ISO TC184/SC4/ EC N 47+ and also **TC184/SC4 N987**

Date: May 1996 **Supersedes SC4/ EC N 47**

PRODUCT DATA REPRESENTATION AND EXCHANGE

Part: 3456 **Title:** LaTeX package files for ISO 10303

Purpose of this document as it relates to the target document is:

- X Primary Content
- ___ Issue Discussion
- ___ Alternate Proposal
- ___ Partial Content

Current Status: Working project draft

ABSTRACT:
This document describes the LaTeX style files created for ISO 10303.

KEYWORDS:
LaTeX, Style file

Document Status/Dates (dd/mm/yy)

Part Documents		Other SC4 Documents
	Project Leader	May 1996
	Working Group Convenor	
	Qualification	
	Integration	
	Editing	

Owner/Editor: Peter R Wilson
Address: NIST
Bldg. 220, Room A127
Gaithersburg, MD 20899
USA

Telephone/FAX: +1 (301) 975-2976
E-mail: pwilson@cme.nist.gov

Alternate: Tony Day
Address: Sikorsky Aircraft

Telephone/FAX:
E-mail:

Comments to Reader

This document serves two purposes. Firstly, it provides a description of the current LaTeX style file for ISO 10303. Secondly, the source can be used as an example of using the LaTeX commands.
The document has been updated following ISO editorial comments on the first release of the STEP IS parts.

Contents

Annexes

Tables

Foreword

The International Organization for Standardization (ISO) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

International Standard ISO 10303–3456 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration,* Subcommittee SC4, *Industrial data.* ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration – Product data representation and exchange:*

- Part 1, Overview and fundamental principles;
- Part 11, Description method: EXPRESS language reference manual;
- Part 12, Description method: EXPRESS-I language reference manual;
- Part 13, Description method: Architecture and methodology reference manual;
- Part 21, Implementation method: Clear text encoding of the exchange structure;
- Part 22, Implementation method: Standard data access interface specification;
- Part 23, Implementation method: C++ language binding to the standard data access interface;
- Part 24, Implementation method: C language binding to the standard data access interface;
- Part 26, Implementation method: Interface definition language binding to the standard data access interface;
- Part 31, Conformance testing methodology and framework: General concepts;
- Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;
- Part 33, Conformance testing methodology and framework: Structure and use of abstract test suites;
- Part 34, Conformance testing methodology and framework: Abstract test methods;

- Part 35, Conformance testing methodology and framework: Abstract test methods for SDAI implementations;
- Part 41, Integrated generic resource: Fundamentals of product description and support;
- Part 42, Integrated generic resource: Geometric and topological representation;
- Part 43, Integrated generic resource: Representation structures;
- Part 44, Integrated generic resource: Product structure configuration;
- Part 45, Integrated generic resource: Materials;
- Part 46, Integrated generic resource: Visual presentation;
- Part 47, Integrated generic resource: Shape variation tolerances;
- Part 49, Integrated generic resource: Process structure and properties;
- Part 101, Integrated application resource: Draughting;
- Part 102, Integrated application resource: Ship structures;
- Part 104, Integrated application resource: Finite element analysis;
- Part 105, Integrated application resource: Kinematics;
- Part 106, Integrated application resource: Building construction core model;
- Part 201, Application protocol: Explicit draughting;
- Part 202, Application protocol: Associative draughting;
- Part 203, Application protocol: Configuration controlled design;
- Part 204, Application protocol: Mechanical design using boundary representation;
- Part 205, Application protocol: Mechanical design using surface representation;
- Part 207, Application protocol: Sheet metal die planning and design;
- Part 208, Application protocol: Life cycle management - Change process;
- Part 209, Application protocol: Composite and metallic structural analysis and related design;
- Part 210, Application protocol: Design of layered electronic products;

- Part 211, Application protocol: Electronics test diagnostics and remanufacture;
- Part 212, Application protocol: Electrotechnical design and installation
- Part 213, Application protocol: Numerical control process plans for machined parts;
- Part 214, Application protocol: Core data for automotive design processes;
- Part 215, Application protocol: Ship arrangement;
- Part 216, Application protocol: Ship moulded forms;
- Part 217, Application protocol: Ship piping;
- Part 218, Application protocol: Ship structures;
- Part 220, Application protocol: Process planning, manufacture, and assembly of layered electronic products;
- Part 221, Application protocol: Functional data and their schematic representation for process plant;
- Part 222, Application protocol: Exchange of product data for composite structures;
- Part 223, Application protocol: Exchange of design and manufacturing product information for cast parts;
- Part 224, Application protocol: Mechanical product definition for process plans using mechanical feature;
- Part 225, Application protocol: Building elements using explicit shape representation;
- Part 226, Application protocol: Ship mechanical systems;
- Part 227, Application protocol: Plant spatial configuration;
- Part 228, Application protocol: Building services: Heating, ventilation, and air conditioning;
- Part 229, Application protocol: Exchange of design and manufacturing product information for forged parts;
- Part 230, Application protocol: Building structural frame: Steelwork;
- Part 231, Application protocol: Process engineering data: Process design and process specification of major equipment;

- Part 301, Abstract test suite: Explicit draughting;
- Part 302, Abstract test suite: Associative draughting;
- Part 303, Abstract test suite: Configuration controlled design;
- Part 304, Abstract test suite: Mechanical design using boundary representation;
- Part 305, Abstract test suite: Mechanical design using surface representation;
- Part 307, Abstract test suite: Sheet metal die planning and design;
- Part 308, Abstract test suite: Life cycle management - Change process;
- Part 309, Abstract test suite: Composite and metallic structural analysis and related design;
- Part 310, Abstract test suite: Design of layered electronic products;
- Part 311, Abstract test suite: Electronics test diagnostics and remanufacture;
- Part 312, Abstract test suite: Electrotechnical design and installation;
- Part 313, Abstract test suite: Numerical control process plans for machined parts;
- Part 314, Abstract test suite: Core data for automotive mechanical design processes;
- Part 315, Abstract test suite: Ship arrangement;
- Part 316, Abstract test suite: Ship moulded forms;
- Part 317, Abstract test suite: Ship piping;
- Part 318, Abstract test suite: Ship structures;
- Part 320, Abstract test suite: Process planning, manufacture, and assembly of layered electronic products;
- Part 321, Abstract test suite: Functional data and their schematic representation for process plant;
- Part 322, Abstract test suite: Exchange of product data for composite structures;
- Part 323, Abstract test suite: Exchange of design and manufacturing product information for cast parts;
- Part 324, Abstract test suite: Mechanical product definition for process plans using mechanical features;

- Part 325, Abstract test suite: Building elements using explicit shape representation;
- Part 326, Abstract test suite: Ship mechanical systems;
- Part 327, Abstract test suite: Plant spatial configuration;
- Part 328, Abstract test suite: Building services: Heating, ventilation, and air conditioning;
- Part 329, Abstract test suite: Exchange of design and manufacturing product information for forged parts;
- Part 330, Abstract test suite: Building structural frame: Steelwork;
- Part 331, Abstract test suite: Process engineering data: Process design and process specification of major equipment;
- Part 501, Application interpreted construct: Edge-based wireframe;
- Part 502, Application interpreted construct: Shell-based wireframe;
- Part 503, Application interpreted construct: Geometrically bounded 2D wireframe;
- Part 504, Application interpreted construct: Draughting annotation;
- Part 505, Application interpreted construct: Drawing structure and administration;
- Part 506, Application interpreted construct: Draughting elements;
- Part 507, Application interpreted construct: Geometrically bounded surface;
- Part 508, Application interpreted construct: Non-manifold surface;
- Part 509, Application interpreted construct: Manifold surface;
- Part 510, Application interpreted construct: Geometrically bounded wireframe;
- Part 511, Application interpreted construct: Topologically bounded surface;
- Part 512, Application interpreted construct: Faceted boundary representation;
- Part 513, Application interpreted construct: Elementary boundary representation;
- Part 514, Application interpreted construct: Advanced boundary representation;
- Part 515, Application interpreted construct: Constructive solid geometry;
- Part 516, Application interpreted construct: Mechanical design context;

- Part 517, Application interpreted construct: Mechanical design geometric presentation;
- Part 518, Application interpreted construct: Mechanical design shaded representation.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of this International Standard reflects its structure:

- Parts 11 to 13 specify the description methods,
- Parts 21 to 26 specify the implementation methods,
- Parts 31 to 35 specify the conformance testing methodology and framework,
- Parts 41 to 49 specify the integrated generic resources,
- Parts 101 to 106 specify the integrated application resources,
- Parts 201 to 231 specify the application protocols,
- Parts 301 to 331 specify the abstract test suites, and
- Parts 501 to 518 specify the application interpreted constructs.

Should further parts of ISO 10303 be published, they will follow the same numbering pattern.

Annexes A, B and C are an integral part of this part of ISO 10303. Annexes D, E, F and G are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303–1. This part of ISO 10303 is a member of the descriptive methods series. Major subdivisions of this International Standard are:

– the step facility;

– the ir facility;

– the ap facility;

– the aic facility;

– the atc facility.

This part of ISO 10303 specifies the LaTeX facilities specifically designed for use in preparing the various parts of this standard.

This part of ISO 10303 is based in part upon material in the ISO Directives Part 3 (*Drafting and presentation of International Standard*) which was used as a basis for the overall typographical layout. The LaTeX facilities described are based upon the specifications given in *ISO TC 184/SC4 N-432*.

Overview

This document describes a set of LaTeX files for use within ISO 10303, commonly called STEP. The electronic source of this document also provides an example of the use of these files.

Commands that have been deleted are noted, as are commands that are deprecated and which will be deleted in the next version.

The current set of files have been developed by Peter Wilson from Kent Reed's files. In turn, these were revisions of files originally created by Phil Spiby, based on early work by Phil Kennicott.[1)]

NOTE – It is important to note that the style files should not be used with any version of LaTeX v2.09 that incorporates the New Font Selection Scheme (NFSS).

[1)] In mid 1994 LaTeX was upgraded from version 2.09 to what is called LaTeX2e. The files described in this document should be compatible with both versions of LaTeX.

Documents produced with the LaTeX files have been twice reviewed by the ISO Editorial Board in Geneva for conformance to their typographical and layout requirements. The first review was of a set of Draft International Standard documents. This review resulted in some changes to the style files. The second review was of a set of twelve International Standard documents. Likewise, this review led to changes in the style files to bring the documents into conformance.

With the issuance of the first STEP release, the opportunity has been taken to provide a new baseline release of the package files. In particular, one STEP specific package file is available for all STEP parts, while others contain only commands relevant to the documentation of particular series of parts. The range of package files may be extended in the future to cater for documentation specific to all STEP parts.

This new baseline release has also been designed to cater for the fact that a major update of LaTeX to LaTeX2e took place during 1994. This version of LaTeX is the only officially supported version.

Because ISO standard documents have a very structured layout, the class and package files have been designed to reflect the logical document structure to a much greater extent than the 'standard' LaTeX files. Further, ISO documents are published in more than one language. The files described here are written for the English language, but the language-specific elements have been parameterized for easy modification for publication in other official ISO languages, such as French.

Industrial automation systems and integration — Product data representation and exchange — Part 3456 : Documentation methods: The LaTeX package files reference manual

1 Scope

This part of ISO 10303 describes a set of LaTeX files for typesetting documents according to the ISO Directives Part 3, together with the STEP Supplementary Directives.

