Activity Report 10, XML Project

2006-02-24
XML Project

by Yushi KOMACHI

In the last Plenary, XML Project reported:

> Considering the comment of project editors, the XML project is developing a simplified XML structure which could be appropriate for project editors' works and should be distributed as an Amendment to ISO/IEC TR 9573-11 2nd edition.

XML Project created a working draft for the Amendment 1, which adds Annex H to ISO/IEC TR 9573-11 2nd edition. The working draft consists of the following clauses:

> Annex H Authoring structure and style specification
>
> > H.1 Introduction
> > H.2 Structure in RELAX NG(ISO/IEC 19757-2)
> > H.3 XSLT specification for preview by HTML browser
> > H.4 XSL-FO specification
> > H.5 Structure conversion for publication using 9573-11 2nd.ed

The working draft was proposed to JTC1/SC34/WG1 meeting and JTC1/SC34 Plenary meeting held in Nov. 2005. Reviewing the working draft, SC34 approved the subdivision of the project of 9573-11 to develop the Amendment 1.

The modification of programme of work was reported to JTC1 in JTC1 N8052.

JTC1 N8052 and the working draft for the Amendment 1 to ISO/IEC TR 9573-11 2nd edition are attached.

TR 9573:1988 (type 3) (CS#17319) *Project ID:* tr9573	WG 1	Publ: 1988		Information processing -- SGML support facilities -- Techniques for using SGML		Dr. Yushi Komachi
TR 9573-11:2004 (type 3) 2nd edition (CS#38368) *Project ID:* tr9573-11-2ed	WG 1	Publ: 2004		Information processing -- SGML support facilities -- Techniques for using SGML -- Part 11: Structure Descriptions and Style Specifications for Standards Document Interchange		Dr. Yushi Komachi
WD TR 9573-11/Amd.1 (No CS #) *Project ID:* tr9573-11-a1	WG 1	10.99:2005-11 Next: 2006-06 FDIS: 2006-12	POW	Information processing - SGML support facilities - Part 11: Structure descriptions and style specifications for standards document interchange Amendment 1: Authoring structure and style specification	Started by project subdivision and added to PoW at 2005-11 plenary session by unanimous resolution of WG1 recommendation.	Dr. Yushi Komachi Mr. MURATA Makoto [FAMILY Given]
TR 9573-13:1991 (type 3) (CS#17332) *Project ID:* tr9573-13-1991	WG 1	Publ: 1991		Information technology -- SGML support facilities -- Techniques for using SGML -- Part 13: Public entity sets for SGML -- for mathematics and science		Mr. Anders Berglund
PDTR 9573-13:2004 (type 3) 2nd Edition (CS#39306) *Project ID:* tr9573-13-2004	WG 1	30.99:2005-05 Next: 2005-11 FDIS: 2005-11	POW 646 629 622 601 599 599b 583r1 456 455 433 432	Information technology -- SGML support facilities -- Techniques for using SGML -- Part 13: Public entity sets for SGML -- for mathematics and science	Document N 387 - Community contribution: problems with ISO entity sets (ISO 8879 and 9573-13) circulated 2003-04-04. Revised CD in preparation.	Dr. David P. Carlisle

WD: ISO/IEC TR 9573-11:2004/Amd.1:2005

Information processing — SGML support facilities — Part 11: Structure descriptions and style specifications for standards document interchange

AMENDMENT 1: Authoring structure and style specification

Traitement de l'information — Facilités de support pour SGML — Partie 11: Descriptions de structure et spécifications de style pour échange de document de normes

AMENDEMENT 11: Titre de l'amendement

Cover page and page ii will be created by ISO/CS.

Comments and editing advice given in green italic type.

Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 9573-11:2004 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology, Subcommittee SC 34, Document description and processing languages*.

Introduction

This Amendment adds the Annex H that specifies a simplified structure description and style specification for draft standards document interchange.

© ISO/IEC 2004 — All rights reserved

Information processing — SGML support facilities — Part 11: Structure descriptions and style specifications for standards document

AMENDMENT 1: Authoring structure and style specification

Add the following Annex H after Annex G:

Annex H (informative) Authoring structure and style specification

H.1 Introduction

ISO/IEC TR 9573-11 2nd edition "Structure descriptions and style specifications for standards document interchange" was published in 2004-04, responding to the user requirements for the interchange of standards documents in an XML environment.

The 2nd edition was developed JTC1/SC34/WG1 coordinating with ITSIG (Information Technology Strategies Implementation Group) XML project, and being based on the SGML "ITSIG exchange DTD". It includes both the SGML and the XML DTDs for standards documents in a publication stage and style specifications described by DSSSL and XSL conforming to ISO/IEC directives.

The structure of standards documents in a publication stage consists of a number of elements precisely defined by ISO/IEC directives. All the elements are not used for description of standards documents in drafting or authoring stage such as WD, CD and etc. For example, attribute gtext is not required in the drafting or authoring stage.

A simplified structure can be used for describing standards documents in drafting or authoring stage. Those standards documents are requested to interchange as well. The simplified structure, therefore, should also be standardized.

This amendment 1 to ISO/IEC TR 9573-11:2004 (2nd edition) specifies the simplified structure of standards documents in drafting or authoring stage and the style specification for rendering the structure.

H.2 Structure in RELAX NG(ISO/IEC 19757-2)

The structure of standards documents in drafting or authoring stage is defined by the following schema file: stdex.rnc described by the compact syntax of ISO/IEC 19757-2.

Major tags of the structure correspond to the tags in ISO/IEC TR 9573-11:2004 as shown in Table H1. Elements that require particular rendering, such as formula, dformula, mathart, inf and fence, are dealt with in artwork. The structure of this Annex include an additional element tabular for aligning some elements in a line.

Table H1 Major tags in Annex H in comparison with tags in ISO/IEC TR 9573-11:2004

Tags in ISO/IEC TR 9573-11:2004	Tags in Annex H
standard	document
profile	head

tpage	
toc	
lpage	
foreword	foreword
intro	introduction
body	
scope	scope
refs	normative-references
defs	terms-and-definitions
ht	title
warning	warning
note	note
xmp	example
fn	
h1,h2,h3,h4,h5,h6,p2,p3,p4,p5,p6	titled-clause
p	p
ul,ol,li,lp,dl,dlentry,dt,dd	ul,ol,li,notation-list,notation-item
rl,rlentry,bl,blentry	referenced-document
annexi,annexn	annex
annexbl	bibliography
xref	xref,Xref,this,This
hp	
superscript,subscript	sup, sub
table	table
figure	figure
artwork	artwork
formula,dformula,mathart,inf,fence	

