will be removed from the table of contents.

6.4.4.2 XML Snippet

The following XML fragment shows the interesting parts concerning versioning and does not contain all mandatory attributes and elements:

```
<caddy-xml ...>
 ...
 <toc>
  <!—- Submission 01.00 TOC -->
  <toc-entry id="IDT001" number="1" title="Section">
   <toc-entry id="IDT011" number="1.1" title="Document A*">
    <document-ref docId="IDD001"/>
    <hyperlink targetType="document" targetPage="1" title="A to B"
     id="IDHL001" sourceType="document" sourcePage="1" targetId="IDT012"/>
   </toc-entry>
   <toc-entry id=" IDT012" number="1.2" title="Document B*">
    <document-ref docId="IDD002"/>
    <hyperlink targetType="document" targetPage="1" title="B to C"
     id="IDHL002" sourceType="document" sourcePage="1" targetId="IDT013"/>
   </toc-entry>
   <toc-entry id=" IDT013" number="1.3" title="Document C*">
    <document-ref docId="IDD003"/>
   </toc-entry>
  </toc-entry>
  <!—- Submission 01.01 TOC -->
  <toc-entry id="IDT002" number="2" title="Section">
   <toc-entry id="IDT021" number="2.1" title="Document KK*">
    <document-ref docId="IDD005"/>
    <hyperlink targetType="document" targetPage="1" title="KK to LL"
     id="IDHL004" sourceType="document" sourcePage="1" targetId="IDT022"/>
   </toc-entry>
   <toc-entry id=" IDT022" number="2.2" title="Document LL*">
    <document-ref docId="IDD006"/>
    <hyperlink targetType="document" targetPage="1" title="LL to MM"
     id="IDHL005" sourceType="document" sourcePage="1" targetId="IDT023"/>
   </toc-entry>
   <toc-entry id=" IDT023" number="2.3" title="Document MM*">
    <document-ref docId="IDD007"/>
```

```
        </toc-entry>
      </toc-entry>
    </toc>
    ...
    </toc>
    <document-list>
    <!—- Submission 01.00 Files -->
    <document id="IDD001" title="Document AA" operation="replaced"
     xlink:href="../01.01/standard/documents/idd001/idd001.pdf"
     addedVersion="01.00" changedVersion="01.01" .../>
    <document id="IDD002" title="Document B" operation="new"
     xlink:href="../01.00/standard/documents/idd002/idd002.pdf"
     addedVersion="01.00" ...>
      <report-data id="IDR001" validDay="true" protect="true"
        date="2005-03-15" published="true" vertebrates="true"
        validMonth="true" glp="true" title="Report Data Title Changed"/>
    </document>
    <document id="IDD003" title="Document CC" operation="replaced"
     xlink:href="../01.01/standard/documents/idd003/idd003.pdf"
     addedVersion="01.00" changedVersion="01.01" .../>
    <document id="IDD004" title="Document D" operation="deleted"
     xlink:href="../01.00/standard/documents/idd004/idd004.pdf"
     addedVersion="01.00" changedVersion="01.01" .../>
    <!—- Submission 01.01 Files -->
    <document id="IDD005" title="Document KK" operation="new"
     xlink:href="../01.01/standard/documents/idd005/idd005.pdf"
     addedVersion="01.01" .../>
    <document id="IDD006" title="Document LL" operation="new"
     xlink:href="../01.01/standard/documents/idd006/idd006.pdf"
     addedVersion="01.01" .../>
    <document id="IDD007" title="Document MM" operation="new"
     xlink:href="../01.01/standard/documents/idd007/idd007.pdf"
     addedVersion="01.01" .../>
    </document-list>
    ...
</caddy-xml>
```

6.4.5 Submission 02.00

This submission will be again a complete submission but also introduces some changes:

6.4.5.1 Operations

The following documents will be newly added:

- Document SSS

- Document TTT

- Document UUU

The following documents will be replaced:

- Document AA will become Document AAA

- Document LL will become Document LLL

The following documents will be deleted:

- Document CC

- Document MM

The following hyperlinks will be added to the already existing ones:

- Document SSS to Document TTT

- Document TTT to Document UUU

Note: Hyperlinks which point to not longer existing documents will be removed from the table of contents.