The following are within the scope of this part of ISO 10303:

- use of LaTeX for preparing ISO 10303 documents.

The following are outside the scope of this part of ISO 10303:

- use of LaTeX for preparing ISO standard documents in general;
- use of LaTeX in general;
- use of other document preparation systems.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO Directives Part 3, *Drafting and presentation of International Standards.*

ISO TC 184/SC4 N-4432, *Supplementary directives for the drafting and presentation of ISO 10303.*

ISO 10303-1:1994, *Industrial automation ssystems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles.*

ISO 10303-11:1994, *Industrial automation ssystems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.*

ISO 10303-12:—[2], *Industrial automation systems and integration — Product data representation and exchange — Part 12: Description methods: The EXPRESS-I language reference manual.*

ISO/IEC 8824-1:1994, *Information technology — Open systems interconnection — Abstract syntax notation one (ASN.1) — Part 1: Specification of basic notation.*

P. R. WILSON:—[2], *LaTeX for standards: The LaTeX package files user manual.*

3 Definitions

3.1 Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms defined in ISO 10303-1:

– application protocol (AP)

– integrated resource

3.2 Other definitions

For the purposes of this part of ISO 10303, the following definitions apply.

3.2.1 boilerplate: Text whose wording is fixed.

3.2.2 style file: A set of LaTeX macros assembled into a single file.

3.2.3 package file: A style file for use with LaTeX2e.

3.2.4 facility: A generic term for a set of LaTeX macros assembled for a common purpose. It includes both style and package files.

4 Symbols and abbreviations

For the purposes of this part of ISO 10303, the following abbreviations apply.

AIC	Application Interpreted Construct
AP	Application Protocol
DIS	Draft International Standard

[2] To be published.

IS International Standard

ISOD ISO directives Part 3

SD Supplementary Directives — *Supplementary directives for the drafting and presentation of ISO 10303*

IS-REVIEW The ISO Editorial Board review (September 1994) of twelve IS documents for conformance to ISO typographical and layout requirements.

5 Conformance requirements

The facility files shall not be modified in any manner.

If there is a need to modify any of the macro definitions then this shall be done using the LaTeX `\renewcommand` and/or the `\renewenvironment` commands. These shall be placed in a new .sty file (or files) which shall be called in within the preamble of the document being typeset.

6 Fundamental concepts and assumptions

It is assumed that the reader of this document is familiar with the LaTeX document preparation system and in particular with the iso facilities.

NOTE 1 – Reference [1] in annex G describes the LaTeX system.

The reader is also assumed to be familiar with the ISO Directives Part 3 and the STEP Supplementary Directives.

If there are any discrepancies between the layout and wording of this document and the requirements of the ISO Directives Part 3 or the Supplementary Directives, then the requirements in those documents shall be followed for ISO 10303 standard documents.

Because of many revisions over the years to the files described herein, a naming convention has been adopted for them. The naming convention is that the primary name of the file is suffixed by v#, where # is the version number of the file in question. All file primary names have bee limited to a maximum of eight characters.

NOTE 2 – Table 1 shows the versions of the files that were current at the time of publication.

EXAMPLE 1 – At the time of publication of this document, any references to ir should be read as actually referring to irv1.sty.

NOTE 3 – This document is not a standard, although it has been laid out in a similar, but not necessarily identical, manner.

Table 1 – File versions current at publication time

Facility	File
step	stepv1.sty
ir	irv1.sty
ap	apv1.sty
aic	aicv1.sty
ats	atsv1.sty

7 The step facility

The step facility provides commands and environments specifically for the ISO 10303 series of documents.

7.1 Cover page

The command `\STEPcover{`⟨*commands*⟩`}` produces a cover page for a STEP document. The complete list of commands is shown below.

– `\wg{`⟨*working group*⟩`}` the working group or other committee producing the document e.g., WG 5

– `\docnumber{`⟨*number*⟩`}` the number of the document e.g., 156

– `\scivnumber{`⟨*number*⟩`}` the number given to the doc by SC4 for ballot purposes e,g. 273

– `\projnumber{`⟨*number*⟩`}` the project number e.g., 4

– `\docdate{`⟨*date*⟩`}` date of publication e.g., 3 July 1993

– `\oldwg{`⟨*working group*⟩`}` superseded working group e.g., WG 1

– `\olddocnumber{`⟨*number*⟩`}` number of previous document e.g., 107

– `\oldprojnumber{`⟨*number*⟩`}` superseded project number e.g., 2

– `\partnumber{`⟨*number*⟩`}` prints the number of the part of the standard e.g., 3456

– `\doctitle{`⟨*text*⟩`}` Short title of the document e.g., ISO Style file

– `\status{`⟨*text*⟩`}` the status of the document e.g., In work

– `\primcont` flags ‘Primary Content’

– `\issue` flags ‘Issue Discussion’

- `\altprop` flags ‘Alternate Proposal’
- `\partialcont` flags ‘Partial Content’
- `\abstract{`⟨*text*⟩`}` an abstract of the document
- `\keywords{`⟨*text*⟩`}` for listing relevant keywords
- `\dateproj{`⟨*date*⟩`}` date approved by project leader
- `\datewg{`⟨*date*⟩`}` date approved as released by WG convenor
- `\datequal{`⟨*date*⟩`}` date approved by Qualification
- `\dateint{`⟨*date*⟩`}` date approved as technically complete by Integration
- `\dateec{`⟨*date*⟩`}` date approved as editorially complete
- `\dateprojo{`⟨*date*⟩`}` date approved by project leader as ‘other’ document
- `\datewgo{`⟨*date*⟩`}` date approved by WG convenor as ‘other’ document
- `\datequalo{`⟨*date*⟩`}` date approved by Qualification as ‘other’ document
- `\dateinto{`⟨*date*⟩`}` date approved as technically complete by Integration for ‘other’ document
- `\dateeco{`⟨*date*⟩`}` date approved as editorially complete for ‘other’ document
- `\owner{`⟨*text*⟩`}` name of document owner/editor
- `\address{`⟨*text*⟩`}` address of owner
- `\telephone{`⟨*number*⟩`}` telephone/fax number of owner
- `\email{`⟨*text*⟩`}` Email address of owner
- `\altowner{`⟨*text*⟩`}` name of document alternate owner/editor
- `\altaddress{`⟨*text*⟩`}` address of alternate owner
- `\alttelephone{`⟨*number*⟩`}` telephone/fax number of alternate owner
- `\altemail{`⟨*text*⟩`}` Email address of alternate owner
- `\comread{`⟨*text*⟩`}` comments to the reader

Only use those commands within `\STEPcover` that are relevant to the purposes at hand. The order of the commands within `\STEPcover` is immaterial.

EXAMPLE 2 – The commands used to produce the cover sheet for one version of this document were:

```
\STEPcover{
\wg{EC}
\docnumber{41}
\oldwg{EC}
\olddocnumber{35}
\docdate{19 August 1994}
\partnumber{3456}
\doctitle{LaTeX Style file for ISO 10303}
\status{Working project draft}
\primcont
\abstract{This document describes the \LaTeX\ style files created for ISO~10303.
          It also describes the program GenIndex which provides some
          capabilities to assist in the creation of indexes for \LaTeX\
          documents in general.}
\keywords{\LaTeX, Style file, GenIndex, Index}
\dateprojo{Not yet}
\owner{Peter R Wilson}
\address{NIST\\
         Bldg. 220, Room A127 \\
         Gaithersburg, MD 20899 \\
         USA }
\telephone{$+\!1$ (301) 975-2976}
\email{\verb|pwilson@cme.nist.gov|}
\altowner{Tony Day}
\altaddress{Sikorsky Aircraft}
\comread{This document serves two purposes. Firstly, it provides a description
         of the current \LaTeX\ style file for ISO 10303. Secondly, the source
         can be used as an example of using the \LaTeX\ commands.} % end comread
} % end of STEPcover
```

7.2 Heading commands

The commands described in this section specify various 'standard' clause headings.

7.2.1 The Foreword commands

The `\Foreword` command specifies that a table of contents, list of figures and a list of tables be produced. Page numbering is roman style and the table of contents starts on page ii. A new unnumbered clause entitled Foreword is started containing both ISO required boilerplate and boilerplate text specific to ISO 10303.

Any text may be written after the `\Foreword` command. The Foreword clause is ended by the `\endForeword{`⟨*norm annexes*⟩`}{`⟨*inf annexes*⟩`}` command. This command takes two parameters.

a) ⟨*norm annexes*⟩ A phrase that starts the sentence '⟨*norm annexes*⟩ an integral part of this part ...'. If there are no normative annexes, then use an empty argument (i.e., `{}` with no spaces between the braces).

b) ⟨*inf annexes*⟩ A phrase that starts the sentence '⟨*inf annexes*⟩ for information purposes only.'. If there are no informative annexes, then use an empty argument.

The `\endForeword` command produces some additional boilerplate text specifically for ISO 10303.

EXAMPLE 3 – The LaTeX source for the Foreword for one version this document was:

```
\Foreword
\endForeword
{Annexes A, B and C are}  % normative annexes
{Annexes D, E, F, G, H, J and K are} % informative annexes
```

7.2.2 The Introduction environment

The `\begin{Introduction}{`⟨*class name*⟩`}` environment starts a new unnumbered clause entitled Introduction and adds some boilerplate text specifically for ISO 10303.

The environment takes one parameter — ⟨*class name*⟩ – which is the name of the part class the document belongs to (e.g. description methods, integrated resources, etc.).

EXAMPLE 4 – The following LaTeX source was used to specify the Introduction to this document.

```
\begin{Introduction}{descriptive methods}
\begin{majorsublist}
\item the \file{step} facility;
\item the \file{ir} facility;
\item the \file{ap} facility;
\item the \file{aic} facility;
\item the \file{atc} facility.
\end{majorsublist}

    This part of ISO 10303 specifies the \LaTeX\ facilities
specifically designed for use ...

\sclause*{Overview}

    This document describes a set of \LaTeX\ files for use
within ISO~10303 ...

\end{Introduction}
```

7.2.3 The stepparttitle command

The `\stepparttitle{`⟨*part title*⟩`}` command produces the title for a STEP part, where ⟨*part title*⟩ is the title of the part.

Table 2 – STEP package parameterless heading commands

Command	Clause	Default text
`\partidefhead`	SC	Terms defined in ISO 10303-1
`\otherdefhead`	SC	Other definitions
`\introsubhead`	SC	Introduction
`\fcandasubhead`	SC	Fundamental concepts and assumptions
`\shortnamehead`	NA	Short names of entities
`\picshead`	NA	Protocol Implementation Conformance Statement (PICS) proforma
`\objreghead`	NA	Information object registration
`\docidhead`	SC	Document identification
`\schemidhead`	SC	Schema identification
`\expresshead`	IA	*EXPRESS* listing
`\expressghead`	IA	*EXPRESS-G* figures
`\modelscopehead`	IA	Model scope
`\techdischead`	IA	Technical discussions

EXAMPLE 5 – The title for this document was produced using:

```
\stepparttitle{Documentation methods:
               The LaTeX package files reference manual}
```

7.2.4 Other headings

Most of these commands take no parameters. They start document clauses with particular titles. The commands that take no parameters are listed in table 2.