FILE [stdex.rnc]: --

```
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"

start = document
document = element document { head, body }
head =
  element head {
    element organization { "ISO" | "ISO/IEC" }    # sourceod
    & element document-type {        # type
        "International Standard"
        | "Technical Report"
        | "Guide"
        | "Publicly Available Specification"
        | "Technical Specification"
        | "International Standardized Profile"
      }
    & element stage {            # status
        "committee" | "enquiry" | "approval" | "publication"
      }
    & element tc-number { positive-integer }
    & element tc-name { text }
    & (element sc-number { positive-integer }
       & element sc-name { text })?
    & element wg-number { positive-integer }?     # wdnumber
    & element serial-number { positive-integer }?
    & element draft-number { positive-integer }?
    & element document-number { positive-integer }   # number
    & element part-number { positive-integer }?      # part
    & element document-language { "E" | "F" }     # language
    & element secretariat { "ANSI" }
    & element date { xsd:date }
    & element title {
   attribute langcode { "E" | "F" }?,      # for multilingual title
        element introductory { text }?,
        element main { text },
        element complementary { text }?
      }+
  }

body =
  foreword,                # foreword
  introduction?,           # intro
  warning?,                # warning(!)
  scope,               # scope
  conf?,               # conf(!)
  normative-references?,         # refs
  terms-and-definitions?,        # defs
  titled-clause+,             # h1
  annex*,              # annexn or annexi
  bibliography?,           # annexbl
  index*               # index(!)

foreword = element foreword { block*, part-list?, block* }

part-list =
  element part-list {
    element part {
      element number { positive-integer },
      element title { text }
    }+
  }

title = element title { title-content }

title-content =
  (text
   | element code { title-content })*

clause-content =
  titled-clause+ | untitled-clause-content

untitled-clause-content = untitled-clause+ | block+
introduction = element introduction { block+ }

warning = element warning { block+ }

scope = element scope { block+ }

conf =  element conf { titled-clause }

normative-references =
  element normative-references { block*, referenced-document*, block* }

referenced-document =
  element referenced-document {
    id,
    element abbrev { text },
    element title { text },
    element field { text }*,
    element url { xsd:anyURI }
  }
# boilerplate generated automatically
# need markup for supplementary sources of terms and definitions

terms-and-definitions =
  element terms-and-definitions { terms-and-definitions-content }

terms-and-definitions-content =
  (term-and-definition
   | element clause { title, terms-and-definitions-content })+

term-and-definition =
  element term-and-definition {
    element term { text },
    element term {
      attribute status { text }?,
      text
    }*,
    element definition {
      attribute subject-field { text }?,
      inline
    },
    (example | note | warn )*
  }+

titled-clause = element clause { id, title, clause-content }

untitled-clause = element clause { id, untitled-clause-content }

annex =
  element annex {
    id,
    attribute normative { "true" | "false" },
    title,
    clause-content
  }

bibliography = element bibliography { referenced-document+ }
```

© ISO/IEC 2004 — All rights reserved

```
block = p | ol | ul | notation-list
  | example | note | warn
  | pre | float

p = element p { p.content+ }

p.content =
    inline
  | ol | ul | example
  | pre | float

ol =
  element ol {
    element li { id, block+ }+
  }

ul =
  element ul {
    element li { block+ }+
  }

notation-list =
  element notation-list {
    element notation-item {
      element notation { inline },
      element notation-definition { p, (p | note)* }
    }+
  }

example = element example { p+ }

note = element note { p+ }

warn =
  element warn { attlist.warn, p* }

attlist.warn &=
  id,
  attribute format { "genwarn" | "warning" | "caution" | "remark" }?

pre = element pre { pre-content }

pre-content =
  (text
   | element var { pre-content })*

inline =
  (text
   | element code | b | i | u | var { inline }
   | ref
   | firstterm
   | strong
   | artwork
   | sub | sup
   | footnote
   | tabular
   )*
```

```
ref = this | xref

# "This part of ISO/IEC"
# "This International Standard"
this = element This | this { empty }

xref =
  element Xref {
    attribute to { xsd:IDREF }
  }
  | element xref {
      attribute to { xsd:IDREF }
    }
id = attribute id { xsd:ID }?

firstterm = element firstterm { text }

strong = element strong { text }

index = element index { attlist.index, text }

attlist.index &=
  id,
  attribute status { "informative" }?

footnote = element footnote { id, (text | p)+ }

sub  = element sub { text }

sup  = element sub { text }

float = table | figure

figure = element figure { id, attribute pgwide { yesorno }?, title?, block+ }

table = element table { id, attribute pgwide { yesorno }?, title?, block+ }

tabular = element tabular { attrlist.tabular, tgroup }

attrlist.tabular &=
  attribute frame {
    "all" | "top" | "bottom" | "topbot" | "sides" | "none"
  }?,
  attribute colsep { yesorno }?,
  attribute rowsep { yesorno }?

tgroup =
  element tgroup {
    attlist.tgroup,
    colspec*,
    spanspec*,
    thead?,
    tfoot?,
    tbody
  }
```

© ISO/IEC 2004 — All rights reserved

```
attlist.tgroup &=
  attribute cols { xsd:NMTOKEN },
  attribute tgroupstyle { xsd:NMTOKEN }?,
  attribute colsep { yesorno }?,
  attribute rowsep { yesorno }?,
  [ a:defaultValue = "left" ]
  attribute align { "left" | "right" | "center" | "justify" }?

colspec = element colspec { attlist.colspec }

attlist.colspec &=
  attribute colnum { xsd:NMTOKEN }?,
  attribute colname { xsd:NMTOKEN }?,
  attribute align { "left" | "right" | "center" | "justify" }?,
  attribute colsep { yesorno }?,
  attribute rowsep { yesorno }?,
  attribute colwidth { xsd:string }?

spanspec = element spanspec { attlist.spanspec }

attlist.spanspec &=
  attribute namest { xsd:NMTOKEN },
  attribute nameend { xsd:NMTOKEN },
  attribute spanname { xsd:NMTOKEN },
  [ a:defaultValue = "center" ]
  attribute align { "left" | "right" | "center" | "justify" }?,
  attribute colsep { yesorno }?,
  attribute rowsep { yesorno }?

thead = element thead { attlist.thead, colspec*, row+ }

tfoot = element tfoot { attlist.tfoot, colspec*, row+ }

attlist.thead &=
  [ a:defaultValue = "bottom" ]
  attribute valign { "top" | "middle" | "bottom" }?

attlist.tfoot &=
  [ a:defaultValue = "top" ]
  attribute valign { "top" | "middle" | "bottom" }?

tbody = element tbody { attlist.tbody, row+ }

attlist.tbody &=
  [ a:defaultValue = "top" ]
  attribute valign { "top" | "middle" | "bottom" }?

row = element row { attlist.row, entry+ }

attlist.row &=
  attribute rowsep { yesorno }?,
  attribute valign { "top" | "bottom" | "middle" }?
entry = element entry { attlist.entry, inline* }

attlist.entry &=
  attribute colname { xsd:NMTOKEN }?,
  attribute namest { xsd:NMTOKEN }?,
  attribute nameend { xsd:NMTOKEN }?,
  attribute spanname { xsd:NMTOKEN }?,
  [ a:defaultValue = "0" ]
  attribute morerows { xsd:NMTOKEN }?,
  attribute colsep { yesorno }?,
  attribute rowsep { yesorno }?,
  attribute rotate { xsd:NMTOKEN }?,
  attribute valign { "top" | "bottom" | "middle" }?,
  attribute align { "left" | "right" | "center" | "justify" }?

artwork = element artwork { attlist.artwork }

attlist.artwork &=
  id,
  attribute entity { xsd:ENTITY },
  attribute align { xsd:NMTOKEN }?,
  attribute angle { text }?,
  attribute bloffset { text }?,
  attribute cropped { xsd:NMTOKEN }?,
  attribute float { xsd:NMTOKEN }?,
  attribute height { text }?,
  attribute nsoffset { text }?,
  attribute position { xsd:NMTOKEN }?,
  attribute width { text }?,
  attribute dpi { xsd:NMTOKEN }?,
  attribute impang { text }?,
  attribute impby { "ref" | "copy" }?,
  attribute impsize { text }?,
  attribute sideways { xsd:NMTOKEN }?,
  attribute xoffset { text }?,
  attribute yoffset { text }?

yesorno = "yes" | "no"

positive-integer = xsd:token { pattern = "[1-9][0-9]*|0" }
```