6.4.5.2 XML Snippet

The following XML fragment shows the interesting parts concerning versioning and does not contain all mandatory attributes and elements:

```
<caddy-xml ...>
 ...
 <toc>
  <!-- Submission 01.00 TOC -->
  <toc-entry id="IDT001" number="1" title="Section">
   <toc-entry id="IDT011" number="1.1" title="Document A*">
    <document-ref docId="IDD001"/>
    <hyperlink targetType="document" targetPage="1" title="A to B"
     id="IDHL001" sourceType="document" sourcePage="1" targetId="IDT012"/>
   </toc-entry>
   <toc-entry id=" IDT012" number="1.2" title="Document B*">
    <document-ref docId="IDD002"/>
   </toc-entry>
  </toc-entry>
  <!-- Submission 01.01 TOC -->
  <toc-entry id="IDT002" number="2" title="Section">
   <toc-entry id="IDT021" number="2.1" title="Document KK*">
    <document-ref docId="IDD005"/>
    <hyperlink targetType="document" targetPage="1" title="KK to LL"
     id="IDHL004" sourceType="document" sourcePage="1" targetId="IDT022"/>
   </toc-entry>
   <toc-entry id=" IDT022" number="2.2" title="Document LL*">
    <document-ref docId="IDD006"/>
   </toc-entry>
  </toc-entry>
  <!-- Submission 02.00 TOC -->
  <toc-entry id="IDT003" number="3" title="Section">
   <toc-entry id="IDT031" number="3.1" title="Document SSS">
    <document-ref docId="IDD008"/>
    <hyperlink targetType="document" targetPage="1" title="SSS to TTT"
     id="IDHL004" sourceType="document" sourcePage="1" targetId="IDT032"/>
   </toc-entry>
   <toc-entry id=" IDT032" number="3.2" title="Document TTT">
    <document-ref docId="IDD009"/>
    <hyperlink targetType="document" targetPage="1" title="TTT to UUU"
```

```
          id="IDHL005" sourceType="document" sourcePage="1" targetId="IDT033"/>
      </toc-entry>
      <toc-entry id=" IDT033" number="3.3" title="Document UUU">
       <document-ref docId="IDD010"/>
      </toc-entry>
    </toc-entry>
  </toc>
  ...
  </toc>
  <document-list>
   <!—- Submission 01.00 Files -->
   <document id="IDD001" title="Document AAA" operation="replaced"
    xlink:href="../02.00/standard/documents/idd001/idd001.pdf"
    addedVersion="01.00" changedVersion="02.00" .../>
   <document id="IDD002" title="Document B" operation="new"
    xlink:href="../02.00/standard/documents/idd002/idd002.pdf"
    addedVersion="01.00" ...>
    <report-data id="IDR001" validDay="true" protect="true"
      date="2005-03-15" published="true" vertebrates="true"
      validMonth="true" glp="true" title="Report Data Title Changed"/>
   </document>
   <document id="IDD003" title="Document CC" operation="deleted"
    xlink:href="../01.01/standard/documents/idd003/idd003.pdf"
    addedVersion="01.00" changedVersion="02.00" .../>
   <document id="IDD004" title="Document D" operation="deleted"
    xlink:href="../01.00/standard/documents/idd004/idd004.pdf"
    addedVersion="01.00" changedVersion="01.01" .../>
   <!—- Submission 01.01 Files -->
   <document id="IDD005" title="Document KK" operation="new"
    xlink:href="../02.00/standard/documents/idd005/idd005.pdf"
    addedVersion="01.01" .../>
   <document id="IDD006" title="Document LLL" operation="replaced"
    xlink:href="../02.00/standard/documents/idd006/idd006.pdf"
    addedVersion="01.01" changedVersion="02.00" .../>
   <document id="IDD007" title="Document MM" operation="deleted"
    xlink:href="../02.00/standard/documents/idd007/idd007.pdf"
    addedVersion="01.01" changedVersion="02.00" .../>
   <!—- Submission 02.00 Files -->
   <document id="IDD008" title="Document SSS" operation="new"
    xlink:href="../02.00/standard/documents/idd008/idd008.pdf"
    addedVersion="02.00" .../>
   <document id="IDD009" title="Document TTT" operation="new"
    xlink:href="../02.00/standard/documents/idd009/idd009.pdf"
    addedVersion="02.00" .../>
   <document id="IDD010" title="Document UUU" operation="new"
    xlink:href="../02.00/standard/documents/idd010/idd010.pdf"
    addedVersion="02.00" .../>
  </document-list>
  ...
</caddy-xml>
```

