



United Nations Centre for Trade Facilitation and
Electronic Business

5

10

15

**UN/CEFACT
Exchange Header Envelope (XHE) Version 1.0**

20

PUBLIC DRAFT

Version 1.0

25

2018-02-28

Abstract

This specification defines a business-oriented artefact either referencing (as a header) or containing (as an envelope) a payload of one or more business documents or other artefacts with supplemental semantic information about the collection of payloads as a whole. This is distinct from any transport-layer infrastructure header or envelope that may be required to propagate documents from one system to another. An exchange header envelope describes contextual information important to the sender and receiver about the payloads, without having to modify the payloads in any fashion.

Contents	Page
1 Introduction	5
1.1 Conveying information about payloads	5
1.1.1 Overview	5
1.1.2 What is an Exchange Header Envelope?	5 5
1.1.3 How is it used in EDI and XML environments?	5
1.1.4 The Scope of the Exchange Header Envelope	6
1.1.5 Stakeholders and Audience	6
1.2 Terminology	7
1.2.1 Key words	7 10
1.2.2 Terms and Definitions	7
1.2.3 Symbols and Abbreviations	7
1.2.4 Key concepts	8
1.3 Normative References	8
1.4 Informative References	8 15
2 Header and envelope information	9
2.1 XHE class diagram	9
2.2 Header envelope information	9
2.3 Header information	10
2.4 Party information	10 20
2.5 Party identification information	10
2.6 Business scope information	11
2.7 Business scope criterion information	11
2.8 External reference information	11
2.9 Payload information	11 25
2.9.1 Payload set information	11
2.9.2 Payload item information	11
2.10 Extension information	12
3 XHE digital signatures	13
3.1 Signing the exchange header envelope	13 30
4 Document model expression	14
5 XML syntax expression	14
5.1 Schema expression	14
5.2 XML namespaces	14
5.3 The schema subdirectories	14 35
5.4 The header envelope schema	15
5.5 Non-content data type common schemas	15
5.6 Content data type common schemas	16
5.6.1 Modifiable schema fragments	16
5.6.2 Extension content	16 40
5.6.3 Payload content	16
5.7 Unqualified data type attributes	17
6 Conformance	20
Annex A (informative) Package structure	22
Annex B (informative) Revision History	23 45
Annex C (informative) Demonstration XML environment	24
Annex D (informative) Encrypting payloads to multiple recipients	29

Annex E (informative) Acknowledgements 30
Annex F (informative) Temporary Annex - Change Log 31

List of Figures

Page

Figure 1 Exchange Header Envelope class diagram 9

1 Introduction

1.1 Conveying information about payloads

1.1.1 Overview

File or document headers have long been used to describe the information about a set of payloads in an entity that is kept separate and arm's-length from the payloads themselves. 5

The metaphor of a paper envelope in which one places business documents for transport or management is apt to describe the role of an exchange header envelope in a container relationship to its payloads. Concepts of routing, authentication, non-repudiation and concealment all apply in both the metaphor and the electronic equivalent.

1.1.2 What is an Exchange Header Envelope? 10

The Exchange Header Envelope (XHE) specifies an XML vocabulary [\[XML\]](#) expressing in machine-processable syntax the semantics of describing either a header to or an envelope of a set of payloads of content with information about that content. This vocabulary is modeled using the UN/CEFACT Core Component Technical Specification Version 2.01 [\[CCTS 2.01\]](#).

XHE, a specification developed jointly by UN/CEFACT and OASIS, is the successor to the UN/CEFACT Standard Business Document Header (SBDH) version 1.3 [\[SBDH\]](#) and the OASIS Business Document Envelope (BDE) version 1.1 [\[BDE\]](#). 15

Note regarding publications The UN/CEFACT Exchange Header Envelope and the OASIS Exchange Header Envelope are the same specification developed in collaboration and published as standards by the two organizations following the practices of each. 20

This specification enumerates the information components of a payload header envelope and formally describes the semantics of each component.

This specification mandates a suite of XML schemas [\[XSD1\]\[XSD2\]](#) and additional limitations describing the document constraints against which a conforming instance SHALL validate without error. 25

Normative markings All clauses not marked as "informative" and also not a subclause of a clause marked as "informative" are to be considered normative. All notes and examples are informative.

The XHE is designed to be either a header as an integral part of a business document (e.g. either XML instance document or EDI interchange), an object associated with the business document itself, or as an envelope functioning as wrapper that contains one or more business documents. 30

1.1.3 How is it used in EDI and XML environments?

There exist several business document exchange architectures and approaches, some using EDI formats and approaches, some using XML document types, and yet others use different document formats or non-standardized approaches. The XHE is designed to work with any document format and business process, whether standardized or not, and as such supports both the EDI, the XML and any other e-business community. Including a XHE in each instance of the business document reduces the effort needed to route and process documents and permits trading partner organizations to use different implementation approaches. 35

When implementing EDI, the provision of an additional business document header may not always be necessary, since EDI interchanges already contain functionality for some of the information in 40

the XHE. An example is the EDIFACT UNB interchange header, the UNH message header, and the 'function' part of the BGM. The XHE specification allows for this existing approach and provides an option to express additional functionality, such as service and correlation information.