NOTE – In the tables, C = clause, SC = subclause, SSC = subsubclause, NA = normative annex, IA = informative annex.

The commands listed in table 3 are equivalent to the general sectioning commands, but are intended to indicate the start of a particular documentation element. With the exception of the `\entityhead` command, each of these takes a single parameter which is used in forming the heading title; the `\anentity` command takes two parameters, the second of which is normally empty.

7.3 Miscellaneous commands

The following commands provide some printing options for commonly occurring situations. The `\nexp{}` command is intended to be used for printing *EXPRESS* entity names etc.

- The command `\B{abc}` prints **abc**

- The command `\E{abc}` prints *abc*

- The command `\Express` prints *EXPRESS*

Table 3 – STEP package parameterized heading commands

Command	Clause	Parameterized title
`\refdefhead`	SC	Terms defined in ⟨*ISO ref*⟩
`\schemahead`	C	⟨*schema name*⟩
`\typehead`	SC	⟨*schema name*⟩ type definitions
`\atypehead`	SSC	⟨*type name*⟩
`\entityhead`	SC	⟨*schema name*⟩ entity definitions ⟨*group*⟩
`\anentityhead`	SSC	⟨*entity name*⟩
`\rulehead`	SC	⟨*schema name*⟩ rule definitions
`\arulehead`	SSC	⟨*rule name*⟩
`\functionhead`	SC	⟨*schema name*⟩ function definitions
`\afunctionhead`	SSC	⟨*function name*⟩
`\aschemaidhead`	SSC	⟨*schema name*⟩ identification

– The command `\ExpressG` prints *EXPRESS-G*

– The command `\ExpressI` prints *EXPRESS-I*

– The command `\BG{mathsymbol}` prints mathsymbol in bold font.

– The command `\HASH` prints `#`

– The command `\LT` prints `<`

– The command `\LE` prints `<=`

– The command `\NE` prints `<>`

– The command `\INE` prints `:<>:`

– The command `\GE` prints `>=`

– The command `\GT` prints `>`

– The command `\CAT` prints `||`

– The command `\QUES` prints `?`

– The command `\IEQ` prints `:=:`

– The command `\INEQ` prints `:<>:`

– The command `\nexp{an\_entity}` prints **an_entity**

The command `\ix{word or phrase}` both prints its parameter and also makes an index entry out of it.

The command `\mnote{<Marginal note text>}` prints its parameter as a marginal note.

Remember, though, that marginal notes are only printed when the draft option is used. This is because marginal notes are not allowed by ISO.

7.3.1 Standard reference commands

Many parts of STEP use the same normative or informative references. The most common of these are provided via commands.

The commands `\nrefparti`, `\nrefpartxi`, and `\nrefasni` are shorthands for normative references to STEP Parts 1 and 11, and to the ASN.1 standards, respectively.

EXAMPLE 6 – The normative references in this document were input as:

```
\begin{nreferences}
\isref{ISO Directives Part 3}{Drafting and presentation of International
           Standards.}
\isref{ISO TC 184/SC4 Editing N-48}{Supplementary directives for the
           drafting and presentation of ISO 10303 Version 2.3.}
\nrefparti
\nrefpartxi
\disref{ISO 10303-12:---}{Industrial automation systems and integration ---
        Product data representation and exchange ---
        Part 12: Description methods:
        The EXPRESS-I language reference manual.}
\nrefasni
\disref{P. R. WILSON:---}{LaTeX for standards.}
\end{nreferences}
```

Some informative bibliographic reference commands are also provided. The commands `\bibidefo` and `\bibidefix` produce the references to the IDEFO and IDEF1X documents, which are entries [10] and [11] in the bibliography of this document.

Corresponding to these commands are the commands `\brefidefo` and `\brefidefix` which can be used for citing these references in the body of the text.

EXAMPLES

7 – Part of the bibliography for this document looks like

```
\begin{references}
...
\reference{BRYAN, M.,}{SGML --- An Author's Guide to the Standard Generalized
           Markup Language,}{Addison-Wesley Publishing Co., 1988. }\label{bryan}
\bibidefo
\bibidefix
\reference{RESSLER, S.,}{The National PDES Testbed Mail Server User's Guide,}
           {NSTIR 4508, National Institute of Standards and Technology,
           Gaithersburg, MD 20899. January 1991.} \label{ressler}
...
\end{references}
```

8 – The source for one of the sentences above was

```
... which are entries \brefidefo\ and \brefidefix\ in the bibliography ...
```

7.4 Commands for documenting EXPRESS code

The Supplementary Directives specify the layout of the documentation of *EXPRESS* code. The following commands are intended to serve two purposes:

a) To provide environments for the documentation of entity attributes, etc.;

b) To provide begin and end tags around all the *EXPRESS* code documentation.

This latter purpose is to provide an enabling capability for the automatic extraction of portions of the documentation of an *EXPRESS* model so that they could be placed into another document. For example, tools could be developed that would automatically extract pieces of resource model documentation and place them into an AP document.

7.4.1 Environment ecode

The `ecode` environment is used for tagging *EXPRESS* code and setting up the relevant fonts.

EXAMPLE 9 – The following

```
\begin{ecode}
\begin{verbatm}  % read verbatm as verbatim
*)
ENTITY an_entity;
  attr : REAL;
END_ENTITY;
(*
\end{verbatm}   % read verbatm as verbatim
\end{ecode}
```

produces:

EXPRESS specification:

```
*)
ENTITY an_entity;
  attr : REAL;
END_ENTITY;
(*
```

7.4.2 Environment attrlist

The `attrlist` environment produces the heading for attribute definitions and sets up a `description` list.

EXAMPLE 10 – The following

```
\begin{attrlist}
\item[attr\_1] The \ldots
\item[attr\_2] This \ldots
\end{attrlist}
```

produces:

Attribute definitions:

attr_1: The ...

attr_2: This ...

7.4.3 Environment fproplist

The `fproplist` is similar to `attrlist` except that it is for formal propositions.

EXAMPLE 11 – The following

```
\begin{fproplist}
\item[un\_1] The value of \ldots\ shall be unique.
\item[gt\_0] The value of \ldots\ shall be greater than zero.
\end{fproplist}
```

produces:

Formal propositions:

un_1: The value of ... shall be unique.

gt_0: The value of ... shall be greater than zero.

7.4.4 Other listing environments

The environments `iproplist`, `enumlist`, and `arglist` are similar to `attrlist` etc. Respectively they are environments for informal propositions, enumerated items, and argument definitions.

7.4.5 Documentation tagging

Several environments are defined to tag the general documentation of *EXPRESS* code.

The environment `\begin{espec}{`⟨*name*⟩`}` may be used to enclose, and give a name to, a complete specification block for an *EXPRESS* entity. There are analogous environments — `fspec`, `rspec`, and `tspec` — for functions, rules and types respectively.

The environment `\begin{dtext}` may be used to anonymously enclose descriptive text.

EXAMPLE 12 – Here is the suggested tagged documentation style for part of an *EXPRESS* model.

```
%\ssclause{committee\_def}
\begin{espec}{committee_def}
\begin{dtext}
    A committee is composed of an odd number of people.
Each committee also has a name.
    The ideal size of a committee is less than three.

\begin{anote} Figures and tables may also be placed here. \end{anote} % end note
\end{dtext}
\begin{ecode}
\begin{verbatm} % read verbatm as verbatim
*)
ENTITY committee_def;
  title   : name;
  members : SET [1:?] OF person;
DERIVE
  ideal : BOOLEAN := SIZEOF(members) = 1;
UNIQUE
  un1 : title;
WHERE
  odd_members : ODD(SIZEOF(members));
END_ENTITY;
(*
\end{verbatm}   % read verbatm as verbatim
\end{ecode}
\begin{attrlist}
\item[title] The name of the committee.
\item[members] The people who form the committee.
\item[ideal] TRUE if there is only one person
             on the committee.
             That is, if the committee is the ideal size.
\end{attrlist}
\begin{fproplist}
\item[un1] The \nexp{title} of the committee shall be unique.
\item[odd\_members] There shall be an odd number of people
                    on the committee.
\end{fproplist}
\begin{iproplist}
\item[chair] The members of a committee shall appoint one of
             their number as
             chair of the committee.
\end{iproplist}
\end{espec}
```

EXAMPLE 13 – The code in example 12 produces the following result:

A committee is composed of an odd number of people. Each committee also has a name. The ideal size of a committee is less than three.

NOTE – Figures and tables may also be placed here.

EXPRESS specification:

```
*)
ENTITY committee_def;
  title   : name;
  members : SET [1:?] OF person;
DERIVE
  ideal : BOOLEAN := SIZEOF(members) = 1;
UNIQUE
  un1 : title;
WHERE
  odd_members : ODD(SIZEOF(members));
END_ENTITY;
(*
```

Attribute definitions:

title: The name of the committee.

members: The people who form the committee.

ideal: TRUE if there is only one person on the committee. That is, if the committee is the ideal size.

Formal propositions:

un1: The **title** of the committee shall be unique.

odd_members: There shall be an odd number of people on the committee.

Informal propositions:

chair: The members of a committee shall appoint one of their number as chair of the committee.

7.5 Commands producing boilerplate text

The following commands produce boilerplate text as specified by the Supplementary Directives. In the examples, the parameters of those commands that take them have been specified in either THIS FONT STYLE or *this font style* so their effects can be seen in the resulting printed text.

7.5.1 Major subdivision listing

The `majorsublist` environment prints the boilerplate for the heading of a listing of major subdivisions of the standard and starts an itemized list. An illustration of its use is given in example 4.

The heading text is produced by the `\majorsubname` command.

EXAMPLE 14 – The command `\majorsubname` command prints:

Major subdivisions of this International Standard are:

7.5.2 Registration commands

Three commands are provided for printing the boilerplate concerning object registration.

The command `\docreg{`⟨*version no*⟩`}` produces the boilerplate for document registration. The command takes one parameter: ⟨*version no*⟩ The version number (1 for first release).[3)]

EXAMPLE 15 – The command `\docreg{1}` prints:

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(3456) version(*1*) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

The command `\schemareg{`⟨*version no*⟩`}{`⟨*p2*⟩`}{`⟨*p3*⟩`}{`⟨*p4*⟩`}{`⟨*p5*⟩`}{`⟨*p6*⟩`}` produces the boilerplate concerning schema registration. The command takes six parameters.

a) ⟨*version no*⟩ The version number (1 for first release);

b) ⟨*p2*⟩ The name of an *EXPRESS* schema (with underscores);

c) ⟨*p3*⟩ The number of the object (typically 1);

d) ⟨*p4*⟩ The name of the schema, with hyphens replacing any underscores in the name;

e) ⟨*p5*⟩ The number identifying the schema;

f) ⟨*p6*⟩ The clause or annex in which the schema is defined.