H.3 XSLT specification for preview by HTML browser

XML texts with the structure defined in H.2 can be previewed by an HTML browser, using the conversion defined by the following XSLT specifications. The XSLT specifications consist of the five files: stdex.xsl, stdex_dront.xsl, stdex_toc.xsl, stdex_back.xsl and stdex.css.

FILE [stdex.xsl]: --

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:src="http://purl.oclc.org/dsdl/9573-11amd1/ns/structure/1.0"
  xmlns="http://www.w3.org/1999/xhtml">
  <xsl:import href="stdex_front.xsl" />
  <xsl:import href="stdex_toc.xsl" />
  <xsl:import href="stdex_back.xsl" />
  <xsl:output method="html" doctype-public="-//W3C//DTD HTML 4.0//EN" encoding="UTF-8" />
  <!-- XSLT stylesheet for ISO/IEC stdex.rnc -->
  <!-- version.0.51 -->
  <xsl:template match="/">
    <!-- contents: { document } -->
```

© ISO/IEC 2004 — All rights reserved

```
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="/src:document">
  <!-- contents: { head, body } -->
  <html>
    <head>
      <title>ISO/IEC IT --- Ver.0.50 ---</title>
    </head>
    <link rel="stylesheet" href="stdex.css" type="text/css"  />
    <body>
      <xsl:call-template name="frontm" />
      <hr />
      <xsl:call-template name="toc" />
      <hr />
      <xsl:apply-templates />
      <xsl:call-template name="backm" />
    </body>
  </html>
</xsl:template>
<xsl:template match="src:head">
  <!-- contents: { organization & ... } -->
</xsl:template>
<xsl:template match="src:foreword">
  <!-- contents: { block*, part-list? } -->
  <xsl:variable name="text">
    <xsl:choose>
      <xsl:when test="/src:document/src:head/src:document-language='E'">
        <xsl:text>Foreword</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>Avant-propos</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <h1>
    <xsl:call-template name="anchor-text">
      <xsl:with-param name="anc">foreword</xsl:with-param>
      <xsl:with-param name="text" select="$text"/>
    </xsl:call-template>
  </h1>
  <xsl:apply-templates />
  <hr />
</xsl:template>
<xsl:template match="src:part-list">
  <!-- contents: { part+ } -->
  <ul>
    <xsl:apply-templates />
  </ul>
</xsl:template>
<xsl:template match="src:part">
  <!-- contents: { number, title } -->
  <li>
    <span class="italic">
      <xsl:text>-&#160;Part&#160;</xsl:text>
      <xsl:apply-templates select="src:number"/>
      <xsl:text>:&#160;</xsl:text>
      <xsl:apply-templates select="src:title"/>
    </span>
  </li>
</xsl:template>
<xsl:template match="src:part/src:number">
  <!-- contents: { positive-integer } -->
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:part/src:title">
  <!-- contents: { text } -->
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:introduction">
  <!-- contents: { block } -->
  <h1>
    <xsl:call-template name="anchor-text">
      <xsl:with-param name="anc">introduction</xsl:with-param>
      <xsl:with-param name="text">Introduction</xsl:with-param>
    </xsl:call-template>
  </h1>
  <xsl:apply-templates />
  <hr />
</xsl:template>
<xsl:template name="anchor-text">
  <xsl:param name="anc" />
  <xsl:param name="text" />
  <span id="{$anc}">
    <xsl:value-of select="$text" />
  </span>
</xsl:template>
<xsl:template match="src:warning">
  <!-- contents: { attlist.warning, p* } -->
  <xsl:text>WARNING-</xsl:text>
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:scope">
  <!-- contents: { block } -->
  <xsl:variable name="nth">
    <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-
definitions|src:clause" />
  </xsl:variable>
  <xsl:variable name="text">
    <xsl:choose>
      <xsl:when test="/src:document/src:head/src:document-language='E'">
        <xsl:text>Scope</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>Domaine d'application</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <h1>
    <xsl:call-template name="anchor-text">
      <xsl:with-param name="anc" select="$nth" />
      <xsl:with-param name="text" select="concat($nth, ' ', $text)" />
    </xsl:call-template>
  </h1>
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:conf">
  <!-- contents: { titled-clause } -->
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:normative-references">
  <!-- contents: { block*, referenced-document+ } -->
  <xsl:variable name="nth">
    <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-definitions|src:clause" />
  </xsl:variable>
  <xsl:variable name="text">
    <xsl:choose>
      <xsl:when test="/src:document/src:head/src:document-language='E'">
        <xsl:text>Normative references</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>R&#233;f&#233;rences normatives</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <h1>
    <xsl:call-template name="anchor-text">
      <xsl:with-param name="anc" select="$nth" />
      <xsl:with-param name="text" select="concat($nth, ' ', $text)" />
    </xsl:call-template>
  </h1>
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:referenced-document">
  <!-- contents: { id, abbrev, title, field*, url } -->
  <dl>
    <dd>
      <xsl:apply-templates />
    </dd>
  </dl>
</xsl:template>
<xsl:template match="src:abbrev">
  <xsl:apply-templates />
  <xsl:text>, </xsl:text>
</xsl:template>
<xsl:template match="src:referenced-document/src:title">
  <span class="italic">
    <xsl:apply-templates />
  </span>
</xsl:template>
<xsl:template match="src:terms-and-definitions">
  <!-- contents: { terms-and-definitions-content } -->
  <xsl:variable name="nth">
    <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-definitions|src:clause" />
  </xsl:variable>
  <xsl:variable name="text">
    <xsl:choose>
      <xsl:when test="/src:document/src:head/src:document-language='E'">
        <xsl:text>Terms and definitions</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>Termes, d&#233;finitions et symboles</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <h1>
    <xsl:call-template name="anchor-text">
      <xsl:with-param name="anc" select="$nth" />
      <xsl:with-param name="text" select="concat($nth, ' ', $text)" />
    </xsl:call-template>
  </h1>
  <dl>
    <xsl:apply-templates />
  </dl>
</xsl:template>
<xsl:template match="src:term-and-definition">
  <!-- contents: { term, term*, definition, (example, note, warning)* } -->
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:term-and-definition/src:term">
  <!-- contents: { text } -->
  <dt>
    <xsl:apply-templates />
  </dt>
</xsl:template>
<xsl:template match="src:term-and-definition/src:definition">
  <!-- contents: { inline } -->
  <dd>
    <xsl:apply-templates />
  </dd>
</xsl:template>
<xsl:template match="src:clause">
  <!-- contents: { id, title, clause-content } -->
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:clause/src:title">
  <!-- contents: { text | code } -->
  <xsl:variable name="level" select="count(ancestor::src:clause|ancestor::src:annex)"  />
  <xsl:variable name="nth">
    <xsl:choose>
      <xsl:when test="ancestor::src:annex">
        <xsl:number level="multiple" count="src:annex|src:annex//src:clause" format="A.1" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-definitions|src:clause" format="1.1" />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <xsl:element name="h{$level}">
    <xsl:call-template name="anchor-text">
      <xsl:with-param name="anc" select="$nth" />
      <xsl:with-param name="text" select="concat($nth, ' ', .)" />
    </xsl:call-template>
  </xsl:element>
</xsl:template>
<xsl:template match="src:annex">
  <!-- contents: { id, @normative, title, clause-content } -->
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="src:annex/src:title">
  <!-- contents: { text | code } -->
  <xsl:variable name="nth">
    <xsl:number level="multiple" count="src:annex" format="A" />
  </xsl:variable>
  <h1 class="annex">
    <xsl:text>Annex </xsl:text>
    <xsl:value-of select="$nth" />
    <br />
    <xsl:choose>
      <xsl:when test="../@normative = 'true'">
        <xsl:text>(normative)</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>(informative)</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
    <br />
    <xsl:call-template name="anchor-text">
      <xsl:with-param name="anc" select="$nth" />
      <xsl:with-param name="text" select="." />
    </xsl:call-template>
  </h1>
</xsl:template>
<xsl:template match="src:bibliography">
  <!-- contents: { referenced-document+ } -->
  <xsl:variable name="text">
    <xsl:choose>
      <xsl:when test="/src:document/src:head/src:document-language='E'">
        <xsl:text>Bibliography</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>Bibliographie</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <h1>
    <xsl:call-template name="anchor-text">
      <xsl:with-param name="anc">bibliography</xsl:with-param>
      <xsl:with-param name="text" select="$text"/>
    </xsl:call-template>
  </h1>
  <ol>
    <xsl:apply-templates />
  </ol>
</xsl:template>
<xsl:template match="src:bibliography/src:referenced-document">
  <li>
    <xsl:number format="[1] " />
    <xsl:apply-templates />
  </li>
</xsl:template>
<xsl:template match="src:index">
  <!-- contents: { attlist.index, text } -->
```