1.1.4 The Scope of the Exchange Header Envelope

Many users, implementers and supporting industry standard bodies are in agreement on the need for an Exchange Header Envelope. In their business-to-business activities, the XHE facilitates several different business needs:

- The routing of business documents from one point to another. This refers not only to the transfer of information from an external originator to receiver, but also from one intermediate application to another. Information in the XHE can help ensure that a document gets to the correct recipient. 10
- Ensuring integrity and confidentiality of business documents when routed over multiple hops, intermediaries, routers or access points, such as in 4-corner networks and architectures.
- Simplifying the bundling of several business documents or support documents into one package for simplified exchange. 15
- Facilitating the exchange of location pointers and access credentials to externally located business documents, not suitable for sending through an e-business network. This is necessary when the sending party needs to keep the business document confidential until a specified date (such as in tendering processes), and when sending very large files. 20
- The simplified processing of documents. Processing refers to taking action on data, for example transforming it from one format into another. Information in the XHE can reduce the effort required to determine the correct processing actions.
- Associating a data message with its originator is important from a business and legal perspective. It is especially important when using intermediaries for data transfer, as information from the transport protocol, may be lost after the initial transmission. Because information in the XHE is retained, it can help ensure that a document's originator is correctly identified. 25

In addition to header functions provided by the XHE for routing and/or processing of business documents, there is the need for a completely separate technical communications transport layer. This deals with communications protocols and physical addresses which are outside the scope of this technical specification. Transport specifications including EDIINT-AS2 and ebXML Message Service (ebMS) are among a number of possible transport options that address technical communications needs by defining a separate technical header. The transport layer is completely outside the scope as it is a different layer of the stack. 30 35

1.1.5 Stakeholders and Audience

All organizations that manage infrastructure operations and business processes for various functional areas (e.g. ordering, invoicing, planning, or financial), all service provider organizations and associations, as well as e-business networks and infrastructures, which create, route and process business documents can benefit from the use of the Exchange Header Envelope. 40

1.2 Terminology

1.2.1 Key words

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, OPTIONAL and SHOULD in this document are to be interpreted as described in [\[RFC 2119\]](#).

1.2.2 Terms and Definitions

5

1.2.2.1

schema

An expression of constraints placed on XML content.

1.2.2.2

value constraints

An expression of constraints placed on the values of attributes and textual content.

10

1.2.3 Symbols and Abbreviations

1.2.3.1

ABIE

Aggregate Business Information Entity

15

1.2.3.2

BBIE

Basic Business Information Entity

1.2.3.3

ASBIE

Association Business Information Entity

20

1.2.3.4

RFC

Request for comment

1.2.3.5

XSD

XML Schema Definition

25

1.2.3.6

XSLT

Extensible Stylesheet Language Transformations

30

1.2.4 Key concepts

1.2.4.1

validation

The act of testing an XML document against a set of structural constraints (as expressed in a schema) or value constraints (as expressed in an arbitrary XML processing language, for example, XSLT). 5

1.3 Normative References

CCTS 2.01 UN/CEFACT Core Components Technical Specification, Version 2.0115 November 2003 http://www.unece.org/fileadmin/DAM/cefact/codesfortrade/CCTS/CCTS_V2-01_Final.pdf

RFC 2119 Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, BCP 14, RFC 2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>. 10

XML Extensible Markup Language (XML) 1.0 (Fifth Edition), Bray T. , Paoli J. , Sperberg-McQueen C. M. , Maler E. , Yergeau F. , Editors, W3C Recommendation, 26 November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>. Latest version available at <http://www.w3.org/TR/xml>.

xmldsig XML-Signature Syntax and Processing Version 1.1, Eastlake D. E. , Reagle J. , Solo D. , Hirsch F. , Nyström M. , Roessler T. , Yiu K. , Editors, W3C Recommendation, 11 April 2013, <http://www.w3.org/TR/2013/REC-xmldsig-core1-20130411/>. Latest version available at <http://www.w3.org/TR/xmldsig-core1>. 15

XSD1 XML Schema Part 1: Structures Second Edition, Thompson H. S. , Beech D. , Maloney M. , Mendelsohn N. , Editors, W3C Recommendation, 28 October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>. Latest version available at <http://www.w3.org/TR/xmlschema-1>. 20

XSD2 XML Schema Part 2: Datatypes Second Edition, Biron P. V. , Malhotra A. , Editors, W3C Recommendation, 28 October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>. Latest version available at <http://www.w3.org/TR/xmlschema-2>.

1.4 Informative References

25

BDE Business Document Envelope Version 1.1 Edited by G. Ken Holman and Kenneth Bengtsson. 05 December 2016. OASIS Committee Specification 01. Latest version: <http://docs.oasis-open.org/bdxr/bdx-bde/v1.1/bdx-bde-v1.1.html>.

CMS Cryptographic Message Syntax (CMS), Housley R. , September 2009, <https://tools.ietf.org/html/rfc5652>. 30

genericode Code List Representation (Genericode) Version 1.0. Edited by Anthony B. Coates. 28 December 2007. Committee Specification 01. <http://docs.oasis-open.org/codelist/genericode/>. Latest version: <http://docs.oasis-open.org/codelist/genericode/doc/oasis-code-list-representation-genericode.html>.

PGP OpenPGP Message Format, Callas J. , Donnerhackle L. , Finney H. , Shaw D. , Thayer R. , November 2007, <https://tools.ietf.org/html/rfc4880>. 35

SBDH CEFACT Standard Business Document Header https://www.gs1.org/sites/default/files/docs/gs1_un-cefact_%20xml_%20profiles/CEFACT_SBDH_TS_version1.3.pdf

XAdES XML Advanced Electronic Signatures. ETSI TS 101 903 V1.4.1, June 2009 http://uri.etsi.org/01903/v1.4.1/ts_101903v010401p.pdf

XMLEnc XML Encryption Syntax and Processing, Eastlake Donald , Reagle Joseph , 10 December 2002, <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>.

5

2 Header and envelope information

2.1 XHE class diagram

The information derived for XHE has been distilled into a suite of CCTS Aggregate Business Information Entities (ABIEs), each comprised of a set of Basic Business Information Entities (BBIEs) and/or Association Business Information Entities (ASBIEs).