EXAMPLE 16 – The command
`\schemareg{1}{\nexp{a\_schema}}{3}{a-schema}{5}{clause 6}` prints:

To provide for unambiguous identification of the **a_schema** in an open information system, the object identifier

{ iso standard 10303 part(3456) version(*1*) object(*3*) *a-schema*(*5*) }

is assigned to the **a_schema** schema (see *clause 6*). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

The command `\apschemareg{`⟨*version no*⟩`}{`⟨*p2*⟩`}{`⟨*p3*⟩`}{`⟨*p4*⟩`}{`⟨*p5*⟩`}{`⟨*p6*⟩`}` produces the boilerplate concerning application protocol schema registrations. The command takes six parameters.

a) ⟨*version no*⟩ The version number (1 for first release);

b) ⟨*p2*⟩ The name of an *EXPRESS* schema (with underscores);

[3)] There is a rumour that the version number should be 0 for a DIS document and -1 for a document that is at a pre-DIS stage.

c) ⟨*p3*⟩ The number of the object (typically 1);

d) ⟨*p4*⟩ The name of the schema, with hyphens replacing any underscores in the name;

e) ⟨*p5*⟩ The number identifying the long form schema;

f) ⟨*p6*⟩ The number identifying the short form schema.

EXAMPLE 17 – The command
`\apschemareg{1}{\nexp{a\_schema}}{3}{a-schema}{5}{6}` prints:

To provide for unambiguous identification of the schema specifications given in this application protocol **a_schema** in an open information system, object identifiers are assigned as follows:

{ iso standard 10303 part(3456) version(*1*) object(*3*) *a-schema*(*5*) }

is assigned to the **a_schema** expanded schema (see annex A).

{ iso standard 10303 part(3456) version(*1*) object(*3*) *a-schema*(*6*) }

is assigned to the **a_schema** short form schema (see 5.2).

The meaning of these values is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

8 The ir facility

The ir facility provides commands and environments specifically for the ISO 10303 Integrated Resources series of documents.

8.1 Boilerplate commands

The following commands produce boilerplate text as specified by the Supplementary Directives. In the examples, the parameters of those commands that take them have been specified in either THIS FONT STYLE or *this font style* so their effects can be seen in the resulting printed text.

8.1.1 Integrated resource short names

The command `\irshortnames` produces the boilerplate for the introduction to the integrated resource annex listing short names.

EXAMPLE 18 – The command `\irshortnames` prints:

Table A.1 provides the short names of entities specified in this part of ISO 10303. Requirements on the use of short names are found in the implementation methods included in ISO 10303.

8.1.2 Integrated resource EXPRESS listing

The command `\irexplisting` produces the boilerplate for the introduction to the integrated resource annex listing *EXPRESS* and short names.

EXAMPLE 19 – The command `\irexplisting` prints:

This annex provides a listing of the short names and a listing of the *EXPRESS* specified in this part of ISO 10303. No text or annotation is included. This annex is provided only in computer-interpretable form.

NOTE – The information provided on this diskette is informative; the normative text is that contained in the body of this part of ISO 10303.

8.1.3 Integrated resource EXPRESS-G

The command `\irexpressg{`⟨*figure range*⟩`}` produces the boilerplate for the introduction to the integrated resource *EXPRESS-G* annex.

EXAMPLE 20 – The command `\irexpressg{Figures D.1 through D.n}` prints:

FIGURES D.1 THROUGH D.N correspond to the *EXPRESS* listing given in annex C. The diagrams use the *EXPRESS-G* graphical notation for the *EXPRESS* language. *EXPRESS-G* is defined in annex D of ISO 10303-11.

9 The ap facility

The `ap` facility provides commands and environments specifically for the ISO 10303 Application Protocol series of documents.

9.1 Preamble commands

9.1.1 The Aptitle command

When the document is an AP, the command `\aptitle{`⟨*title of AP*⟩`}` shall be put into the preamble. The parameter shall be of such a form that it will read naturally in a sentence of the form: '…for the ⟨*title of AP*⟩ application protocol.'.

EXAMPLE 21 – For the purposes of later examples, this document has the command `\aptitle{coffee cup}` in its preamble.

9.1.2 The Aicinaptrue command

When the document is an AP which makes use of one or more AICs, then the command `\aicinaptrue` shall be put in the document preamble.

Table 4 – AP package parameterless heading commands

Command	Clause	Default text
`\inforeqhead`	C	Information requirements
`\uofhead`	SC	Units of functionality
`\applobjhead`	SC	Application objects
`\applasserthead`	SC	Application assertions
`\aimhead`	C	Application interpreted model
`\maptableheadhead`	SC	Mapping table
`\aimshortexphead`	SC	AIM *EXPRESS* short listing
`\confreqheadhead`	C	Conformance requirements
`\aimlongexphead`	NA	AIM *EXPRESS* expanded listing
`\aimshortnameshead`	NA	AIM short names
`\impreqhead`	NA	Implementation method specific requirements
`\aamhead`	IA	Application activity model
`\aamdefhead`	SC	Application activity model definitions and abbreviations
`\aamfighead`	SC	Application activity model diagrams
`\armhead`	IA	Application reference model
`\aimexpressghead`	IA	AIM *EXPRESS-G*
`\aimexpresshead`	IA	AIM *EXPRESS* listing
`\apusagehead`	IA	Application protocol usage guide

Table 5 – AP package parameterized heading commands

Command	Clause	Parameterized title
`\auofhead`	SSC	$\langle UoF \rangle$

9.2 Heading commands

These commands start document clauses with particular titles. The commands that take no parameters are listed in table 4.

The commands listed in table 5 take parameters.

9.3 Boilerplate commands

The following commands produce boilerplate text as specified by the Supplementary Directives. In the examples, the parameters of those commands that take them have been specified in either THIS FONT STYLE or *this font style* so their effects can be seen in the resulting printed text.

9.3.1 AP introduction

The command `\apintroend` produces the boilerplate for the last paragraph in the Introduction to an AP.

EXAMPLE 22 – The command `\apintroend` prints:

Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for the conformance testing of AP implementations.

Clause 1 defines the scope of the application protocol and summarizes the functionality and data covered by the AP. Clause 3 lists the words defined in this part of ISO 10303 and gives pointers to words defined elsewhere. An application activity model that is the basis for the definition of the scope is provided in annex F. The information requirments of the application are specified in clause 4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in annex G.

Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in 5.1, shows the correspondence between the information requirements and the AIM. The short listing of the AIM specifies the interface to the integrated resources and is given in 5.2. Note that the definitions and *EXPRESS* provided in the the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. The expanded listing given in annex A contains the complete *EXPRESS* for the AIM without annotation. A graphical representation of the AIM is given in annex H. Additional requirements for specific implementation methods are given in annex D.

9.3.2 AP scope

The command `\apscope{`⟨*application purpose and context*⟩`}` produces the boilerplate for the start of an AP scope clause.

EXAMPLE 23 – The command `\apscope{application purpose and context.}` prints:

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for APPLICATION PURPOSE AND CONTEXT.

NOTE – The application activity model in annex F provides a graphical representation of the processes and information flows which are the basis for the definition of the scope of this part of ISO 10303.

9.3.3 AP information requirements

The command `\apinforeq{`⟨*AP purpose*⟩`}` produces the boilerplate for the AP clause on information requirements.

EXAMPLE 24 – The command `\apinforeq{AP purpose.}` prints:

This clause specifies the information required for AP PURPOSE.

The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using the terminology of the subject area of this application protocol.

NOTES

1 – A graphical representation of the information requirements is given in annex G.

2 – The information requirements correspond to those of the activities identified as being within the scope of this application protocol in annex F.

3 – The mapping table specified in 5.1 shows how the integrated resources and application interpreted constructs are used to meet the information requirements of this application protocol.

9.3.4 AP UoF

The command `\apuof{`⟨*item list*⟩`}` produces the boilerplate for the introduction to the AP UoF clause.

EXAMPLE 25 – The command

```
\apuof{\item Name of UoF1;
       \item Name of UoF2;
       \item Name of UoFn. }
```

prints:

This subclause specifies the units of functionality for the coffee cup application protocol. This part of ISO 10303 specifies the following units of functionality:

– NAME OF UOF1;

– NAME OF UOF2;

– NAME OF UOFN.

The units of functionality and a description of the functions that each UoF supports are given below. The application objects included in the UoFs are defined in 4.2.

NOTE – This note is not printed by the command. Remember that `\aptitle` has been set to `coffee cup`.

9.3.5 AP assertions

The command `\apassert` produces the boilerplate for the AP assertions subclause.

EXAMPLE 26 – The command `\apassert` prints:

This subclause specifies the application assertions for the coffee cup application protocol. Application assertions specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects and UoFs. The application assertions and their definitions are given below.

NOTE – This note is not printed by the command. Remember that `\aptitle` has been set to `coffee cup`.

9.3.6 AP application objects

The command `\apapplobj` produces the boilerplate for the introduction to the AP clause on Application Objects.

EXAMPLE 27 – The command `\apapplobj` prints:

This subclause specifies the application objects for the coffee cup application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

NOTE – This note is not printed by the command. Remember that `\aptitle` has been set to `coffee cup`.

9.3.7 AP mapping table

The command `\apmappingtable` produces the boilerplate for the introduction to the AP mapping table.

NOTE – AICs are included in the boilerplate only if the command `\aicinaptrue` is included in the preamble.

EXAMPLE 28 – The command `\apmappingtable` prints:

This clause contains the mapping table that shows how each UoF and application object of this part of ISO 10303 (see clause 4) maps to one or more AIM constructs (see annex A). The mapping table is organized in five columns.

– Column 1) Application element: Name of an application element as it appears in the application object definition in 4.2. Application object names are written in uppercase. Attribute names and assertions are listed after the application object to which they belong and are written in lower case.

– Column 2) AIM element: Name of an AIM element as it appears in the AIM (see annex A), the term 'IDENTICAL MAPPING', or the term 'PATH'. AIM entities are written in lower case. Attribute names of AIM entities are referred to as <entity name> . <attribute name>. The mapping of an application element may result in several related AIM elements. Each of these AIM elements requires a line of its own in the table. The term 'IDENTICAL MAPPING' indicates that both application objects of an application assertion map to the same AIM element. The term 'PATH' indicates that the application assertion maps to the entire reference path.

– Column 3) Source: For those AIM elements that are interpreted from the integrated resources, this is the number of the corresponding part of ISO 10303. For those AIM elements that are created for the purpose of this part of ISO 10303, this is the number of this part. For those AIM elements that are directly incorporated from an application interpreted construct (AIC) this is the AIC reference.

– Column 4) Rules: One or more numbers may be given that refer to rules that apply to the current AIM element or reference path. For rules that are derived from relationships between application objects, the same rule is referred to by the mapping entries of all the involved AIM elements. The expanded names of the rules are listed after the table.

– Column 5) Reference path: To describe fully the mapping of an ARM element, it may be necessary to specify a reference path through several related AIM elements. The reference path column documents the role of an AIM element relative to the AIM element in the row succeeding it. Two or more such related AIM elements define the interpretation of the integrated resources that satisfies the requirement specified by the application object. For each AIM element that has been created for use within this part of ISO 10303, a reference path up to its supertype from an integrated resource is specified.

For the expression of reference paths the following notational conventions apply:

a) `[]` : multiple AIM elements or sections of the reference path are required to satisfy an information requirement;

b) `()` : multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;

c) `{}` : enclosed section constrains the reference path to satisfy an information requirement;

d) `->` : attribute references the entity or select type given in the following row;

e) `<-` : entity or select type is referenced by the attribute in the following row;

f) `[i]` : attribute is an aggregation of which a single member is given in the following row;

g) `[n]` : attribute is an aggregation of which member `n` is given in the following row;

h) `=>` : entity is a supertype of the entity given in the following row;

i) `<=` : entity is a subtype of the entity given in the following row;

j) `=` : the string, select, or enumeration type is constrained to a choice or value;

k) \ : the line continuations for strings that wrap.