```
    <h1>
      <xsl:call-template name="anchor-text">
        <xsl:with-param name="anc">index</xsl:with-param>
        <xsl:with-param name="text">Index</xsl:with-param>
      </xsl:call-template>
    </h1>
    <xsl:apply-templates />
  </xsl:template>
  <xsl:template match="src:p">
    <!-- contents: { inline } -->
    <p>
      <xsl:apply-templates />
    </p>
  </xsl:template>
  <xsl:template match="src:ol">
    <!-- contents: { li } -->
    <ol>
      <xsl:apply-templates />
    </ol>
  </xsl:template>
  <xsl:template match="src:ol/src:li">
    <!-- contents: { id, block+ } -->
    <xsl:variable name="level" select="count(ancestor::src:ol) mod 3" />
    <xsl:variable name="format">
      <xsl:choose>
        <xsl:when test="$level=1">
          <xsl:text>a) </xsl:text>
        </xsl:when>
        <xsl:when test="$level=2">
          <xsl:text>1) </xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>i) </xsl:text>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <li>
      <xsl:number format="{$format}" />
      <xsl:apply-templates />
    </li>
  </xsl:template>
  <xsl:template match="src:ul">
    <!-- contents: { li } -->
    <ul>
      <xsl:apply-templates />
    </ul>
  </xsl:template>
  <xsl:template match="src:ul/src:li">
    <!-- contents: { block+ } -->
    <xsl:variable name="level" select="count(ancestor::src:ul)" />
    <xsl:choose>
      <xsl:when test="$level=2">
        <li>&#183; <xsl:apply-templates /></li>
      </xsl:when>
      <xsl:otherwise>
        <li>- <xsl:apply-templates /></li>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
  <xsl:template match="src:notation-list">
    <!-- contents: { notation-item+ } -->
    <dl compact="1">
      <xsl:apply-templates />
    </dl>
  </xsl:template>
  <xsl:template match="src:notation-item">
    <!-- contents: { notation, notation-definition } -->
    <xsl:apply-templates />
  </xsl:template>
  <xsl:template match="src:notation">
    <!-- contents: { inline } -->
    <dt>
      <xsl:apply-templates />
    </dt>
  </xsl:template>
  <xsl:template match="src:notation-definition">
    <!-- contents: { p, (p | note)* } -->
    <dd>
      <xsl:apply-templates />
    </dd>
  </xsl:template>
  <xsl:template match="src:example">
    <!-- contents: { p+ } -->
    <xsl:variable name="text">
      <xsl:choose>
        <xsl:when test="/src:document/src:head/src:document-language='E'">
          <xsl:text>EXAMPLE</xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>EXEMPLE</xsl:text>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <p>
      <xsl:value-of select="$text" />
      <xsl:text> </xsl:text>
      <xsl:if test="count(../src:xmp) &gt; 1">
        <xsl:number level="single" count="src:xmp" format="1 " />
      </xsl:if>
      <xsl:apply-templates />
    </p>
  </xsl:template>
  <xsl:template match="src:note">
    <!-- contents: { p+ } -->
    <p class="note">
      <xsl:text>NOTE&#160;&#160;</xsl:text>
      <xsl:if test="count(../src:note) &gt; 1">
        <xsl:number level="single" count="src:note" format="1 " />
      </xsl:if>
      <xsl:apply-templates />
    </p>
  </xsl:template>
  <xsl:template match="src:code">
    <!-- contents: { text } -->
    <code>
      <xsl:apply-templates />
    </code>
  </xsl:template>
  <xsl:template match="src:b">
    <!-- contents: { text } -->
    <b>
      <xsl:apply-templates />
    </b>
  </xsl:template>
  <xsl:template match="src:i">
    <!-- contents: { text } -->
    <span class="italic">
      <xsl:apply-templates />
    </span>
  </xsl:template>
  <xsl:template match="src:u">
    <!-- contents: { text } -->
    <u>
      <xsl:apply-templates />
    </u>
  </xsl:template>
  <xsl:template match="src:sup">
    <!-- contents: { text } -->
    <sup>
      <xsl:apply-templates />
    </sup>
  </xsl:template>
  <xsl:template match="src:sub">
    <!-- contents: { text } -->
    <sub>
      <xsl:apply-templates />
    </sub>
  </xsl:template>
  <xsl:template match="src:var">
    <!-- contents: { text } -->
    <var>
      <xsl:apply-templates />
    </var>
  </xsl:template>
  <xsl:template match="src:This | src:this">
    <!-- contents: { empty } -->
    <!-- F: la pr&#233;sente Norme internationale -->
    <xsl:choose>
      <xsl:when test="/src:document/src:head/src:part-number">
        <xsl:choose>
          <xsl:when test="/src:document/src:head/src:document-language='E'">
            <xsl:value-of select="local-name()"/>
            <xsl:text> part of </xsl:text>
          </xsl:when>
        </xsl:choose>
        <xsl:value-of select="/src:document/src:head/src:organization"/>
        <xsl:text> </xsl:text>
        <xsl:value-of select="/src:document/src:head/src:document-number"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:choose>
```