10

The relationships between these business information entities are depicted in this class diagram:

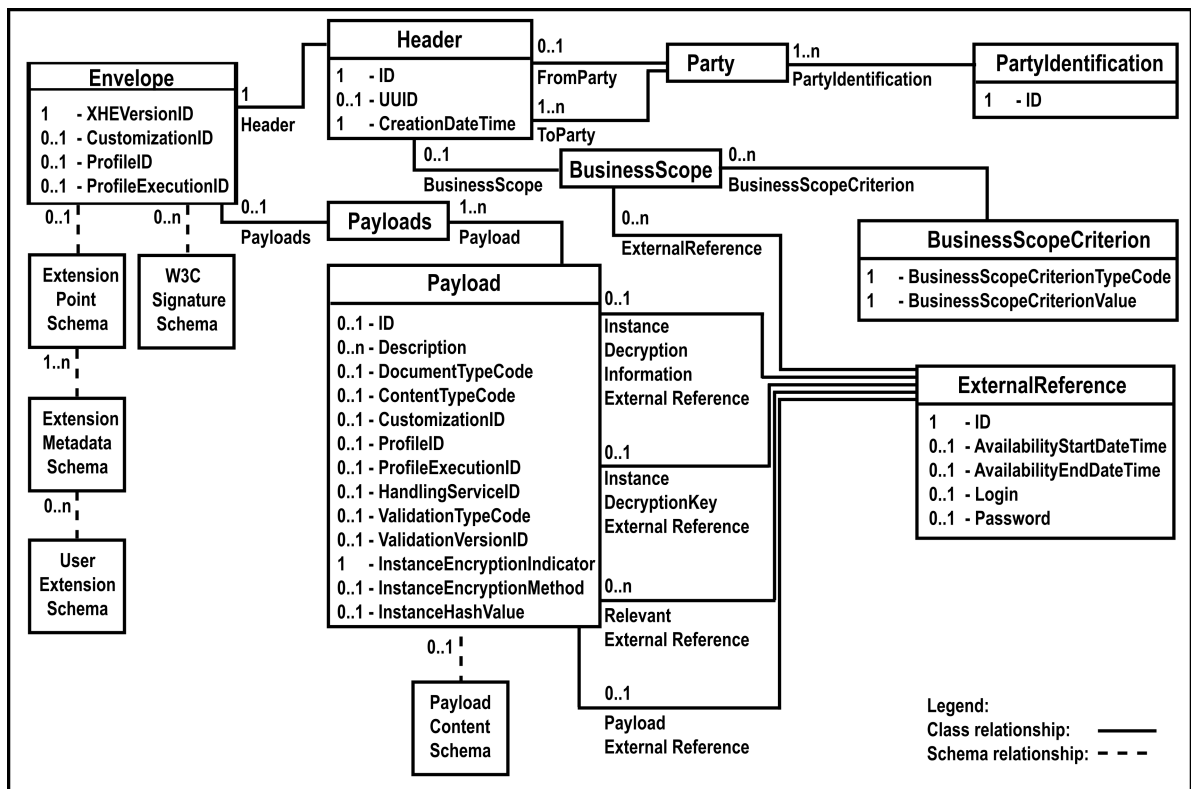


Figure 1 — Exchange Header Envelope class diagram

2.2 Header envelope information

Metadata information about the header envelope itself, independent of the information it contains, includes the following:

15

Semantic identifier	Card.	Definition
XHE. XHE Version Identifier. Identifier	1	The version of the specific envelope model in use.
XHE. Customization Identifier. Identifier	0..1	The identification of a customization or use of the envelope model.
XHE. Profile Identifier. Identifier	0..1	The identification of a specific profile found within the customization.

Semantic identifier	Card.	Definition
XHE. Profile Execution Identifier. Identifier	0..1	The identification of a particular instance of using the given profile.
XHE. Header	1	Information relevant to the handling of the envelope.
XHE. Payloads	0..1	The set of payloads

NOTE The set of payloads can be omitted to allow the XHE either to be embedded in a document that contains the actual payloads (as in the Standard Business Document) or in a standalone document entirely separate from the payloads (as in the standalone Standard Business Document Header). When present, the set of payloads is wrapped by the XHE document itself (as in the Business Document Envelope).

The information for a header envelope begins with an additional optional extensions item that is neither a BBIE nor an ASBIE and so is not modeled using CCTS. Rather, this extensions item is a schema artefact. This extensions item has the cardinality 0..1. See [5.6.2 Extension content](#) for more information. 5

The information for a header envelope ends with an additional optional and repeatable number of digital signatures that are neither a BBIE nor an ASBIE and so are not modeled using CCTS. Rather, these signatures are a schema artefact published by the W3C. See [3.1 Signing the exchange header envelope](#) for more information. 10

2.3 Header information

Metadata information about the header envelope itself, independent of the information it contains or references, includes the following: 15

Semantic identifier	Card.	Definition
Header. Identifier	1	Unique ID of the envelope for tracking purposes.
Header. UUID	0..1	An additional identifier of the envelope.
Header. Creation Date Time. Date Time	1	Date and time when the envelope was created.
Header. Business Scope	0..1	Documentation of the scope of business or other contextual details useful to understand the purpose of the envelope and its contents. For examples: Europe vs Asia, Direct-to-Consumer vs Replenishment, or Prepaid vs Credit.
Header. From_ Party. Party	0..1	Information about the party that originated the envelope.
Header. To_ Party. Party	1..n	Information about the parties to receive the envelope.

2.4 Party information

The information about a party includes the following:

Semantic identifier	Card.	Definition
Party. Party Identification	1..n	Unambiguous identifications of a party.

2.5 Party identification information

The information about a party's identification includes the following:

Semantic identifier	Card.	Definition
Party Identification. Identifier	1	An unambiguous identification of a party.

2.6 Business scope information

Documentation of the scope of business or other contextual details useful to understand the purpose of the envelope and its contents includes the following:

Semantic identifier	Card.	Definition
Business Scope. Business Scope Criterion	0..n	Internal specification of the scope and/or context of business.
Business Scope. External Reference	0..n	External documentation of the scope and/or context of business.