9.3.8 AP short EXPRESS listing

The command `\apshortexpress` produces the boilerplate for the first paragraph of the AP short *EXPRESS* listing.

NOTE – AICs are included in the boilerplate only if the command `\aicinaptrue` is included in the preamble.

EXAMPLES

29 – The command `\apshortexpress` without `\aicinaptrue` in the preamble produces:

This clause specifies the *EXPRESS* schema that uses elements from the integrated resources and contains the types, entity specializations, rules, and functions that are specific to this part of ISO 10303. This clause also specifies modifications to the text for constructs that are imported from the integrated resources. The definitions and *EXPRESS* provided in the integrated resources for constructs used in the AIM may include select list items and subtypes that are not imported into the AIM. Requirements stated in the integrated resources that refer to such items and subtypes apply exclusively to those items that are imported into the AIM.

30 – With `\aicinaptrue` set in the preamble the command `\apshortexpress` produces the following:

This clause specifies the *EXPRESS* schema that uses elements from the integrated resources (and the AICs) and contains the types, entity specializations, rules, and functions that are specific to this part of ISO 10303. This clause also specifies modifications to the text for constructs that are imported from the integrated resources (and the AICs). The definitions and *EXPRESS* provided in the integrated resources for constructs used in the AIM may include select list items and subtypes that are not imported into the AIM. Requirements stated in the integrated resources that refer to such items and subtypes apply exclusively to those items that are imported into the AIM.

9.3.9 AP conformance

The command `\apconformance{`⟨*implementation methods*⟩`}` produces the boilerplate for the introduction to an AP conformance requirements clause.

The command `\apconformclasses{`⟨*item list*⟩`}` provides some additional boilerplate.

EXAMPLE 31 – The command `\apconformance{ISO 10303-21, ISO 10303-22}` prints:

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation method(s) supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods: *ISO 10303-21, ISO 10303-22*.

Requirements with respect to implementation methods-specific requirements are specified in annex C.

The Protocol Information Conformance Statement (PICS) proforma lists the options or the combination of options that may be included in the implementation. The PICS proforma is provided in annex D.

EXAMPLE 32 – The command

```
\apconformclasses{     \item first class;
                       \item second class;
                       \item last class.}
```

prints:

This part of ISO 10303 provides for a number of options that may be supported by an implementation. These options have been grouped into the following conformance classes:

– FIRST CLASS;

– SECOND CLASS;

– LAST CLASS.

Support for a particular conformance class requires support of all the options specified in this class.

9.3.10 AP short names

The command `\apshortnames` produces the boilerplate for the introduction to the AP short name annex.

EXAMPLE 33 – The command `\apshortnames` prints:

Table B.1 provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

9.3.11 AP PICS

The command `\picsannex` produces the boilerplate for the start of the AP PICS annex.

EXAMPLE 34 – The command `\picsannex` prints:

This clause lists the optional elements of this part of ISO 10303. An implementation may choose to support any combination of these optional elements. However, certain combinations of options are likely to be implemented together. These combinations are called conformance classes and are described in the subclauses of this annex.

This annex is in the form of a questionnaire. This questionnaire is intended to be filled out by the implementor and may be used in preparation for conformance testing by a testing laboratory. The completed PICS proforma is referred to as a PICS.

9.3.12 AAM annex

The command `\apaamintro` produces the introductory boilerplate for the AP annex on application activity models.

EXAMPLE 35 – The command `\apaamintro` prints:

The application activity model (AAM) is provided as an aid in understanding the scope and information requirements defined in this application protocol. The model is presented as a set of activity figures that contain the activity diagrams and a set of definitions of the activities and their data.

9.3.13 AP AAM definitions

The command `\apaamdefs` produces the boilerplate at the start of the AP clause on AAM definitions and abbreviations.

EXAMPLE 36 – The command `\apaamdefs` prints:

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this application protocol.

The definitions given in this annex do not supersede the definitions given in the main body of the text.

9.3.14 AAM figures annex

The command `\aamfigures{`⟨*figure range*⟩`}{`⟨*name of language*⟩`}` produces the boilerplate for the introduction to an APs AAM figure annex.

EXAMPLE 37 – The command `\aamfigures{figures F.1 through F.n}{IDEFo}` prints:

The application activity model diagrams are given in FIGURES F.1 THROUGH F.N. The graphical form of the application activity model is presented in the *IDEFo* activity modelling format. Activities and data flows that are out of scope are marked with asterisks.

9.3.15 ARM figures annex

The command `\armfigures{`⟨*figure range*⟩`}{`⟨*name of language*⟩`}` produces the boilerplate for the introduction to the ARM figures.

EXAMPLE 38 – The command `\armfigures{figures G.1 through G.n}{\ExpressG\/}` prints:

This annex provides the application reference model for this part of ISO 10303 and is given in FIGURES G.1 THROUGH G.N. The application reference model is a graphical representation of the structure and constraints of the application objects specified in clause 4. The graphical form of the application reference model is presented in the *EXPRESS-G* format. The application reference model is independent from any implementation method.

9.3.16 AIM EXPRESS-G annex

The command `\aimexpressg{`⟨*Figure range*⟩`}` produces the boilerplate for the introduction to an APs AIM *EXPRESS-G* model.

EXAMPLE 39 – The command `\aimexpressg{Figures H.1 through H.n}` prints:

FIGURES H.1 THROUGH H.N correspond to the AIM *EXPRESS* expanded listing given in annex A. The diagrams use the *EXPRESS-G* graphical notation for the *EXPRESS* language. *EXPRESS-G* is defined in annex D of ISO 10303-11.

9.3.17 AIM EXPRESS listing

The command `\aimexplisting` produces the boilerplate for the introduction to an AIMs short name and *EXPRESS* listing.

EXAMPLE 40 – The command `\aimexplisting` prints:

This annex provides a listing of the table of short names and a listing of the *EXPRESS* specified in the AIM of this part of ISO 10303. No text or annotation is included. This annex is provided only in computer-interpretable form.

NOTE – The information provided on this diskette is informative; the normative text is that contained in the body of this part of ISO 10303.

9.3.18 EXPRESS expanded listing

The command `\aimlongexp` produces the boilerplate for the introduction to an AIMs expanded *EXPRESS* listing.

EXAMPLE 41 – The command `\aimlongexp` prints:

The following *EXPRESS* is the expanded form of the short form schema given in 5.2. In the event of any discrepancy between the short form and this expanded listing, the expanded listing shall be used.

9.3.19 Requirements on exchange structure

The command `\apimpreq{`⟨*schema name*⟩`}` produces the boilerplate for the requirements on exchange structure.

EXAMPLE 42 – The command `\apimpreq{schema name}` prints:

The implementation method defines what types of exchange behaviour are required with respect to this part of ISO 10303. Conformance to this part of ISO 10303 shall be realized in an exchange structure. The file format shall be encoded according to the syntax and *EXPRESS* language mapping defined in ISO 10303-21 and in the AIM defined in annex A of this part of ISO 10303. The header of the exchange structure shall identify use of this part of ISO 10303 by the schema name 'SCHEMA NAME'.

10 The aic facility

The aic facility provides commands and environments specifically for the ISO 10303 Application Interpreted Construct series of documents.

NOTE 1 – The Supplementary Directives give few details on the documentation of AICs. The AIC package is likely to be extended in the future as more details become agreed.

10.1 Heading commands

The commands described in this section start document clauses with particular titles.

The commands that take no parameters are listed in table 6.

10.2 Boilerplate commands

The following commands produce boilerplate text as specified by the Supplementary Directives. In the examples, the parameters of those commands that take them have been specified in either THIS FONT STYLE or *this font style* so their effects can be seen in the printed text.

Table 6 – AIC package parameterless heading commands

Command	Clause	Default text
`\aicshortexphead`	C	*EXPRESS* short listing
`\aicshortnameshead`	NA	Short names of entities
`\aicexpresghead`	IA	*EXPRESS-G* diagrams

10.2.1 Introduction text

The command `\aicextraintro` prints additional boilerplate for the Introduction to an AIC.

EXAMPLE 43 – The command `\aicextraintro` prints:

An application interpreted construct (AIC) provides a logical grouping of interpreted constructs that supports a specific functionality for the usage of product data across multiple application contexts. An interpreted construct is a common interpretation of the integrated resources that supports shared information requirements among application protocols.

10.2.2 Definition of AIC

The command `\aicdef` prints the definition of 'AIC'. It shall only be used within the `definitions` environment.

EXAMPLE 44 – The commands

```
\begin{definitions}
\aicdef
\end{definitions}
```

produces:

10.2.5 application interpreted construct (AIC): a logical grouping of interpreted constructs that supports a specific functionality for the usage of product data across multiple application contexts.

10.2.3 Short EXPRESS listing

The command `\aicshortexpintro` prints boilerplate for the introduction to the short *EXPRESS* listing.

EXAMPLE 45 – The command `\aicshortexpintro` prints:

This clause specifies the *EXPRESS* schema that uses elements from the integrated resources and contains the types, entity specializations, and functions that are specific to this part of ISO 10303.

NOTE 1 – There may be subtypes and items of select lists that appear in the integrated resources that are not imported into the AIC. Constructs are eliminated from the subtype tree or select list through the use of the implicit interface rules of ISO 10303-11. References to eliminated constructs are outside the scope of the AIC.

10.2.4 EXPRESS-G figures

The command `\aicexpressg{`⟨*Figure range*⟩`}` prints boilerplate for the introduction to the *EXPRESS-G* figures.

EXAMPLE 46 – The command `\aicexpressg{Figures C.1 through C.n}` prints:

FIGURES C.1 THROUGH C.N correspond to the *EXPRESS* generated from the short listing given in clause 4 using the interface specifications of ISO 10303-11. The diagrams use the *EXPRESS-G* graphical notation for the *EXPRESS* language. *EXPRESS-G* is defined in annex D of ISO 10303-11.

11 The ats facility

The `ats` facility provides commands and environments specifically for the ISO 10303 Abstract Test Suite series of documents.

NOTE 1 – The Supplementary Directives give no details on the documentation of ATSs. The `ats` facility is based on *Guidelines for the development of abstract test suites*, ISO TC 184/SC4 WG6 N100b, 13 December 1995.

11.1 Preamble commands

When the document is an ATS the following commands shall be put in the preamble.

11.1.1 The APnumber command

The command `\APnumber{`⟨*number*⟩`}` shall be put in the preamble, where ⟨*number*⟩ is the ISO 10303 part number of the corresponding AP.

EXAMPLE 47 – For the purposes of later examples, this document has the command `\APnumber{299}` in the preamble.

11.1.2 The APtitle command

The command `\APtitle{`⟨*title of AP*⟩`}` shall be put in the preamble, where ⟨*title of AP*⟩ is the ISO 10303 part title of the corresponding AP. This must be given in such a manner that it reads sensibly in a sentence of the form '. . . for ISO 10303-299, application protocol ⟨*title of AP*⟩.'