```
        <xsl:when test="/src:document/src:head/src:document-language='E'">
          <xsl:value-of select="local-name()"/>
          <xsl:value-of select="/src:document/src:head/src:document-type"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:choose>
            <xsl:when test="local-name()='This'">
              <xsl:text>La</xsl:text>
            </xsl:when>
            <xsl:otherwise>
              <xsl:text>la</xsl:text>
            </xsl:otherwise>
          </xsl:choose>
          <xsl:text> pr&#233;sente </xsl:text>
          <xsl:choose>
            <xsl:when test="/src:document/src:head/src:document-type='International Standard'">
              <xsl:text>Norme internationale</xsl:text>
            </xsl:when>
          </xsl:choose>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
<xsl:template match="src:Xref | src:xref">
  <xsl:variable name="to" select="@to" />
  <xsl:for-each select="//src:*[@id=$to]">
    <xsl:call-template name="xref-one" />
  </xsl:for-each>
</xsl:template>
<xsl:template name="xref-one">
  <xsl:choose>
    <xsl:when test="self::src:figure">
      <xsl:choose>
        <xsl:when test="ancestor::src:annex">
          <xsl:number level="multiple" count="src:annex|src:figure" format="A.1 " />
        </xsl:when>
        <xsl:otherwise>
          <xsl:number level="any" count="src:figure" />
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:when test="self::src:table">
      <xsl:choose>
        <xsl:when test="ancestor::src:annex">
          <xsl:number level="multiple" count="src:annex|src:table" format="A.1 " />
        </xsl:when>
        <xsl:otherwise>
          <xsl:number level="any" count="src:table" />
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:when test="self::src:li">
      <xsl:variable name="level" select="count(ancestor::src:ol) mod 3" />
      <xsl:variable name="format">
        <xsl:choose>
          <xsl:when test="$level=1">
```