2.7 Business scope criterion information

Documentation of one criterion of the scope of business or other contextual detail useful to understand the purpose of the envelope and its contents includes the following:

Semantic identifier	Card.	Definition
Business Scope Criterion. Business Scope Criterion Type. Code	1	Identifies the property of the scope by a code.
Business Scope Criterion. Business Scope Criterion Value. Text	1	Specifies the value of the given property.

2.8 External reference information

A reference to a business case, document or other issues which are relevant to the handling of the envelope includes the following:

Semantic identifier	Card.	Definition
External Reference. Identifier	1	Identifies the referenced object by some identifier or URI.
External Reference. Availability Start Date Time. Date Time	0..1	The start date and time when the information is available
External Reference. Availability End Date Time. Date Time	0..1	The end date and time when the information is available
External Reference. Login. Text	0..1	Text describing any login details to access the information.
External Reference. Password. Text	0..1	A password needed to access the information.

2.9 Payload information

10

2.9.1 Payload set information

Information about the complete set of payloads includes the following:

Semantic identifier	Card.	Definition
Payloads. Payload	1..n	The actual payload instance, such as a single invoice, conveyed within the envelope.

2.9.2 Payload item information

Information about an individual payload within the set of payloads includes the following:

Semantic identifier	Card.	Definition
Payload. Identifier	0..1	A unique identification of this payload instance contained within the envelope.
Payload. Description. Text	0..n	Text description of the payload instance.
Payload. Document Type Code. Code	0..1	Identifies the abstract archetype of the payload instance.
Payload. Content Type Code. Code	0..1	Identifies the file format or octet representation of the payload instance.
Payload. Customization Identifier. Identifier	0..1	Identifies the customization that applies to the payload instance.
Payload. Profile Identifier. Identifier	0..1	Identifies the profile that the payload instance is part of.
Payload. Profile Execution Identifier. Identifier	0..1	Identifies the particular instance of an executing profile that the payload instance is part of.
Payload. Handling Service Identifier. Identifier	0..1	Identifies the service that should process the payload instance.
Payload. Validation Type. Code	0..1	The validation type of the payload, used for the task of verifying that the grammar of a payload is valid.
Payload. Validation Version Identifier. Identifier	0..1	Descriptor containing version information of the validation type.
Payload. Instance Encryption Indicator. Indicator	1	Indicator to state whether the payload instance is encrypted or not.
Payload. Instance Encryption Method. Text	0..1	Method used to encrypt the payload instance.
Payload. Instance Hash Value. Text	0..1	SHA-256 hash total of the unencrypted payload instance.
Payload. Instance Decryption Information_ External Reference. External Reference	0..1	Decryption information that is available external to the envelope.
Payload. Instance Decryption Key_ External Reference. External Reference	0..1	Decryption key data that is available external to the envelope.
Payload. Relevant_ External Reference. External Reference	0..n	A reference to a business case, document or other issues which are relevant to the handling of the payload.
Payload. Payload_ External Reference. External Reference	0..1	The reference to the payload when it is not included within the envelope.

The information for an individual payload ends with an additional optional payload content item that is neither a BBIE nor an ASBIE and so is not modeled using CCTS. Rather, this content item is a schema artefact. This content item has the cardinality 0..1. This content item can have as its child either text only (no elements) or a single element, but not a combination of both nor more than one element. See [5.6.3 Payload content](#) for more information.

5

2.10 Extension information

Through the use of extension metadata and content, additional user-defined information that is not modelled by the CCTS classes can be added to the envelope instance.

The extension point is an optional construct as the initial child of the document element. The extension point, when it exists, SHALL contain one or more user-defined extensions,. Each extension contains optional extension metadata identifying properties of the extension as well as the extension content.

10

Name	Description	Crd	Rationale
------	-------------	-----	-----------

(Unqualified Data Type)			
XHEExtensions	A container for all extensions present in the document.	0..1	This is the single point of access to all extensions as the first child of the main document.
XHEExtension	A single extension for private use.	1..n	There may be many extensions added to a single document.
ExtensionID (Identifier)	An identifier for the Extension assigned by the creator of the extension.	0..1	This identifies the extension amongst other extensions within the document.
ExtensionName (Name)	A name for the Extension assigned by the creator of the extension.	0..1	This identifies the extension in natural language within the document.
ExtensionAgencyID (Identifier)	An agency that maintains one or more Extensions.	0..1	This identifies who created the extension.
ExtensionAgencyName (Name)	The name of the agency that maintains the Extension.	0..1	This identifies who created the extension.
ExtensionAgencyURI (Identifier)	A URI for the Agency that maintains the Extension.	0..1	This identifies who created the extension.
ExtensionVersionID (Identifier)	The version of the Extension.	0..1	This distinguishes one version of the extension from another.
ExtensionURI (Identifier)	A URI for the Extension.	0..1	This identifies the extension amongst other extensions outside of any document.
ExtensionReasonCode (Code)	A code for the reason the Extension is being included.	0..1	This gives the author the opportunity to give rationale by way of a code.
ExtensionReason (Text)	A description of the reason for the Extension.	0..1	This gives the author the opportunity to give rationale by way of a text description.
ExtensionContent	The definition of the extension content.	1	This is the parent element of the extension content.

There are no restrictions on the extension content. See [5.6.2 Extension content](#) for more information.

3 XHE digital signatures

3.1 Signing the exchange header envelope

Using the IETF/W3C XML Digital Signature specification [\[xmldsig\]](#) one can add multiple “non-final” signatures or a single “final” signature to the exchange header envelope as the last children of the document element, that is, after the last BIE of the document element. A non-final signature digitally signs all content other than any of the other sibling signature elements that may exist in the document. A final signature digitally signs all content including the other sibling signature elements that may exist in the document.

The schema fragment for [5.6.2 Extension content](#) included in this distribution provides for using digital signature extensions supporting XML Advanced Electronic Signatures [\[XAdES\]](#) (ETSI TS 101 903), when the electronic signing of an exchange header envelope is necessary to satisfy legal and technical requirements. The schema fragment can be modified to accommodate such future extension requirements without impacting on the conformance clauses of this specification.