6.5 References

6.5.1 Normative References

[eCTD]

ICH eCTD IWG, *Electronic Common Technical Document Specification V 3.2*, ICH M2 EWG, 2004

(See http://www.ich.org)

[ISO 3166-1-alpha-2]

ISO 1366-1:1997: *Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes -- Alpha-2, ISO, 1997*

[ISO-19005-1]

ISO-19005-1:2004: *Document management – Electronic document file format for long-term preservation - Part 1: Use of PDF 1.4 (PDF/A-1)*, ISO, 2004

[RFC 1321]

RFC 1321: The MD5 Message-Digest Algorithm

(See http://www.ietf.org/rfc/rfc1321.txt)

[RFC 2119]

RFC 2119: Key words for use in RFCs to Indicate Requirement Levels , S. Bradner, 1997.
(See http://www.ietf.org/rfc/rfc2119.txt)

[URL]

RFC 1738, Uniform Resource Locators (URL). Internet Engineering Task Force.
(See http://www.ietf.org/rfc/rfc1738.txt)

[XHTML]

M. Altheim, et al. *Modularization of XHTML*, 2001. W3C

(See http://www.w3.org/TR/xhtml-modularization/).

[XML]

Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, and Eve Maler, editors. *Extensible Markup Language (XML) 1.0 (Second Edition).* World Wide Web Consortium, 2000.

(See http://www.w3.org/TR/2000/REC-xml-20001006.)

[XLink]

Steve DeRose, Eve Maler, David Orchard, and Ben Trafford, editors. *XML Linking Language (XLink)* . World Wide Web Consortium, 2000.

(See http://www.w3.org/TR/xlink.)

[XML Schema]

David C. Fallside, Priscilla Walmsley - Second Edition, editors. *XML Schema Part 0: Primer Second Edition*, World Wide Web Consortium, 2004.

(See http://www.w3.org/TR/xmlschema-0/)

[XSLT]

James Clark , editor. *World Wide Web Consortium. XSL Transformations (XSLT). W3C Recommendation.*, World Wide Web Consortium, 1999

(See http://www.w3.org/TR/xslt)

6.5.2 Informative References

[Adobe-Ref]

How to link from an HTML page to a specific page in PDF file , Adobe, 2005.

(See http://www.adobe.com/support/techdocs/315385.html)

6.6 Version History

The following table gives an overview of the existing versions of this specification.

Version	Date	Author	Status
1.0	29/04/2005	Oliver Bojahr	Initial version
1.0.1	19/07/2005	Oliver Bojahr	- version-history-list must not have entries for the initial submission. MinOccurrence of version-history set to "0". - Complex types have the suffix "-type" to allow import with JAXB
1.1.0	03/07/2006	Oliver Bojahr	- unique dossier ID added to section 4.3 and 4.20.8 - master and issue dates added to version in section 4.2 - string types reduced - set header attributes as optional - dossier id as root folder of dossier - Section 5.1: CADDY-*xml* definition replaced by reference
1.1.0 prop. 3	06/07/2007	Andreas Wastl	- Old section 3.1.4 "Version History" and all references to version-history in texts or images removed - New section "Specification Version" (3.1.2) introduced - "Table of contents" (3.1.5): Reference to Standard TOC and "intentionally left blank" introduced - "Documents" (3.1.6): Reference numbers (sections) corrected - "Attachments" (3.1.73.1.6): Reference numbers (sections) corrected "Additional files description" (3.1.8): Text slightly enhanced - New section "Standard Table of Contents" (3.2) introduced. - "Version Handling of submission" (3.3): Enforcement of complete TOC structure for complete and incremental submissions introduced - "Directory Structure and Volume Handling" (3.4): Recommendation for max. 200 chars for paths and restriction to 500 files per directory; Standard TOC xsd/xsl file added. - "File References" (3.7): Prefix "**Error! Hyperlink reference not valid.**" removed to have consistent examples - "Common file formats" 3.8: Used naming for files adapted to match used section titles of 3.8.x; directory structure of XML files changed; filename restrictions softend; missing "oecd-data" added - "CADDY-*xml* in a Web-Browser" (3.9): Restriction to MSIE 6.0+ and Mozilla Firefox 1.5+ introduced - <caddy-xml> (4.1.x): Attribute 'xmlVersion' introduced