```
              <xsl:text>a) </xsl:text>
            </xsl:when>
            <xsl:when test="$level=2">
              <xsl:text>1) </xsl:text>
            </xsl:when>
            <xsl:otherwise>
              <xsl:text>i) </xsl:text>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:variable>
        <xsl:number format="{$format}" />
      </xsl:when>
      <xsl:when test="ancestor::src:clause">
        <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-definitions|src:clause" format="1.1 " />
      </xsl:when>
      <xsl:when test="self::src:referenced-document">
        <xsl:value-of select="src:abbrev" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:number level="multiple" count="src:annex" format="A.1 " />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
  <xsl:template match="src:firstterm">
    <!-- contents: { text } -->
      <xsl:apply-templates />
  </xsl:template>
  <xsl:template match="src:strong">
    <!-- contents: { text } -->
    <strong>
      <xsl:apply-templates />
    </strong>
  </xsl:template>
  <xsl:template match="src:artwork">
    <!-- contents: { attlist.artwork } -->
    <img src="{@entity}.jpg" />
  </xsl:template>
  <xsl:template match="src:footnote">
    <!-- contents: { id, (text | p)+ } -->
    <xsl:element name="a">
      <xsl:attribute name="onClick">
        <xsl:text>alert('</xsl:text>
        <xsl:call-template name="fn-mark" />
        <xsl:apply-templates />
        <xsl:text>')</xsl:text>
      </xsl:attribute>
      <sup>
        <xsl:call-template name="fn-mark" />
      </sup>
    </xsl:element>
  </xsl:template>
  <xsl:template name="fn-mark">
    <xsl:choose>
      <xsl:when test="ancestor::src:figure">
        <xsl:number level="single" count="src:fn" format="a) " />
      </xsl:when>
      <xsl:when test="ancestor::src:table">
        <xsl:number level="single" count="src:fn" format="a) " />
      </xsl:when>
      <xsl:otherwise>
        <xsl:number value="count(preceding::src:fn)-count(preceding::src:fn[ancestor::src:figure or
          ancestor::src:table])+1" format="1) " />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
  <xsl:template match="src:table">
    <!-- contents: { @pgwide, title?, block+ } -->
    <div align="center">
      <xsl:apply-templates />
      <br />
    </div>
  </xsl:template>
  <xsl:template match="src:table/src:title">
    <xsl:variable name="ref">
      <xsl:choose>
        <xsl:when test="ancestor::src:annex">
          <xsl:number level="multiple" count="src:annex|src:table" format="A.1 " />
        </xsl:when>
        <xsl:otherwise>
          <xsl:number level="any" count="src:table" />
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <xsl:variable name="text">
      <xsl:choose>
        <xsl:when test="/src:document/src:head/src:document-language='E'">
          <xsl:text>Table</xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>Tableau</xsl:text>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <!-- caption -->
      <b>
        <span id="tab{$ref}">
          <xsl:value-of select="concat($text, ' ', $ref)" /> &#8211; <xsl:value-of select="." />
        </span>
      </b>
    <!-- /caption -->
  </xsl:template>
  <xsl:template match="src:figure">
    <!-- contents: { @pgwide, title?, block+ } -->
    <div align="center">
      <xsl:apply-templates select="*[not(self::src:title)]"/>
      <xsl:apply-templates select="src:title"/>
    </div>
  </xsl:template>
  <xsl:template match="src:figure/src:title">
    <xsl:variable name="ref">
      <xsl:choose>
        <xsl:when test="ancestor::src:annex">
          <xsl:number level="multiple" count="src:annex|src:figure" format="A.1 " />
        </xsl:when>
        <xsl:otherwise>
          <xsl:number level="any" count="src:figure" />
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <div align="center">
      <b>
        <span id="fig{$ref}">
          <xsl:text>Figure </xsl:text>
          <xsl:value-of select="$ref" /> &#8211; <xsl:value-of select="." />
        </span>
      </b>
    </div>
  </xsl:template>
  <xsl:template match="src:tabular">
    <table>
      <xsl:choose>
        <xsl:when test="@frame=&quot;none&quot;" />
        <xsl:otherwise>
          <xsl:attribute name="border" />
        </xsl:otherwise>
      </xsl:choose>
      <xsl:apply-templates />
    </table>
  </xsl:template>
  <xsl:template match="src:tgroup">
    <xsl:apply-templates select="src:colspec" />
    <xsl:apply-templates select="src:thead" />
    <xsl:apply-templates select="src:tbody" />
    <xsl:apply-templates select="src:tfoot" />
    <tr>
      <td colspan="100">
        <xsl:apply-templates select=".//src:note" />
      </td>
    </tr>
  </xsl:template>
  <xsl:template match="src:colspec" />
  <xsl:template match="src:thead">
    <xsl:apply-templates />
  </xsl:template>
  <xsl:template match="src:tbody">
    <xsl:apply-templates />
  </xsl:template>
  <xsl:template match="src:tfoot">
    <xsl:apply-templates />
  </xsl:template>
  <xsl:template match="src:row">
    <tr>
      <xsl:choose>
        <xsl:when test="parent::src:thead or parent::src:tfoot">
          <xsl:attribute name="bgcolor">#d0d0d0</xsl:attribute>
        </xsl:when>
      </xsl:choose>
      <xsl:choose>
        <xsl:when test="@valign">
          <xsl:attribute name="valign">
            <xsl:value-of select="@valign" />
          </xsl:attribute>
        </xsl:when>
        <xsl:when test="parent::src:*[@valign]">
          <xsl:attribute name="valign">
            <xsl:value-of select="parent::src:*[@valign]/@valign" />
          </xsl:attribute>
        </xsl:when>
        <xsl:when test="parent::src:thead">
          <xsl:attribute name="valign">bottom</xsl:attribute>
        </xsl:when>
        <xsl:otherwise>
          <xsl:attribute name="valign">top</xsl:attribute>
        </xsl:otherwise>
      </xsl:choose>
      <xsl:choose>
        <xsl:when test="@align">
          <xsl:attribute name="align">
            <xsl:value-of select="@align" />
          </xsl:attribute>
        </xsl:when>
        <xsl:when test="ancestor::src:*[@align]">
          <xsl:attribute name="align">
            <xsl:value-of select="ancestor::src:*[@align]/@align" />
          </xsl:attribute>
        </xsl:when>
      </xsl:choose>
      <xsl:apply-templates />
    </tr>
  </xsl:template>
  <xsl:template match="src:entry">
    <xsl:choose>
      <xsl:when test="ancestor::src:thead">
        <xsl:call-template name="ent">
          <xsl:with-param name="tag">th</xsl:with-param>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:call-template name="ent">
          <xsl:with-param name="tag">td</xsl:with-param>
        </xsl:call-template>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
  <xsl:template name="ent">
    <xsl:param name="tag" />
    <xsl:element name="{$tag}">
      <xsl:if test="@valign">
        <xsl:attribute name="valign">
          <xsl:value-of select="@valign" />
        </xsl:attribute>
      </xsl:if>
      <xsl:choose>
        <xsl:when test="@align">
          <xsl:attribute name="align">
            <xsl:value-of select="@align" />
          </xsl:attribute>
        </xsl:when>
        <xsl:when test="ancestor::src:*[@align]">
          <xsl:attribute name="align">
            <xsl:value-of select="ancestor::src:*[@align]/@align" />
          </xsl:attribute>
        </xsl:when>
      </xsl:choose>
      <xsl:if test="@morerows">
        <xsl:attribute name="rowspan">
          <xsl:value-of select="@morerows + 1" />
        </xsl:attribute>
      </xsl:if>
      <xsl:if test="@namest and @nameend">
        <xsl:call-template name="colspan">
          <xsl:with-param name="namest" select="@namest" />
          <xsl:with-param name="nameend" select="@nameend" />
        </xsl:call-template>
      </xsl:if>
      <xsl:apply-templates />
    </xsl:element>
  </xsl:template>
  <xsl:template name="colspan">
    <xsl:param name="namest" />
    <xsl:param name="nameend" />
    <xsl:call-template name="colspan-sub">
      <xsl:with-param name="numst">
        <xsl:choose>
          <xsl:when test="parent::src:row/preceding-sibling::src:colspec[@colname=$namest]">
            <xsl:value-of
              select="count(parent::src:row/preceding-sibling::src:colspec[@colname=$namest]/preceding-
sibling::src:colspec)"
             />
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of
              select="count(ancestor::src:tgroup/child::src:colspec[@colname=$namest]/preceding-
sibling::src:colspec)"
             />
          </xsl:otherwise>
        </xsl:choose>
      </xsl:with-param>
      <xsl:with-param name="numend">
        <xsl:choose>
          <xsl:when test="parent::src:row/preceding-sibling::src:colspec[@colname=$nameend]">
            <xsl:value-of
              select="count(parent::src:row/preceding-sibling::src:colspec[@colname=$nameend]/preceding-
sibling::src:colspec)"
             />
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of
              select="count(ancestor::src:tgroup/child::src:colspec[@colname=$nameend]/preceding-
sibling::src:colspec)"
             />
          </xsl:otherwise>
        </xsl:choose>
      </xsl:with-param>
```

© ISO/IEC 2004 — All rights reserved

```
  </xsl:call-template>
 </xsl:template>
 <xsl:template name="colspan-sub">
  <xsl:param name="numst" />
  <xsl:param name="numend" />
  <xsl:attribute name="colspan">
   <xsl:value-of select="$numend - $numst + 1" />
  </xsl:attribute>
 </xsl:template>