4 Document model expression

The document model is expressed in four ways, found in four files of the model subdirectory:

- [mod](#)
 - [XHE-Model-1.0.ods](#)
 - model information expressed in an Open Office spreadsheet 5
 - [XHE-Model-1.0.xls](#)
 - model information expressed in an Excel spreadsheet
 - [XHE-Model-1.0.gc](#)
 - model information expressed in a genericcode [\[genericcode\]](#) file
 - [XHE-Model-1.0.html](#) 10
 - model information expressed in hyperlinked human-readable tables as HTML

5 XML syntax expression

5.1 Schema expression

The structural document constraints of the header envelope are expressed normatively as a set of W3C XSD XML Schemas [\[XSD1\]](#)[\[XSD2\]](#). 15

5.2 XML namespaces

The following XML namespace URI strings are specified in the XSD schemas to be used in the XML syntax expressions:

oasis-cefact-xhe-1.0-ExchangeHeaderEnvelope
 oasis-cefact-xhe-1.0-AggregateComponents
 oasis-cefact-xhe-1.0-BasicComponents
 oasis-cefact-xhe-1.0-ExtensionComponents
 oasis-cefact-xhe-1.0-QualifiedDataTypes
 oasis-cefact-xhe-1.0-UnqualifiedDataTypes

5.3 The schema subdirectories

The schemas are delivered in two subdirectories: 20

- [xsd](#)
 - CCTS documentation is included as XSD annotations
- [xsdr](#)
 - runtime version such that CCTS documentation is not included as XSD annotations

- without the annotations a W3C schema processor has less work to prepare for validating documents

In both subdirectories there is a single subdirectory of common files:

- [common](#)
 - included schema fragments by any document fragment 5

5.4 The header envelope schema

The following is the only Document ABIE schema:

- [xsd/XHE-1.0.xsd](#)
 - the base header envelope schema fragment that incorporates other schema fragments 10

The following is the runtime version of the schema that has documentary annotations removed:

- [xsdrt/XHE-1.0.xsd](#)
 - the base header envelope schema fragment that incorporates other schema fragments

5.5 Non-content data type common schemas 15

The following are read-only schema fragments in the common subdirectory:

- [XHE-CommonAggregateComponents-1.0.xsd](#)
 - the Library ABIE element declarations
- [XHE-CommonBasicComponents-1.0.xsd](#)
 - the Library BBIE element declarations 20
- [XHE-CommonExtensionComponents-1.0.xsd](#)
 - the Document ABIE extension metadata declarations
- [XHE-QualifiedDataTypes-1.0.xsd](#)
 - the qualified data types (empty; none are defined)
- [XHE-UnqualifiedDataTypes-1.0.xsd](#) 25
 - the unqualified data types based on the core component types
 - see [5.7 Unqualified data type attributes](#) for more details
- [XHE-xmlsig1-schema-1.0.xsd](#)
 - the XML Digital Signature schema driver fragment copyrighted by W3C
- [XHE-xmlsig11-schema-1.0.xsd](#) 30

- the XML Digital Signature schema fragment copyrighted by W3C
- [XHE-xmldsig-core-schema-1.0.xsd](#)
 - the XML Digital Signature Core schema fragment copyrighted by W3C
- [CCTS_CCT_SchemaModule-1.0.xsd](#)
 - the Core Component Types schema fragment copyrighted by UN/CEFACT 5

5.6 Content data type common schemas

5.6.1 Modifiable schema fragments

There are two content data type schema fragments in the common subdirectory, one for each of the extension content and the payload content. These are the only schemas intended to be edited by users should they wish to validate the content of their extensions or payloads. No changes are necessary to the schemas if it is not important to validate these portions of the document. 10

Should users wish to impose constraints on the extension or the payload contents, the only edits necessary of the content schema are for the importation of the schemas to be engaged for validation purposes. No edits are necessary for the content element, though one may wish to do so to exclude content other than that for which schemas are provided. 15

5.6.2 Extension content

The extension content schema fragment describes constraints on content placed in extensions.

- [XHE-ExtensionContentDataType-1.0.xsd](#)

The extension content element's name is `<{extensions prefix}:ExtensionContent>`, for example, `<ext:ExtensionContent>`. It is the last child element of `<{extensions prefix}:XHEExtension>`. 20

Any given extension content may have zero or one apex (or top-most) element in the XML element tree. The absence of content is provided for situations where a processing application chooses to remove foreign unrecognized-namespace elements from the XML element tree.

The distributed version of this file imports the version of XAdES schemas that are current at the time of publication. XAdES constructs are used within W3C XML Digital Signatures. These import directives can be replaced with the importation of future versions of XAdES schemas as needed. 25

- [XHE-XAdES01903v132-201601-1.0.xsd](#)
 - the v1.3.2 XAdES schema fragment from the etsi.org web site
- [XHE-XAdES01903v141-201601-1.0.xsd](#) 30
 - the v1.4.1 XAdES schema fragment from the etsi.org web site

5.6.3 Payload content

The payload content schema fragment describes constraints on content placed in payloads.

- [XHE-PayloadContentDataType-1.0.xsd](#)

The payload content element's name is <{aggregate prefix}:PayloadContent>, for example, <eac:PayloadContent>. It is the last child element of <{aggregate prefix}:Payload>.

Any given payload content element may have as its child exactly one apex (or top-most) element in the XML element tree, or it may consist solely of text that would typically represent encrypted content or non-XML content. Special care needs to be taken that all non-XML payload content is encoded according to XML text encoding rules, such as the escaping of special markup characters, so as to permit an XML processing application to correctly interpret the non-XML content.

The schema declarations are unable to trigger a constraint error in the situation where the payload content has a combination of both text and a single element. Detecting such a condition is the responsibility of the processing agent.

The schema declarations are unable to trigger a constraint error in the situation where the payload content is empty. Detecting such a condition is the responsibility of the processing agent.