Version	Date	Author	Status
			- \<version\> (4.2.x): Missing 'additional-file-list' child added
			- \<header\> (4.3.x): Sequence of child elements in definition and example adapted to sequence of description table
			- \<active-substance\> (4.6.x): Attribute length for 'cas' in definition corrected; sequence of attributes in example adapted
			- \<concentration\> (4.7.x): Mandatory flag in attribute table for attribute 'productId' and 'substanceId' changed to 'true' to match definition
			- \<toc-entry\> (4.8.x): Reference to Standard TOC introduced
			- \<toc-entry\> (4.9.x): New attributes 'intentionallyLeftBlank' and 'intentionallyLeftBlankComment' introduced; Example shortened
			- \<hyperlink\> (4.11.x): Reference to CADDY standard; attribute 'targetDestination' introduced; attribute 'title' in attributes table and definition corrected; changed example with 'targetDestination'
			- \<document\> (4.13.x): Duplicate attribute 'changedVersion' from definition removed; prefix "**Error! Hyperlink reference not valid.**" removed within example
			\<report-data\> (4.14.2): Name of attribute 'dossierFilenumber' in attribute table corrected
			- \<attachment\> (4.15.2): Attribute type for 'addedVersion' corrected to 'VersionNumber'; example corrected, "**Error! Hyperlink reference not valid.**" removed
			- Old section 4.16 'The version-history-list Element' removed
			- Old section 4.17 'The version-history Element' removed
			- \<additional-file-list\> (4.16.x): Attribute sequence in example changed
			- \<additional-file\> (4.17.x): Attribute sequence in example changed
			- New section 'The xmlVersionNumber Type' (4.18.1) inserted
			- Old section 4.20.10 "The elementType Type" removed
			- New chapter 5 "The CADDY-xml Standard TOC Definitions" introduced
			- "Customisation of Images" (**Error! Reference source not found.**): New size for
			tab-background.gif
			- Overall: Formatting of XML code
			- Overall: Slight changes in wording or removal of spelling or grammar mistakes

Version	Date	Author	Status
1.1.0 prop. 4	12/11/2007	Andreas Wastl	- Overall replacement of ecpa.be with ecpa.eu - "Hyperlinks" (3.1.9): Prefix "**Error! Hyperlink reference not valid.**" removed to have consistent examples - Version Handling of Submissions (3.3): Hyperlink handling for deleted toc entries; No more inclusion of deleted files in major submissions, only document references. - File Refrences (3.7): Use of service part of URL (file:// or http://) stated as forbidden. - Original Files (3.8.1): Only compliance with PDF/A-1b required - Attachments (3.8.2) / Additional Files (3.8.3): Recommended restriction of allowed characters and necessary agreement between sender and receiver. - CADDY-*xml* in a Web-Browser (3.9): Restriction to support of Firefox for MS Windows - Example (4.1.4): Use of ISO-8859-15 (as this supports some characters which are missing in ISO-8859-1. - <toc-entry> (4.9.2>: IntentionallyLeftBlankComment only allowed for intentionallyLeftBlank 'true' - Versioning Example (6.3.x.2): Adaption to changes within 3.3.
1.1.0 prop. 5	28/01/2008	Andreas Wastl	- Attributes (4.11.2): Accidentally deleted sourceType added again. - The dossier Type (4.18.9): Mixture of "volume set ID" and "dossierID" corrected and clearified. - Original Documents (3.8.1): Typo "doucuments" corrected.
2.0.0 final proposal	25/07/2008	Andreas Wastl	- New Version Number 2.0.0 for Specification to allow unique version numbering with Caddy Conversion Software; - Change of all sample version numbers within XML fragments from 01.01.00 to 02.00.00 - New files suffixes "2-0" for xsd and xsl files - <additional-files-list> element consistently corrected (missing "s" appended) to match existing caddy.xsd and technical examples - All attribute names with first char in lower caps (5.1.1 / 5.1.2) - New standard viewer files included (6.1.2) - <submission> corrected to <version> (4.1.4) - Missing quotes for attribute 'xmlns:xsi' added (4.1.4) - "XSLT transformation" corrected to "XSL transformation" (6.2)
2.0.1	11/08/2008	Andreas	Clarification in section 3.8.1 about file names and locations