 <xsl:template match="text()">
  <xsl:if test="normalize-space(.)!=' '">
   <xsl:value-of select="." />
  </xsl:if>
 </xsl:template>

</xsl:stylesheet>
```

FILE [stdex_front.xsl]: ---

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns:src="http://purl.oclc.org/dsdl/9573-11amd1/ns/structure/1.0"
 xmlns="http://www.w3.org/1999/xhtml">
 <xsl:template name="frontm">
  <!-- front matter -->
  <xsl:apply-templates select="src:head" mode="frontm" />
 </xsl:template>
 <xsl:template match="src:head" mode="frontm">
  <!-- head for front matter -->
  <xsl:variable name="lang" select="src:document-language/text()" />
  <div class="title">
   <xsl:variable name="title">
    <xsl:choose>
     <xsl:when test="src:document-type='International Standard'">
      <xsl:choose>
       <xsl:when test="src:document-language='E'">
        <xsl:value-of select="src:document-type"/>
       </xsl:when>
       <xsl:otherwise>
        <xsl:text>Norme Internationale</xsl:text>
       </xsl:otherwise>
      </xsl:choose>
     </xsl:when>
    </xsl:choose>
   </xsl:variable>
   <xsl:call-template name="capitalize">
    <xsl:with-param name="str" select="$title" />
   </xsl:call-template>
   <div align="right">
    <xsl:value-of select="src:organization" />
    <xsl:value-of select="src:document-number" />
    <xsl:if test="src:part-number">
     <xsl:text>-</xsl:text>
     <xsl:value-of select="src:part-number" />
    </xsl:if>
```

```
        <br />
        <!-- div class="edition">
          <xsl:value-of select="src:edition" />
          <xsl:text> edition</xsl:text>
        </div *** -->
        <br />
        <div class="date">
          <xsl:value-of select="src:date" />
        </div>
      </div>
      <xsl:apply-templates select="src:title[@langcode=$lang]" mode="frontm" />
    </div>
    <br />
    <div class="subtitle">
      <xsl:apply-templates select="src:title[@langcode!=$lang]" mode="frontm" />
    </div>
    <br />
    <br />
    <div align="right">
      <div class="refnum">
        <xsl:choose>
          <xsl:when test="/src:document/src:head/src:document-language='E'">
            <xsl:text>Reference number</xsl:text>
          </xsl:when>
          <xsl:otherwise>
            <xsl:text>Num&#233;ro de r&#233;f&#233;rence</xsl:text>
          </xsl:otherwise>
        </xsl:choose>
        <br />
        <xsl:call-template name="refnum" />
      </div>
    </div>
  </xsl:template>
  <xsl:template name="capitalize">
    <xsl:param name="str"/>
    <xsl:value-of                    select="translate($str,                    'abcdefghijklmnopqrstuvwxyz',
'ABCDEFGHIJKLMNOPQRSTUVWXYZ')"/>
  </xsl:template>
  <xsl:template match="src:title" mode="frontm">
    <xsl:apply-templates mode="frontm" />
  </xsl:template>
  <xsl:template match="src:introductory" mode="frontm">
    <xsl:apply-templates />
    <br />
  </xsl:template>
  <xsl:template match="src:main" mode="frontm">
    <xsl:apply-templates />
    <br />
  </xsl:template>
  <xsl:template match="src:complementary" mode="frontm">
    <xsl:call-template name="part">
      <xsl:with-param name="lang" select="../@langcode" />
    </xsl:call-template>
    <xsl:value-of select="../../src:part-number" />
    <xsl:text>:</xsl:text>
    <br />
    <xsl:apply-templates />
```

© ISO/IEC 2004 — All rights reserved

```
    <br />
  </xsl:template>
  <xsl:template name="part">
    <xsl:param name="lang" />
    <xsl:choose>
      <xsl:when test="$lang='E'">
        <xsl:text>Part</xsl:text>
      </xsl:when>
      <xsl:when test="$lang='F'">
        <xsl:text>Partie</xsl:text>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
  <xsl:template name="refnum">
    <xsl:value-of select="src:organization" />
    <xsl:value-of select="src:document-number" />
    <xsl:if test="src:part-number">
      <xsl:text>-</xsl:text>
      <xsl:value-of select="src:part-number" />
    </xsl:if>
    <xsl:text>:</xsl:text>
    <xsl:variable name="date" select="src:date" />
    <xsl:value-of select="substring($date, 0, 5)" />
    <xsl:text>(</xsl:text>
    <xsl:value-of select="src:document-language" />
    <xsl:text>)</xsl:text>
  </xsl:template>
</xsl:stylesheet>
```

FILE [stdex_toc.xsl]: --

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:src="http://purl.oclc.org/dsdl/9573-11amd1/ns/structure/1.0"
  xmlns="http://www.w3.org/1999/xhtml">

  <xsl:template name="toc">
    <h1>Contents</h1>
    <ul>
      <xsl:apply-templates
        select="src:foreword|src:introduction|src:scope|src:conf|src:normative-references|src:terms-and-
definitions|//src:clause|src:annex|src:bibliography|src:index" mode="toc" />
    </ul>
  </xsl:template>
  <xsl:template match="src:foreword" mode="toc">
    <xsl:variable name="text">
      <xsl:choose>
        <xsl:when test="/src:document/src:head/src:document-language='E'">
          <xsl:text>Foreword</xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>Avant-propos</xsl:text>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <li>
```