5.7 Unqualified data type attributes

In the Exchange Header Envelope model each BBIE is indicated to have a particular component name (specifying the element name) and to be of a particular unqualified data type (specifying the base type value constraints and the attributes). Writers of extensions using CCTS and their own BBIEs need to know the available unqualified data types for their extended business objects.

Based on the 10 approved core component types described in section 8.1 of [\[CCTS 2.01\]](#), there are 20 available unqualified data types for BBIE values. Each data type has a constraint on its content (the component) and a possibly-empty selection of available possibly-mandatory attributes (the supplementary components).

NOTE Not all of the unqualified data types listed in this table are used in the standardized components of the header envelope. All defined types are enumerated here for completeness in the event that a CCTS-based extension is created by a community of users that relies on one of the unqualified data types not used by the standardized components of the header envelope.

Data Type	Base type (XSD)	Supplementary component (attribute)	Cardinality	Type (XSD)	Definition
Amount	xsd:decimal	A number of monetary units specified using a given unit of currency.			
		currencyID	required	xsd:normalizedString	The currency of the amount.
		currencyCodeListVersionID	optional	xsd:normalizedString	The VersionID of the UN/ECE Rec9 code list.
Binary Object Graphic Picture Sound Video	xsd:base64Binary	A set of finite-length sequences of binary octets.			
		mimeCode	required	xsd:normalizedString	The mime type of the binary object.
		characterSetCode	optional	xsd:normalizedString	The character set of the binary object if the mime type is text.
		encodingCode	optional	xsd:normalizedString	Specifies the decoding

				algorithm of the binary object.	
		filename	optional	xsd:string	The filename of the binary object.
		format	optional	xsd:string	The format of the binary content.
		uri	optional	xsd:anyURI	The Uniform Resource Identifier that identifies where the binary object is located.
Code	xsd:normalizedString	A character string (letters, figures, or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute, together with relevant supplementary information.			
		languageID	optional	xsd:language	The identifier of the language used in the code name.
		listAgencyID	optional	xsd:normalizedString	An agency that maintains one or more lists of codes.
		listAgencyName	optional	xsd:string	The name of the agency that maintains the list of codes.
		listID	optional	xsd:normalizedString	The identification of a list of codes.
		listName	optional	xsd:string	The name of a list of codes.
		listSchemeURI	optional	xsd:anyURI	The Uniform Resource Identifier that identifies where the code list scheme is located.
		listURI	optional	xsd:anyURI	The Uniform Resource Identifier that identifies where the code list is located.

		listVersionID	optional	xsd:normalizedString	The version of the list of codes.
		name	optional	xsd:string	The textual equivalent of the code content component.
DateTime	xsd:dateTime	An instance of time according the Gregorian calendar.			
Date	xsd:date	One calendar day according the Gregorian calendar.			
Time	xsd:time	An instance of time that occurs every day.			
Identifier	xsd:normalizedString	A character string to identify and uniquely distinguish one instance of an object in an identification scheme from all other objects in the same scheme, together with relevant supplementary information.			
		schemeAgencyID	optional	xsd:normalizedString	The identification of the agency that maintains the identification scheme.
		schemeAgencyName	optional	xsd:string	The name of the agency that maintains the identification scheme.
		schemeDataURI	optional	xsd:anyURI	The Uniform Resource Identifier that identifies where the identification scheme data is located.
		schemeID	optional	xsd:normalizedString	The identification of the identification scheme.
		schemeName	optional	xsd:string	The name of the identification scheme.
		schemeURI	optional	xsd:anyURI	The Uniform Resource Identifier that identifies where the identification scheme is located.
		schemeVersionID	optional	xsd:normalizedString	The version of the identification scheme.

Indicator	xsd:boolean	A list of two mutually exclusive Boolean values that express the only possible states of a property.			
Measure	xsd:decimal	A numeric value determined by measuring an object using a specified unit of measure.			
		unitCode	required	xsd:normalizedString	The type of unit of measure.
		unitCodeListVersionID	optional	xsd:normalizedString	The version of the measure unit code list.
Numeric Value Percent Rate	xsd:decimal	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.			
		format	optional	xsd:string	Whether the number is an integer, decimal, real number or percentage.
Quantity	xsd:decimal	A counted number of non-monetary units, possibly including a fractional part.			
		unitCode	optional	xsd:normalizedString	The unit of the quantity
		unitCodeListAgencyID	optional	xsd:normalizedString	The identification of the agency that maintains the quantity unit code list
		unitCodeListAgencyName	optional	xsd:string	The name of the agency which maintains the quantity unit code list.
		unitCodeListID	optional	xsd:normalizedString	The quantity unit code list.
Text Name	xsd:string	A character string (i.e. a finite set of characters), generally in the form of words of a language.			
		languageID	optional	xsd:language	The identifier of the language used in the content component.
		languageLocaleID	optional	xsd:normalizedString	The identification of the locale of the language.

6 Conformance

A Exchange Header Envelope instance exhibits conformance when complying with all of the following criteria:

- 1) The instance SHALL NOT violate any document constraints expressed by the schema in [5.4 The header envelope schema](#)
- 2) Any XML element that is not extension content SHALL NOT be empty.
- 3) The `<{aggregate prefix}:Payload` element SHALL have one or the other of the `<{aggregate prefix}:PayloadContent>` element child or the `<{aggregate prefix}:PayloadExternalReference>` element child (that is, it SHALL NOT have both).
- 4) The `<{aggregate prefix}:PayloadContent>` element SHALL NOT have a combination of text and an element (that is, it SHALL either be a non-empty string of text or be a single element).

Annex A (informative)

Package structure

This Committee Specification Public Review Draft 01 is published as a zip archive in the <http://docs.oasis-open.org/bdxr/bdx-xhe/v1.0/csprd01/> directory. Unzipping this archive creates a 5 directory tree containing a number of files and subdirectories. Note that while the two XML files comprise the revisable version of this specification, this revisable XML may not be directly viewable in all currently available web browsers.