Version	Date	Author	Status
final proposal		Wastl	for XML documents.
3.0.0 final	30/09/2009	Andreas Wastl	- Name Change to CADDY-*xml* (whole document) - Version references and version parts within filenames changed to 3.0 accordingly (whole document) - Removal of attribute 'title' from report data (4.14.x) - Clearification for report data attributes 'companyFilenumber' (4.14.2)
3.0.1 final	16/12/2009	Andreas Wastl	- Changed term "unique number" to "unique identifier" for xs:id (whole document) - Corrected example value "15 g/m" to "15 mg/l" (4.7.2) - Change in Headline: 3.0.0 → 3.0.x (1.5)
3.0.2 Final	17/05/2010	Markus Kriegbaum	- Correct ECPA name on header page - Changed output-format from ISO-8859-15 to UTF-8 in chapter 4.1.4 (example)
3.0.3 Final	22/06/2010	Markus Kriegbaum	- Chapter 4.4.4: changed "de" to "DE" (countrycode) in sample
3.0.4 Final	12/07/2010	Markus Kriegbaum	- Chapter 4.18.4: changed lower to upper case [a-z] → [A-Z] - Chapter 4.3.4: changed "ch" to "CH" (usage of upper case required for Rapporteur)
3.0.5 Final	25/01//2012	Andrea Oermann	- Chapter 3.8.1: changed "Additionally the filename must contain only lowercase characters, numbers, hyphens or underscore characters. It can be expressed with the following regular expression ([a-z] | [0-9] | [-] | [_])*." to "Additionally the filename must contain only lowercase or uppercase characters, numbers, dots, hyphens or underscore characters. It can be expressed with the following regular expression ([A-Z] | [a-z] | [0-9] | [.] | [-] | [_])*." - Chapter 4.13.2: changed to checksum mandatory "yes/no" and to "Hold a md5 checksum that uniquely identifies the submitted core PDF file for this document. If incremental submissions (minor version scenarios) with only document metadata but no content (core PDF file) is provided the md5 is optional." - Chapter 4.14.2: changed to "Is report GLP conform?" to "Is report GLP and GEP conform?"
3.0.6. prop. 2	18/12/2012	Georg Schifferdecker	- Chapter 3.8.1: relaxed file naming convention: removed constraint that document must be named using the ID. Added sentence: "The filenames does not need to contain the ID in its filename. Changed example for valid file name. Regrouped statements about folder and file names in chapter 3.7. This change requires a new version of the rules set for the CADDY-xml Conformity Check. - Chapter 4.18.9: changing the definition of the dossierID.

Version	Date	Author	Status
			Current dossier number may now be between 3 and 5 characters long (has been 3 characters before), complete dossier ID between 8 and 13 characters. This requires a change in the XSD.
			- Chapter 2.1. removed textual description of obsolete element version-history-list
			- Chapter 4.11.2. changed description of the hyperlink ID attribute, contained copy/paste error
			- Chapter 4.11.2, 4.11.3, 4.11.4: changed attribute "title" of element "hyperlink" to be mandatory as in existing XSD, updated description, definition and examples.
			- Chapter 5.2: changed description of toc element as single parent element of all toc-entry nodes.
			- Introduction of new file naming convention, no more restriction to use ID as part of file or folder names. See chapter 3.7.
			- Removed download information for TOCs in chapter 3.2.5, as currently there are no official TOC for download.
			- Deleted comment "If incremental submissions (minor version scenarios) with only document metadata but no content (core PDF file) is provided the md5 is optional" for MD5 attribute in the document type attributes in chapter 4.13.2. Rationale: There must be no submission with only document meta data, this is not allowed according to the specification.
			- Changed definition of hyperlink title to be string100, to be consistent with XSD
			- Added new chapter 2.2 about versioning of CADDY-xml artefacts
			- Modified chapter 3.4 to clearly separate between mandatory and optional files of a submission. Removed CADDY-xml viewer files from being mandatory files that are controlled by CADDY-xml Conformity Check.
3.0.6. final	15/01/2013	Georg Schifferdecker	- Update of chapter 3.7: Changed pattern for file names and file paths, added recommendations for file name changes, updated examples.
			- Update of chapter 3.8 to clarify that chapter 3.7 applies to all original documents (PDF and XML), attachments and additional documents.
03.06.00	15/05/2013	Georg Schifferdecker	- Removed inconsistency: Changed usage of versioning scheme of specification compliant to the initial description in chapter about xmlversionNumberType that specifies the versioning scheme of a specification. Before 03.06.00 the versioning schema for attributes in an XML file was used differently than for the filename itself, e.g. the file caddy-3-0.xsd was referenced with "xmlVersion=03.00.00". This inconsistency is resolved with this version.
			As consequence the specification version 3.0.6 became