EXAMPLE 48 – For the purposes of later examples, this document has the command `\APtitle{beer mug}` in the preamble.

Table 7 – ATS package parameterless heading commands

Command	Clause	Default text
`\purposeshead`	C	Test purposes
`\domainpurposehead`	SC	Domain test purposes
`\aepurposehead`	SC	AE test purposes
`\apasserthead`	SSC	Application assertions
`\aimpurposehead`	SC	AIM test purposes
`\extrefpurposehead`	SC	External reference test purposes
`\implementpurposehead`	SC	Implementation method test purposes
`\rulepurposehead`	SC	Rule violation test purposes
`\gtpvchead`	C	General test purposes and verdict criteria
`\generalpurposehead`	SC	General test purposes
`\gvcatchead`	SC	General verdict criteria for all abstract test cases
`\gvcprephead`	SC	General verdict criteria for preprocessor abstract test cases
`\gvcposthead`	SC	General verdict criteria for postprocessor abstract test cases
`\atchead`	C	Abstract test cases
`\confclassannexhead`	NA	Conformance classes

Table 8 – ATS package parameterized heading commands

Command	Clause	Parameterized title
`\apobjhead`	SSC	⟨*Application object n*⟩
`\confclasshead`	SC	Conformance class ⟨*number*⟩

11.2 Heading commands

These commands start document clauses with particular titles. The commands that take no parameters are listed in table 7.

The commands that take a parameter are listed in table 8.

Except for the `\apobjhead` command, all the commands in table 7 and 8 of the form `\...head` have an associated command of the form `\...name` which holds the text of the clause title.

EXAMPLE 49 – The command `\gvcatcname` prints:

General verdict criteria for all abstract test cases

11.3 Boilerplate commands

The following commands produce boilerplate text. In the examples, the parameters of those commands that take them have been specified in either THIS FONT STYLE or *this font style* so that their effects can be seen in the printed text.

11.3.1 ATS introduction

The command `\atsintroendbp` produces the boilerplate for the end of the Introduction to an ATS.

EXAMPLE 50 – The command `\atsintroendbp` prints:

The purpose of an abstract test suite is to provide a basis for evaluating whether a particular implementation of an application protocol actually conforms to the requirements of that application protocol. A standard abstract test suite helps ensure that evaluations of conformance are conducted in a consistent manner by different test laboratories.

This part of ISO 10303 specifies the abstract test suite for ISO 10303-299, application protocol beer mug. The abstract test cases presented here are the basis for conformance testing of implentations of ISO 10303-299.

This abstract test suite is made up of two major parts:

– the test purposes, the specific items to be covered by conformance test;

– the set of abstract test cases that meet those test purposes.

The test purposes are statements of the application protocol requirements that are to be addressed by the abstract test cases. Test purposes are derived primarily from the application protocol's AEs and AIM, as well as from other sources such as standards referenced by the application protocol and other requirements stated in the application protocol conformance requirements clause.

The abstract test cases address the test purpose by:

– specifying the requirements for input data to be used when testing an implementation of the application protocol;

– specifying the verdict criteria to be used when evaluating whether the implementation successfully converted the input data to a different form.

The abstract test cases set the requirements for the executable test cases that are required to actually conduct a conformance test. Executable test cases contain the scripts, detailed values, and other explicit information required to conduct a conformance test on a specific implementation of the application protocol.

At the time of publication of this document, conformance testing requirements had been established for implementations of application protocols in combination with ISO 10303-21 and ISO 10303-22. Accordingly, this part of ISO 10303 only specifies test purposes and abstract test cases appropriate to such implementations. For either ISO 10303-21 or ISO 10303-22, two kinds of implementations must be tested, preprocessors and postprocessors. The nature of the input data and verdict criteria are different for these two kinds of implementations. Both are addressed in this abstract test suite.

NOTE 1 – (This is not part of the boilerplate text) Remember that `\APnumber` was set to '299' in the preamble, and `\APtitle` was set to 'beer mug'.

11.3.2 ATS scope

The command `\atsscopebp` produces boilerplate for an ATS scope clause.

EXAMPLE 51 – The command `\atsscopebp` prints:

This part of ISO 10303 specifies the abstract test suite to be used in the conformance testing of implementations of ISO 10303-299. The following are within the scope of this part of ISO 10303:

– the specification of the test purposes associated with ISO 10303-299;

– the verdict criteria to be applied during conformance testing of an implementation of ISO 10303-299 using ISO 10303-21 or ISO 10303-22;

– the abstract test cases to be used as the basis for the executable test cases for conformance testing.

The following are outside the scope of this part of ISO 10303:

– the creation of executable test cases;

– testing other than conformance testing;

– other implementation methods.

11.3.3 Test purpose clause

The command `\atspurposebp{`⟨*comma or and*⟩`}{`⟨*stop or other*⟩`}{`⟨*clause number*⟩`}` prints boilerplate for the introduction for the test purposes clause.

EXAMPLE 52 – The command `\atspurposebp{+}{+ and others.}{235}` prints:

This clause specifies the test purposes for this part of ISO 10303. Test purposes are derived from the information requirements contained in clause 4 of ISO 10303-299+ the AIM *EXPRESS* schema in annex A of ISO 10303-299+ AND OTHERS. The test purposes are organized in this clause by type.

AE test purposes are individually identified by the prefix “ae” in the test purpose number. Each test purpose derived from the information requirements shall be interpreted as:

> Correctly instantiate in the implementation under test the semantic associated with the unique application concept corresponding to *(insert test purpose here)* in at least one test case within the test suite.

AE test purposes apply to the input specifications of both preprocessor and postprocessor test cases. AE test purposes are derived from the AP information requirements as follows:

– application objects (4.2 of ISO 10303-299). A test purpose derived from an application object is a simple statement of the object’s name. Each application object test purpose is documented in a separate subclause.

– application objects with categorisations (subtypes) (4.2 of ISO 10303-299). Test purposes derived from application objects with categorisations are statements of the application object name as a specific subtype.

– application object attributes (4.2 of ISO 10303-299). Test purposes derived from application object attributes are statements of the application object name with a specific attribute name.

– application asertions (4.3 of ISO 10303-299). Test purposes derived from application assertions are statements describing the relationship between two application objects. Application assertion test purposes address the directions of relationships as well as the number (cardinality) of relationships.

Each application object test purpose is listed as a separate subclause, with its related application object attribute test purposes. The application assertion test purposes form another subclause.

AIM test purposes are identified by the prefix “aim” in the test purpose number. Each test purpose derived from the AIM *EXPRESS* shall be interpreted as follows:

> Correctly instntiate in the implementation under test the AIM element associated with the unique AIM entity corresponding to *(insert test purpose here)* in at least one test case within the test suite.

AIM test purposes apply to the input specifications of postprocessor test cases only. AIM test purposes are derived directly from the expanded *EXPRESS* listing contained in annex A of ISO 10303-299 as follows:

– AIM entities. A test purpose derived from an AIM element is a simple statement of the entity name.

– AIM entity attributes. Test purposes derived from AIM entity attributes are statements of the AIM entity with a given attribute.

Each AIM entity test purpose is grouped with its attribute test purposes, all of which are contained in *235*.

11.3.4 General test purposes

The command `\generalpurposebp` prints the boilerplate for the General test purposes clause.

EXAMPLE 53 – The command `\generalpurposebp` prints:

The following are the general test purposes for this part of ISO 10303:

11.3.5 General test purposes and verdict criteria clause

The command `\atsgtpvcbp` prints boilerplate for the General test purposes and verdict criteria clause.

EXAMPLE 54 – The command `\atsgtpvcbp` prints:

General test purposes are statements of requirements that apply to all abstract test cases, all preprocessor abstract test cases, or all postprocessor abstract test cases. General verdict criteria are the means for evaluating whether the general test purposes are met. General verdict criteria shall be evaluated as a part of every executable test case to which they apply.

11.3.6 General verdict criteria for all abstract test cases clause

The command `\gvatcbp` prints boilerplate for the General verdict criteria for all abstract test cases clause.

EXAMPLE 55 – The command `\gvatcbp` prints:

The following verdict criteria apply to all abstract test cases contained in this part of ISO 10303:

11.3.7 General verdict criteria for preprocessor abstract test cases clause

The command `\gvcprepbp` prints boilerplate for the General verdict criteria for preprocessor abstract test cases clause.

EXAMPLE 56 – The command `\gvcprepbp` prints:

The following verdict criteria apply to all preprocessor abstract test cases contained in this part of ISO 10303:

11.3.8 General verdict criteria for postprocessor abstract test cases clause

The command `\gvcpostbp` prints boilerplate for the General verdict criteria for postprocessor abstract test cases clause.

EXAMPLE 57 – The command `\gvcpostbp` prints:

The following verdict criteria apply to all postprocessor abstract test cases contained in this part of ISO 10303:

11.3.9 Abstract test cases clause

The command `\atcbp` prints boilerplate for the Abstract test cases clause.

EXAMPLE 58 – The command `\atcbp` prints:

This clause specifies the abstract test cases for this part of ISO 10303. Each abstract test case addresses one or more test purposes from clause 4.

11.3.10 Conformance class annex

The command `\atsnoclassesbp` prints the entire boilerplate for the Conformance class annex when the AP has no conformance classes.

EXAMPLE 59 – The command `\atsnoclassesbp` prints:

Conformance to ISO 10303-299 is defined only in terms of the entire AP. Therfore, conformance requires that an implementation pass executable versions of all abstract test cases in clause 6.

The command `\confclassbp{`⟨*number*⟩`}` prints the boilerplate for the introduction to a Conformance class clause, where ⟨*number*⟩ is the number of the conformance class.

EXAMPLE 60 – The command `\confclassbp{27}` prints:

To conform to conformance class *27* of ISO 10303-299, an implementation must pass executable versions of the following abstract test cases:

Annex A
(normative)
Additional commands

A.1 STEP configuration commands

This clause describes the internal LaTeX commands that may be used to modify some of the boilerplate text in the step and other facility files.

A.1.1 The foreword

The \endForeword command uses two internal commands to specify the listing of STEP parts and the assignment of parts to the divisions within the STEP standard. The first of these is the command \fwdpartslist and the second is \fwddivslist. Both of these commands just \input a file — stpptlst.tex and stpdvlst.tex respectively. The contents of these files can be modified to reflect the actual status of STEP parts and the division list for the document in question.

A.1.2 AP boilerplate

Two commands in the ap facility \input boilerplate text files. These are the \apintroend command which calls in the apendint.tex file and the \apmappingtable command which calls in the apmptbl.tex file.

A.1.3 ATS boilerplate

Two commands in the ats facility \input boilerplate text files. These are the \atsintroendbp command which calls in the tsendint.tex file and the \atspurposebp command which calls in the atsprpbp.tex file.