The base directory has the following files:

- `bdx-xhe-v1.0-csprd01.xml` 10
 - The revisable form of the document.
- `bdx-xhe-v1.0-csprd01.html`
 - An HTML rendering of the document.
- `bdx-xhe-v1.0-csprd01.pdf`
 - A PDF rendering of the document. 15

These are the informative subdirectories in the package:

- `art`
 - Diagrams and illustrations used in this specification.
- `db`
 - DocBook stylesheets for viewing in HTML the XML of this work product. 20
- `val`
 - Demonstrative validation of the example instances with the header envelope schemas.

See [Annex C Demonstration XML environment](#) for details.

The normative subdirectories in the package are listed in normative clauses. 25

Annex B (informative)

Revision History

B.1 Major version XHE 1.0

XHE version 1.0 Committee Specification 01 establishes the namespaces to be used for all subsequent minor revisions of the Exchange Header Envelope.

Annex C (informative)

Demonstration XML environment

A working example of using the schemas with an XML instance is demonstrated in the `val/` directory. 5

The following support files are in this directory:

- [test.bat](#)
 - Windows invocation of the testing of the sample files
- [test.sh](#)
 - shell invocation of the testing of the sample files 10
- [validate.bat](#)
 - Windows invocation of the two-pass validation of a single file
- [validate.sh](#)
 - shell invocation of the two-pass validation of a single file
- [w3cschema.bat](#) 15
 - Windows invocation of the schema validation of a single file
- [w3cschema.sh](#)
 - shell invocation of the schema validation of a single file
- [xslt.bat](#)
 - Windows invocation of XSLT transformation on a single file 20
- [xslt.sh](#)
 - shell invocation of XSLT transformation on a single file

This directory has a number of simple test files:

- [simpleExample.xml](#)
 - a simple envelope with three payload instances, the second of which is simple text 25
(note the escaped special characters) and the other two of which are XML

```
<?xml version="1.0" encoding="UTF-8"?>
<XHE xmlns="oasis-cefact-xhe-1.0-ExchangeHeaderEnvelope"
      xmlns:xhb="oasis-cefact-xhe-1.0-BasicComponents"
      xmlns:xha="oasis-cefact-xhe-1.0-AggregateComponents"
      xmlns:ext="oasis-cefact-xhe-1.0-ExtensionComponents">
```

30


```

<xhb:XHEVersionID>1.0</xhb:XHEVersionID>
<xha:Header>
  <xhb:ID>123</xhb:ID>
  <xhb:CreationDateTime>2015-02-08T20:34:00-04:00</xhb:CreationDateTime>
  <xha:BusinessScope> 5
    <xha:BusinessScopeCriterion>
      <xhb:BusinessScopeCriterionTypeCode
        >test</xhb:BusinessScopeCriterionTypeCode>
      <xhb:BusinessScopeCriterionValue>123</xhb:BusinessScopeCriterionValue>
    </xha:BusinessScopeCriterion> 10
    <xha:ExternalReference>
      <xhb:ID>http://www.company.com</xhb:ID>
    </xha:ExternalReference>
  </xha:BusinessScope>
  <xha:FromParty> 15
    <xha:PartyIdentification>
      <xhb:ID>A</xhb:ID>
    </xha:PartyIdentification>
  </xha:FromParty>
  <xha:ToParty> 20
    <xha:PartyIdentification>
      <xhb:ID>B</xhb:ID>
    </xha:PartyIdentification>
  </xha:ToParty>
</xha:Header> 25
<xha:Payloads>
  <xha:Payload>
    <xhb:InstanceEncryptionIndicator>>false</xhb:InstanceEncryptionIndicator>
    <xha:PayloadContent>
      <myDocumentHere> 30
        <myElement>My Content</myElement>
        <myElement>My Content</myElement>
        <myElement>My Content</myElement>
      </myDocumentHere>
    </xha:PayloadContent> 35
  </xha:Payload>
  <xha:Payload>
    <xhb:ContentTypeCode>text/plain</xhb:ContentTypeCode>
    <xhb:InstanceEncryptionIndicator>>false</xhb:InstanceEncryptionIndicator>
    <xha:PayloadContent> 40
Non-XML payload here, with sensitive characters
escaped such as &amp;, &lt; and &gt;.

Any text, provided it has been escaped, can be included in a payload.
  </xha:PayloadContent> 45
</xha:Payload>
<xha:Payload>
  <xhb:InstanceEncryptionIndicator>>false</xhb:InstanceEncryptionIndicator>
  <xha:PayloadContent>
    <myOtherDocumentHere> 50
      <myOtherElement>My Content</myOtherElement>
      <myOtherElement>My Content</myOtherElement>
      <myOtherElement>My Content</myOtherElement>
    </myOtherDocumentHere>
  </xha:PayloadContent> 55
</xha:Payload>
</xha:Payloads>
</XHE>

```

— [simpleExampleFailSyntax.xml](#)

- an envelope document with an XML well-formedness error (the end tag for the creation date and time is missing the closing right-angle bracket)
- [simpleExampleFailModel.xml](#)
 - an envelope document with an XML validity error (a misspelled element for the creation date and time) 5
- [simpleExampleExtension.xml](#)
 - a simple envelope with a user-defined extension adding information to the envelope
- [simpleExampleTyped.xml](#)
 - a simple envelope with a user-defined extension adding information to the envelope

To invoke the schemas with the demonstration instances, navigate to the directory and invoke the test script: 10

- in Windows:

```
test.bat
```

- in shell:

```
sh test.sh
```

15

The result on the screen should appear as follows:

```
val $ sh test.sh
```

```
#####
Validating simpleExample.xml
```

20

```
#####
===== Phase 1: XSD schema validation =====
No schema validation errors.
===== Phase 2: XSLT code list validation =====
No code list validation errors.
```