Version	Date	Author	Status
			version 03.06.00, the last digits would be patch level versions.
			This version contains the correction of all examples or references in text from "3.0.6" to "03.06.00".
			As the viewer is an independent artefact that can change independent on this specification the list of mandatory files was shortened, the list of viewer files has been removed. A submission is valid without supplied viewer or a different viewer.
			The chapters about viewer customizations have been removed. Viewer customizations should not be part of the format specification, when a viewer is mandatory element of the submission, but the viewer version is undefined during the definition of the format specification.
			Please note that the value of the attribute noNamespaceSchemaLocation of the element caddy-xml contains the filename and has to be set to "xsi:noNamespaceSchemaLocation="utils/caddy_03-06-00.xsd" for this version.
			Added comment to chapter 4.13.2 (md5 checksum attribute): For complete submission the value has to be provided for all referenced content files. For incremental submission, the checksum has to be provided for referenced content files that have changed and part of the current submission, not for previously submitted content that is also referenced from the XML backbone.
			Addition to chapter 3.1.6 with details about documents that are included multiple times in a submission.
			Reworked chapter 6.2 and subchapter to apply to the latest CADDY-xml viewer (e.g. mandatory files).
			Removed chapters after chapter 6.2 about configuration possibilities of the CADDY-xml viewer.
			Added additional information for the usage of the GIFAP codes in the formulation attribute of the product element (chapter 4.5.2)
			Corrected cardinality between document and document list in figure of Table of contents and complete figure at beginning of chapter 4: There must be at least one document in document list.
			Corrected text in chapter 4.2: The element additional-file-list was missing in the listing of the child elements.
03.06.01	20/01/2014	Georg Schifferdecker	Removed the following statement: "Furthermore a directory must not have more than 500 files in total. Depending on the affected directory such files need to be put into additional subdirectories instead." Rationale: There is no longer a justification for file limitations in a directory, as coming from CADDY-V2. Current submissions are media independent. This eases the compilation of dossiers.

Version	Date	Author	Status
03.07.00	14/03/2014	Georg Schifferdecker	The checksum attribute was set to optional for all usages, in addition to changes applied in chapter 03.06.00. It was still mandatory for the element attachment and additional-file. The XSD has been changed to set all occurences of checksum to "optional". The new XSD carries the version 03.07.00.
03.08.00	27/062019	Georg Schifferdecker	Removal of the embedded CADDY-xml viewer, replaced by the CADDY-xml viewer module and the eSubmission viewer. Corresponding changes in the file system structure, e.g. removal of index.html. Support of new file naming "main.caddy" next to "caddy.xml" for the CADDY XML backbone file. Addition of the mandatory static HTML representation caddy.html in the root directory, introduction of the XSLT file that is to be used for the generation. Removal of outdated software CADDY Conformity Check, replacement with eSubmission validator. Clearer separation between CADDY-xml specification artefacts and further associated artefacts (e.g. eSubmission Viewer, Converter) – see chapter 2.2 Replacement of strict PDF/A-1b compliance with the definition of a "CADDY-xml PDF profile" Change of the attribute "title" in the element "toc-entry" from string100 to string250 (chapter 4.9.2) → update of CADDY-xml XSD, new XSD version is 03.08.00
03.08.01	17/09/2019	Georg Schifferdecker	Clarification for the required stylesheet reference to caddy.xsl added, also for specification version beyond 03.08.00, when embedded viewer module is outdated. Correction of unresolved textual references