Annex B
(normative)
Ordering of LaTeX commands

The LaTeX commands to produce an ISO 10303 document are:

```
\documentclass[<options>]{iso}
\usepackage{step}                                    % required package
\usepackage{ir}                                      % for an IR document
\usepackage{ap}                                      % for an AP document
\usepackage{aic}                                     % for an AIC document
\usepackage{atc}                                     % for an ATC document
\usepackage{<name>}                                  % additional packages
\standard{<standard number>}
\yearofedition{<year>}
\languageofedition{<code letter>}
\partno{<part number>}
\aptitle{<title of AP>}                              % if doc is an AP
\aicinaptrue                       % if doc is an AP that uses AICs
  % other preamble commands
\begin{document}
\STEPcover{< title commands >}
\Foreword                          % start Foreword and boilerplate
  % other text
\endForeword{<param1>}{<param2>}                     % boilerplate
\Introduction{<param1>}{<param2>}{<param3>}          % boilerplate
  \aicextraintro             % additional boilerplate for an AIC
  % other text for the introduction
  \apintroend                % boilerplate at end of Intro to an AP doc.
\stepparttitle{<Part title>}
\scopeclause                       % start scope clause
  \apscope                         % boilerplate if an AP
   % text of scope
\normrefsclause                    % start norm refs clause
  \normrefbp                       % boilerplate
  \begin{nreferences}
    % \item list of normative references
  \end{nreferences}
\defclause                         % definitions clause
  \partidefhead                    % defs from Part1 subclause
    % olddefinition list
  \refdefhead{<ISO 10303-NN>}      % defs from Part NN subclause
    % olddefinition list
  \otherdefhead                    % defs in this part
    % definition list
\symabbclause                      % Symbols & abbreviations clause
```

```
  % symbol lists
% THE BODY OF THE DOCUMENT
\bibannex                           % optional; the final Bibliography annex
  % bibliography listing
% the index
\end{document}
```

NOTES

1 – Remember that in the above the facility names should be replaced by the current file names.

2 – If the document is an Application Protocol the ap facility is to be included among the options.

3 – If the document is an AIC the aic facility is to be included among the options.

B.1 Body of a resource document

The body of a resource document has the following structure:

```
\schemahead{<Schema name>}          % repeat for each schema
  \introsubhead                     % intro subclause
     % text
  \fcandasubhead                    % concepts subclause
     % text
  \typehead{<Schema>}               % if type defs
     \atypehead{<type>}             % type heading
  \entityhead{<Schema>}{<group>}    % if entity defs
     \anentityhead{<entity>}        % entity heading
  \rulehead{<Schema>}               % if rule defs
     \arulehead{<rule>}             % rule heading
  \functionhead{<Schema>}           % if function defs
     \afunctionhead{<function>}     % function heading
% repeat above for each schema
\shortnamehead                      % Annex A: Short names of entities
  \irshortnames                     % boilerplate
  % list of short names
\objreghead                         % Annex B: Information object registration
  \docidhead                        % Document identification subclause
    \docreg{<param1>}                                    % boilerplate
  \schemaidhead                     % Schema identification subclause
% Either (for single schema)
     \schemareg{<6 parameters>}     % boilerplate
% Or (for multiple schemas) repeat:
     \aschemaidhead{<schema name>} % Schema id subsubclause
       \schemareg{<6 parameters>}
\expresshead                        % Annex C: EXPRESS listing
  \irexplisting                     % boilerplate
```

```
  % listing
\expressghead                      % Annex D: EXPRESS-G figures
  \irexpressg{<param1>}            % boilerplate
  %  EXPRESS-G diagrams
\modelscopehead                    % optional Model scope annex
  % text
```

B.2 Body of an application protocol

The body of an AP has the following structure:

```
\inforeqhead                   % Info. Req. clause
  \apinforeq{<param1>}         % boilerplate
  \uofhead                     % Units of functionality clause
    \apuof{<\item list>}       % boilerplate
    \auofhead{<UoF1>}          % repeat for each UoF
      % text
    \applobjhead               % Application objects
      \apapplobj               % boilerplate
    \applasserthead            % Application assertions
      \apassert                % boilerplate
\aimhead                       % Application interpreted model
  \maptablehead                % Mapping table
    \apmappingtable            % boilerplate
     % mapping table
  \aimshortexphead             % AIM EXPRESS short listing
    \apshortexpress            % boilerplate
\confreqhead                   % Conformance requirements
  \apconformance{<param1>}     % boilerplate
  \apconformclasses{<list>}    % optional boilerplate
\aimlongexphead                % Annex A: AIM EXPRESS expanded listing
  \aimlongexp                  % boilerplate
\aimshortnameshead             % Annex B: AIM short names
  \apshortnames                % boilerplate
\impreqhead                    % Annex C: Impl. specific reqs
  \apimpreq                    % boilerplate
\picshead                      % Annex D: PICS
  \picsannex                   % boilerplate
\objreghead                    % Annex E: Information object registration
  \docidhead                   % Document identification
    \docreg{<param1>}          % boilerplate
  \schemaidhead                % Schema identification
    \apschemareg{<6 params>}   % boilerplate
\aamhead                       % Annex F: AAM
  \apaamintro                  % boilerplate
  \aamdefhead                  % AAM defs and abbreviations
```

```
    \apaamdefs                  % boilerplate
  \aamfighead                   % AAM diagrams
    \aamfigures{<p1>}{<p2>}     % boilerplate
\armhead                        % Annex G: ARM
  \armfigures{<p1>}{<p2>}       % boilerplate
\aimexpressghead                % Annex H: AIM EXPRESS-G
  \aimexpressg{<p1>}            % boilerplate
\aimexpresshead                 % Annex J: AIM EXPRESS listing
  \aimexplisting                % boilerplate
\apusagehead                    % optional Annex: AP usage
\techdischead                   % optional Annex: Technical discussions
```

B.3 Body of an AIC

The body of an AIC has the following structure:

```
\aicshortexphead                % EXPRESS short listing clause
  \aicshortexpintro                                       % boilerplate
  \typehead{<Schema>}           % if type definitions
     \atypehead{<type>}         % repeat for each type
  \entityhead{<Schema>}{}       % if entity defs
     \anentityhead{<entity>}    % repeat for each entity
  \functionhead{<Schema>}       % if function defs
     \afunctionhead{<function>} % repeat for each function
\aicshortnameshead              % Annex A: Short names of entities
  \aicshortnames                % boilerplate
\objreghead                     % Annex B: Information object registration
  \docidhead                    % Document identification
    \docreg{<version no>}       % boilerplate
  \schemaidhead                 % Schema identification
    \schemareg{<6 parameters>}  % boilerplate
\aicexpressghead                % Annex C: EXPRESS-G diagrams
  \aicexpressg                  % boilerplate
\techdischead                   % optional Annex: Technical discussions
```

Annex C
(normative)

Information object registration

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(3456) version(-1) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex D
(informative)

Deprecated, deleted, new and modified commands

The facilities described herein are the result of a major overhaul, extension, and repackaging of earlier versions. This release is being treated as a new baseline. Consequently, the specific changes from earlier versions are not noted.

Annex E
(informative)
The relationship with SGML

SGML (Standard Generalized Markup Language) is a document tagging language that is described in ISO 8879 [7] and whose usage is described in [9], among others. The principal mover behind SGML is Charles Goldfarb from IBM, who has authored a detailed handbook 8 on the SGML standard.

The concepts lying behind both LaTeX and SGML are similar, but on the face of it they are distinctly different in both syntax and capabilities. ISO is moving towards electronic versions of its standard documents and, naturally, would like these to be SGML tagged. NIST have SGML tagged some STEP documents using manual methods, which are time consuming and expensive. There is a small effort underway to develop an auto-tagger that will (semi-)automatically convert a LaTeX tagged document to one with SGML tags.

Like LaTeX, SGML has a concept of style files, which are termed DTDs, and both systems support powerful macro-like capabilities. The design of an auto-tagger essentially boils down to being able to convert from a source document tagged according to a LaTeX style file to one which is tagged according to an SGML DTD. Also, reverse translation may be necessary as SGML has no means of displaying a finally formatted document — commercial SGML systems often use TeX or LaTeX as their printing engine.

Fully automatic conversion is really only possible if the authors' of the documents to be translated avoid using any 'non-standard' macros within their documents.

It is highly recommended that, for STEP purposes at least, document editors refrain from defining their own LaTeX macros. If new generally applicable LaTeX commands are found to be necessary, these should be sent to the editor of this document for incorporation into the iso facility and/or appropriate other facilities.

Some other points to watch when writing LaTeX documents that will assist in translations into SGML are given below. Typically, attention to these points will make it easier to parse the LaTeX source.

- Avoid using the `\label` command within clause headings or captions. It can just as easily be placed immediately after these constructs.

- Avoid using the `\index` command within clause headings or captions. It can just as easily be placed immediately after these constructs.

- Use all the specified tagging constructs when defining an *EXPRESS* model — this will also assist any program that attempts to extract *EXPRESS* source code and descriptive text from a document.

Annex F
(informative)
Obtaining LaTeX and friends

LaTeX is a freely available document typesetting system. There are many public domain additions to the basic system, for example the iso.cls and step.sty styles. The information below gives pointers to where you can obtain LaTeX etc., from the Internet. General information on Internet access is obtainable from many sources, for example reference [14].

F.1 CTAN

LaTeX runs on a wide variety of hardware, from PCs to Crays. Source to build a LaTeX system is freely available via anonymous ftp from what is called CTAN (Comprehensive TeX Archive Network). There are three sites; pick the one nearest to you.

- `ftp.dante.de` in Germany;
- `ftp.shsu.edu` in the USA;
- `ftp.tex.ac.uk` in the UK.

All three sites mirror each other; this means that they should all have the same contents (at least within 24 hours of each other). The top level directory for LaTeX and friends is `/tex-archive`. CTAN contains a wide variety of (La)TeX sources, style files, and software tools and scripts to assist in document processing.

Another mirror in the USA is `ftp.cdrom.com` and directory `pub/tex/ctan`. WWW access is via `http://www.cdrom.com`. This site is often more reliable than the one at `shsu`.

> NOTE – As of October 1994 they do not contain any of the style files or software developed to support STEP. These are available from SOLIS.

The UK and USA CTAN also support *gopher* access. A World Wide Web (hypertext) interface to the archives is available from:

`http://jasper.ora.com/CTAN/ctan.html`

Walsh [6] provides instructions on various non-ftp means of accessing CTAN and on building a LaTeX system.

Further information extracted from CTAN itself is given below. The best advice is to explore the archives yourself.