25

```
#####
Validating simpleExampleTyped.xml
```

30

```
#####
===== Phase 1: XSD schema validation =====
No schema validation errors.
===== Phase 2: XSLT code list validation =====
No code list validation errors.
```

35

```
#####
Validating simpleExampleFailSyntax.xml
```

```
#####
===== Phase 1: XSD schema validation =====
org.xml.sax.SAXParseException; systemId: file:///Users/admin/t/
artefacts-xhe-v1.0-csd01wd04-test/val/simpleExampleFailSyntax.xml;
lineNumber: 10; columnNumber: 5; The end-tag for element type
"xbh:CreationDateTime" must end with a '>' delimiter.
```

40

```
at org.apache.xerces.parsers.AbstractSAXParser.parse(Unknown Source)
at org.apache.xerces.jaxp.SAXParserImpl$JAXPSAXParser.parse(Unknown Source)
at org.apache.xerces.jaxp.SAXParserImpl.parse(Unknown Source)
at javax.xml.parsers.SAXParser.parse(SAXParser.java:274)
```

45

```

    at com.nwalsh.parsers.XJParser.xsdParse(Unknown Source)
    at com.nwalsh.parsers.XJParser.parse(Unknown Source)
    at com.nwalsh.parsers.XJParse.run(Unknown Source)
    at com.nwalsh.parsers.XJParse.main(Unknown Source)
Exception in thread "main" java.lang.NullPointerException                    5
    at com.nwalsh.parsers.XJParser.printParseStats(Unknown Source)
    at com.nwalsh.parsers.XJParse.run(Unknown Source)
    at com.nwalsh.parsers.XJParse.main(Unknown Source)
Attempting well-formed, namespace-aware parse
Fatal error:file:///Users/admin/t/artefacts-xhe-v1.0-csd01wd04-test/      10
val/simpleExampleFailSyntax.xml:10:5:The end-tag for element type
"xhb:CreationDateTime" must end with a '>' delimiter.

#####
Validating simpleExampleFailModel.xml                                    15
#####
===== Phase 1: XSD schema validation =====
Attempting well-formed, namespace-aware parse
Error:file:///Users/admin/t/artefacts-xhe-v1.0-csd01wd04-test/val/      20
simpleExampleFailModel.xml:9:29:cvc-complex-type.2.4.a: Invalid content
was found starting with element 'xhb:CreationDateTimeXX'. One of
'{"oasis-cefact-xhe-1.0-BasicComponents":CreationDateTime}' is expected.
Parse succeeded (0.7) with 1 error and no warnings.

#####
Validating simpleExampleExtension.xml                                    25
#####
===== Phase 1: XSD schema validation =====
No schema validation errors.
===== Phase 2: XSLT code list validation =====
No code list validation errors.                                       30

val $

```

The test script invokes the validation script using the following::

- in Windows: 35

```

validate.bat schema-file instance-file

```
- in shell:

```

sh validate.sh schema-file instance-file

```

The validation script invokes the schema script using the following:

- in Windows: 40

```

w3cschema.bat schema-file instance-file

```
- in shell:

```

sh w3cschema.sh schema-file instance-file

```

The validation script invokes the XSLT script using the following:

- in Windows: 45

```

xslt.bat instance-file stylesheet-file output-file

```

- in shell:

```
sh xslt.sh instance-file stylesheet-file output-file
```

The empty stylesheet [XHE-DefaultDTQ-1.0.xsl](#) is a placebo that would be replaced with an XSLT stylesheet imposing value validation constraint checking on a given instance of an exchange header envelope. An example of such data type qualification checking would be for code list 5 enumerations.

Components of two freely available software distributions were used to create the tools in the `val` directory. Sources are given below so that these components can be updated as later releases become available.

- [xjparse.jar](#) (renamed from `xjparse-2.0.1.jar`) and the files in the “`val/lib`” directory *10* are from the Xjparse 2.0.1 distribution at

<http://xjparse.org>

- [saxon.jar](#) is from the Saxon 6.5.5 distribution at

<http://prdownloads.sourceforge.net/saxon/saxon6-5-5.zip>

Annex D (informative)

Encrypting payloads to multiple recipients

The XHE supports sending business documents to multiple recipients using a single envelope, which is obtained by adding multiple instances of the ToParty element to the XHE envelope. 5

When encrypting payloads of envelopes with multiple recipients, users SHOULD make use of encryption technologies that support multiple recipients so that an encrypted payload to multiple recipients can be contained in a single instance of an XHE envelope's PayloadContent. Examples of encryption technologies supporting multiple recipients are [\[CMS\]](#), [\[PGP\]](#) and [\[XMLEnc\]](#).

The workings of individual encryption technologies and methodologies are beyond the scope of 10 this specification.

Annex E (informative)

Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

5

Kenneth Bengtsson (co-chair)

Michel Bormans

Gait Boxman

Anders Grangard (co-chair)

G. Ken Holman (editor)

Annex F (informative)

Temporary Annex - Change Log

NOTE This temporary appendix will be removed in the final version of the committee specification.

Revision	Date	Editor	Changes made
csprd01	26 February 2018	GKH	OASIS cover page changes, namespace changes to prevent violating guidelines
csprd01	22 February 2018	GKH	Prepared second edition of public review revision dated February 28, 2018 based on feedback in editing group
csprd01	20 February 2018	GKH	Prepared first public review revision dated February 22, 2018
csd01wd07	08 February 2018	GKH	Comments from February 8 teleconference; putative first public review content; added UUID; documented namespace URI strings in the body of the document
csd01wd06	06 February 2018	GKH	Comments from February 1 teleconference; repaired bibliographic references; added class diagram, added introductory text
csd01wd05	15 December 2017	GKH	Comments from December 14 teleconference
csd01wd04	19 November 2017	GKH	First draft of possibly the complete content
csd01wd03	09 November 2017	GKH	Updates after teleconference of 2017-11-07
csd01wd02	06 November 2017	GKH	Updates after teleconference of 2017-11-02