```
      <xsl:call-template name="link-text">
        <xsl:with-param name="anc">foreword</xsl:with-param>
        <xsl:with-param name="text" select="$text"/>
      </xsl:call-template>
    </li>
  </xsl:template>
  <xsl:template match="src:introduction" mode="toc">
    <li>
      <xsl:call-template name="link-text">
        <xsl:with-param name="anc">introduction</xsl:with-param>
        <xsl:with-param name="text">Introduction</xsl:with-param>
      </xsl:call-template>
    </li>
  </xsl:template>
  <xsl:template name="link-text">
    <xsl:param name="anc" />
    <xsl:param name="text" />
    <a href="#{$anc}">
      <xsl:value-of select="$text" />
    </a>
  </xsl:template>
  <xsl:template match="src:scope" mode="toc">
    <xsl:variable name="nth">
      <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-
definitions|src:clause" />
    </xsl:variable>
    <xsl:variable name="text">
      <xsl:choose>
        <xsl:when test="/src:document/src:head/src:document-language='E'">
          <xsl:text>Scope</xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>Domaine d'application</xsl:text>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <li>
      <xsl:call-template name="link-text">
        <xsl:with-param name="anc" select="$nth" />
        <xsl:with-param name="text" select="concat($nth, ' ', $text)" />
      </xsl:call-template>
    </li>
  </xsl:template>
  <xsl:template match="src:conf">
    <!-- contents: { titled-clause } -->
    <xsl:apply-templates />
  </xsl:template>
  <xsl:template match="src:normative-references" mode="toc">
    <xsl:variable name="nth">
      <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-
definitions|src:clause" />
    </xsl:variable>
    <xsl:variable name="text">
      <xsl:choose>
        <xsl:when test="/src:document/src:head/src:document-language='E'">
          <xsl:text>Normative references</xsl:text>
        </xsl:when>
    <xsl:otherwise>
     <xsl:text>R&#233;f&#233;rences normatives</xsl:text>
    </xsl:otherwise>
   </xsl:choose>
  </xsl:variable>
  <li>
   <xsl:call-template name="link-text">
    <xsl:with-param name="anc" select="$nth" />
    <xsl:with-param name="text" select="concat($nth, ' ', $text)" />
   </xsl:call-template>
  </li>
 </xsl:template>
 <xsl:template match="src:terms-and-definitions" mode="toc">
  <xsl:variable name="nth">
   <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-definitions|clause" />
  </xsl:variable>
  <xsl:variable name="text">
   <xsl:choose>
    <xsl:when test="/src:document/src:head/src:document-language='E'">
     <xsl:text>Terms and definitions</xsl:text>
    </xsl:when>
    <xsl:otherwise>
     <xsl:text>Termes, d&#233;finitions et symboles</xsl:text>
    </xsl:otherwise>
   </xsl:choose>
  </xsl:variable>
  <li>
   <xsl:call-template name="link-text">
    <xsl:with-param name="anc" select="$nth" />
    <xsl:with-param name="text" select="concat($nth, ' ', $text)" />
   </xsl:call-template>
  </li>
 </xsl:template>
 <xsl:template match="src:clause" mode="toc">
  <xsl:variable name="nth">
   <xsl:choose>
    <xsl:when test="ancestor::src:annex">
     <xsl:number level="multiple" count="src:annex|src:annex//src:clause" format="A.1" />
    </xsl:when>
    <xsl:otherwise>
     <xsl:number level="multiple" count="src:scope|src:conf|src:normative-references|src:terms-and-definitions|src:clause" format="1.1" />
    </xsl:otherwise>
   </xsl:choose>
  </xsl:variable>
  <li>
   <xsl:call-template name="link-text">
    <xsl:with-param name="anc" select="$nth" />
    <xsl:with-param name="text" select="concat($nth, ' ', src:title)" />
   </xsl:call-template>
  </li>
 </xsl:template>
 <xsl:template match="src:annex" mode="toc">
  <xsl:variable name="nth">
   <xsl:number level="multiple" count="src:annex" format="A" />
  </xsl:variable>
```

```
    <li>
      <xsl:call-template name="link-text">
        <xsl:with-param name="anc" select="$nth" />
        <xsl:with-param name="text" select="concat($nth, ' ', src:title)" />
      </xsl:call-template>
    </li>
  </xsl:template>
  <xsl:template match="src:bibliography" mode="toc">
    <xsl:variable name="text">
      <xsl:choose>
        <xsl:when test="/src:document/src:head/src:document-language='E'">
          <xsl:text>Bibliography</xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>Bibliographie</xsl:text>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <li>
      <xsl:call-template name="link-text">
        <xsl:with-param name="anc">bibliography</xsl:with-param>
        <xsl:with-param name="text" select="$text" />
      </xsl:call-template>
    </li>
  </xsl:template>
  <xsl:template match="src:index" mode="toc">
    <li>
      <xsl:call-template name="link-text">
        <xsl:with-param name="anc">index</xsl:with-param>
        <xsl:with-param name="text">Index</xsl:with-param>
      </xsl:call-template>
    </li>
  </xsl:template>
</xsl:stylesheet>
```

FILE [stdex_back.xsl]: --

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:src="http://purl.oclc.org/dsdl/9573-11amd1/ns/structure/1.0"
  xmlns="http://www.w3.org/1999/xhtml">

  <xsl:template name="backm">
    <!-- back matter -->
    <hr />
    <xsl:apply-templates select="src:head" mode="backm" />
  </xsl:template>
  <xsl:template match="src:head" mode="backm">
    <div class="backm">
      <xsl:text>ICS </xsl:text>
      <!-- *** xsl:for-each select="classifn">
        <xsl:call-template name="listel">
          <xsl:with-param name="delimchar">:</xsl:with-param>
          <xsl:with-param name="termchar" />
        </xsl:call-template>
      </xsl:for-each *** -->
```

© ISO/IEC 2004 — All rights reserved

```
    </div>
    <br />
    <xsl:if test="src:keyword">
      <div class="keyword">
        <b>
          <xsl:text>Descripters: </xsl:text>
        </b>
        <xsl:for-each select="src:keyword">
          <xsl:call-template name="listel">
            <xsl:with-param name="delimchar">, </xsl:with-param>
            <xsl:with-param name="termchar">.</xsl:with-param>
          </xsl:call-template>
        </xsl:for-each>
      </div>
    </xsl:if>
    <hr />
  </xsl:template>
  <xsl:template name="listel">
    <xsl:param name="delimchar" />
    <xsl:param name="termchar" />
    <xsl:apply-templates />
    <xsl:choose>
      <xsl:when test="position()=last()">
        <xsl:value-of select="$termchar" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="$delimchar" />
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

FILE [stdex.css]: --

```
h1  {
    font-size:24pt ;
}
h1.annex  {
    font-size:20pt ;
    text-align:center
}
h2  {
    font-soze:24pt ;
}
h3  {
    font-size:12pt ;
}
h4  {
    font-size:10pt ;
}
h5  {
    font-size:10pt ;
}
h6  {
    font-size:10pt ;
}
strong.blue  {
   color:blue ;
   font-weight:bold
}
dt  {
   font-weight:bold
}
ol  {
   list-style:none outside
}
ul  {
   list-style:none outside
}
a:link  {
   color:blue
}

a:active  {
   color:red
}
a:hover  {
   color:green
}
p  {
   margin-top:0.5em ;
   margin-bottom:1.5em
}
p.note > p:first-child {
   display:inline
}
p.note > p {
   display:block
}
li > p:first-child {
   display:inline
}
item  {
   color:blue ;
   font-weight:bold
}
fn  {
   color:blue ;
   font-weight:bold
}
div.title  {
   font-size:x-large ;
   font-weight:bold
}
div.date  {
   font-size:x-small
}
div.subtitle  {
   font-size:small ;
   font-style:italic
}
div.refnum  {
   font-size:x-small
}
```

© ISO/IEC 2004 — All rights reserved

```
span.italic  {
   font-style:italic
}
```

H.4 XSL-FO specification

TBD

H.5 Structure conversion for publication using 9573-11 2nd.ed

TBD