F.1.1 The archive

The following is a copy of the file `tex-archive/README.mirrors` obtained via ftp from the USA CTAN site in October 1994. Identical information should be obtainable from the other CTAN sites.

```
In order to reduce network load, it is recommended that you use the
```

```
Comprehensive TeX Archive Network (CTAN) host which is located in the
closest network proximity to your site.
Known partial mirrors of the CTAN reside on (alphabetically):
  dongpo.math.ncu.edu.tw (Taiwan)       /tex-archive
  ftp.adfa.oz.au (Australia)            /pub/tex/ctan
  ftp.germany.eu.net (Deutschland)      /pub/packages/TeX
  ftp.muni.cz (The Czech Republic)      /pub/tex/CTAN
  ftp.cs.ruu.nl (The Netherlands)       /pub/tex-archive
  ftp.uu.net (Virginia, USA)            /pub/text-processing/TeX
  nic.switch.ch (Switzerland)           /mirror/tex

Known mirrors of the CTAN reside on (alphabetically):
  ftp.center.osaka-u.ac.jp (Japan)      /CTAN
  ftp.ccu.edu.tw (Taiwan)               /pub/tex
  ftp.cs.rmit.edu.au  (Australia)       /tex-archive
  ftp.duke.edu (North Carolina, USA)    /tex-archive
  ftp.gwdg.de (Deutschland)             /pub/dante
  ftp.jussieu.fr (France)               /pub4/TeX/CTAN
  ftp.loria.fr (France)                 /pub/unix/tex/ctan
  ftp.mpi-sb.mpg.de (Deutschland)       /pub4/tex/mirror/ftp.dante.de
  ftp.uni-bielefeld.de (Deutschland)    /pub/tex
  ftp.uni-stuttgart.de (Deutschland)    /tex-archive (/pub/tex)
  ftpserver.nus.sg (Singapore)          /pub/zi/TeX
  kadri.ut.ee (Estonia)                 /pub/tex
  src.doc.ic.ac.uk (England)            /packages/tex/uk-tex
  sunsite.unc.edu (North Carolina, USA) /pub/packages/TeX
  wuarchive.wustl.edu (Missouri, USA)   /packages/TeX
Please send updates to this list to <CTAN-Mgr@SHSU.edu>.

The participating hosts in the Comprehensive TeX Archive Network are:
  ftp.dante.de  (Deutschland)
       -- anonymous ftp                 /tex-archive (/pub/tex /pub/archive)
       -- gopher on node sun.dante.de
       -- e-mail via ftpmail@dante.de
       -- Administrator: <ftpmaint@dante.de>
  ftp.shsu.edu  (Texas, USA)
       -- anonymous ftp and gopher      /tex-archive (/pub/tex /pub/archive)
       -- NFS mountable from ftp.SHSU.edu:/pub/ftp/tex-archive
       -- e-mail via ftpmail@ftp.SHSU.edu
       -- World Wide Web access on www.SHSU.edu
       -- Administrator: <CTAN-Mgr@SHSU.edu>
  ftp.tex.ac.uk (England)
       -- anonymous ftp                 /tex-archive (/pub/tex /pub/archive)
       -- gopher on node gopher.tex.ac.uk
       -- NFS mountable from nfs.tex.ac.uk:/public/ctan/tex-archive
       -- World Wide Web access on www.tex.ac.uk
       -- Administrator: <ctan-uk@tex.ac.uk>
```

The following is a copy of the file `tex-archive/README.archive-features` obtained via ftp from the USA CTAN site in October 1994. Identical information should be obtainable from the other CTAN sites.

```
This ftp archive allows for a number of important features on the fly.
```

```
                    Complete directory tree retrieval
You may retrieve an entire directory subtree as an archive file using
 GET DIRNAME.EXT
where DIRNAME is the name of the directory file, and EXT is the name of
the archiving mechanism you wish to retrieve.  Supported archiving
extensions include:
 .tar    -- standard U*ix tar archives (GNU tar)
 .tar.Z  -- LZW (U*ix "compress") compressed tar archive
 .tar.z  -- gzip compressed U*ix tar archive (uses gzip 1.2.3)
 .tar.gz -- gzip compressed U*ix tar archive (uses gzip 1.2.3)
 .zip    -- Info-ZIP platform-independent ZIP archive (uses ZIP 1.9)
 .zoo    -- platform independent ZOO archive (uses ZOO 2.10)
Sources and selected executables for the archiving tools for various
platforms are available in the /archive/archive-tools directory tree.

For example, to retrieve all files in a ZIP archive within the directory
  /archive/archive-tools/tar/vmstar
(the VMSTAR distribution), use:
 cd /archive/archive-tools/tar
 binary
 get vmstar.zip
This will create vmstar.zip on your system, which will UNZIP to create
the sub-directory vmstar on your system (if it does not already exist),
then place all files within that directory into the vms sub-directory on
your system.

Replace the extension .zip with any of the extensions identified above
as being supported.  Note that binary mode should be used to transfer
these files.

                                  +++
                  Automatic compression and decompression

Some files are retained in compressed format on this archive (shown by
by filenames such as FILENAME.Z, FILENAME.z or FILENAME.gz).  To retrieve
the file in uncompressed format, use:
 get FILENAME
and the file FILENAME.Z, FILENAME.z, or FILENAME.gz will be uncompressed
prior to transferring it to your system.

Alternately, many files are archived in uncompressed format (shown by a
a filename such as FILENAME which does not end in .Z, .z, or .gz).  To
retrieve the file in compressed format, use:
 get FILENAME.Z
 get FILENAME.z
or
 get FILENAME.gz
and the file FILENAME will be compressed prior to transferring it to your
system.  In the case of the ".Z" extension, the compression technique is
the LZW (Unix "compress") method; in the case of the ".z" or ".gz"
extensions, the compression technique is GNU's gzip.
```

```
                                  +++
                   Getting a listing of a ZIP archive file

To get a listing of the contents within a ZIP archive file, such as foo.zip
use:
  get foo.zip-lst
and the contents will be written to this file as it is created, then
transferred to your system.

                                  +++
                     Searching through the whole archive

To search for a file of a given name in the complete archive listing use:
  quote site index string
where "string" is your search string. Note that this will NOT work if you
entered a '-' as the first character of your password.
```

F.1.2 A listserver

The Sam Houston State University, as well as being the USA CTAN site, also supports some listservers that may be of interest.

NOTE – I do not know whether any of the other sites have similar capabilities.

The following is an edited combination of messages about these lists. The information in them reflects the status as of March 1994.

```
Sam Houston State University supports four lists of potential interest
to (La)TeX users -- Tex-Pubs, INFO-TeX, ctt-Digest, and LitProg.

TeX-Pubs is a redistribution list for TeX-related electronic-format
periodicals, including TeXhax, UKTeX, TeXMaG, the "Frequently Asked
Questions" and "Supplementary TeX Information" from the comp.text.tex
newsgroup, and the TeX Users Group's "TeX and TUG News".  TeX-Pubs provides
subscribers with a single address for handling the administrative details
associated with a subscription to each of these journals, so you will need
to subscribe or signoff only once instead of multiple times.  TeX-Pubs will
provide you with copies of each of these electronic digests as soon as they
are received for forwarding. To minimize the size of daily postings from
ctt-Digest, these periodicals are removed from the ctt-Digest distribution.
Users interested in receiving these periodicals are encouraged to subscribe
to to TeX-Pubs by including the command:
 SUBSCRIBE TeX-Pubs "Your Name"
in the body of a MAIL message to LISTSERV@SHSU.edu.

INFO-TeX is an unmoderated discussion list for TeX-related topics,
providing a broad range of discussion on virtually all topics associated
with TeX and it's derivative products, such as LaTeX, AMS(La)TeX, REVTEX,
METAFONT, etc.  It is mirrored to the USENET newsgroup comp.text.tex, so a
post to it is broadcast to quite possibly the widest audience of skilled
individuals to assist you.  You are invited to become involved on this list
```

```
or to simply monitor it.  To subscribe to INFO-TeX, please include the
command:
 SUBSCRIBE INFO-TeX "Your Name"
in the body of a mail message to the SHSU list server, LISTSERV@SHSU.edu.
If you would like to post a query to INFO-TeX (and on to comp.text.tex),
the address is INFO-TeX@SHSU.edu.

ctt-Digest is an automatically generated daily digest of posts made to
USENET's comp.text.tex newsgroup which do not originate on INFO-TeX.  Also,
efforts are made to exclude periodical postings, such as those carried on
TeX-Pubs.  The creation of this list allows mail-based subscribers to have
the widest coverage possible on all discussions associated with TeX-related
topics.  You are invited to receive these digests, along with your TeX-Pubs
postings and interactive access on INFO-TeX.  To subscribe to ctt-Digest,
please include the command:
 SUBSCRIBE ctt-Digest "Your Name"
in the body of a mail message to the SHSU list server, LISTSERV@SHSU.edu.

LitProg@SHSU.edu is an unmoderated discussion list on Literate Programming,
primarily focusing on the WEB structure of programming introduced by Donald
Knuth.  This includes general issues of style and philosophy, such as "what
is literate programming?" or "is literate programming compatible with
writing portable programs?", as well as specific issues relating to
particular literate programming systems, such as "is it possible to use
CWEB with ANSI C?"  Novices are welcome; it is intended that this group
should be a place where newcomers can be welcomed into the fold as well as
a place where seasoned literate programmers can discuss fine points of
technique.  If you would like to subscribe to LitProg, please include the
command:
 SUBSCRIBE LitProg "Your Name"
in the body of a mail message to LISTSERV@SHSU.edu.   If you would like to
post a query to LitProg, the address is LitProg@SHSU.edu.

If you have any questions or comments about these lists, or need assistance,
please do not hesitate to contact the list owner directly at any of the
addresses provided below.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
George D. Greenwade, Ph.D.                           Bitnet:  BED_GDG@SHSU
Department of Economics and Business Analysis        THEnet: SHSU::BED_GDG
College of Business Administration                   Voice: (409) 294-1266
P. O. Box 2118                                       FAX:   (409) 294-3612
Sam Houston State University             Internet:        bed_gdg@SHSU.edu
Huntsville, TX 77341                     bed_gdg%SHSU.decnet@relay.the.net
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

F.2 SOLIS

SOLIS is the *STEP On Line Information Service*. It contains many electronic sources of STEP related documents. The relevant top level directory is `pub/step`. In particular it contains the source for this document and the .sty files, as well as other LaTeX related files. The latest versions of the LaTeX related files are kept in the directory `pub/step/latex/current`.

There are several ways of accessing SOLIS. These are detailed by Ressler [12] and Rinaudot [13]. Copies of these reports may be obtained by telephoning the IPO Office at `+1 (301) 975-3983`. Abbreviated instructions are given below and are current as of September 1994.

F.2.1 Anonymous ftp

SOLIS is at `ftp.cme.nist.gov` (or `129.6.32.54`).

Login as `anonymous` and when requested for a password, supply your Email address. After the connection is established, do

```
cd pub/step
```

to get to the top level directory.

F.2.2 Kermit

a) Use a communications package that supports the kermit protocol.

b) Dial into the NIST modem pool at `+1 (301) 948-9720`.

c) When prompted by `Enter Username>` type in your last name.

d) Connect to SOLIS by typing in: `connect solis.cme`.

e) At the `Login:` prompt, type in `kermit`.

f) Answer the prompts to register yourself as a user.

You will then be logged into the kermit server and will be able to access all the files available on SOLIS.

F.2.3 Electronic mail

Send Email to `nptserver@cme.nist.gov`. Anything in the Subject line is ignored — all processed commands are in the body of the message. Each command must be on a separate line in the body. The following commands are available.

- `help` to get the help file (instructions for use of the archive server).
- `send step/howto/mailhelp.txt` to receive an explanation of the archive server.
- `index all` to get a list of directories, subdirectories and file names.
- `index <dname>` to get a list of the contents of the `<dname>` directory.
- `send step/<dname>` to receive all the files in the `<dname>` directory.

– `send step/<dname>/current` to receive all files of most recent release of the `<dname>` directory.

– `send step/<dname>/<subdname>/.../<filename>` to receive the particular `<filename>` file.

F.2.4 Gopher

The gopher server is running on `elib.cme.nist.gov` on port 70. There are two ways to access the SOLIS gopher server.

F.2.4.1 Gopher client

Connect to the host `elib.cme.nist.gov`.

For example, if your executable file is `gopher`, type

```
gopher elib.cme.nist.gov
```

You are now in SOLIS, using the gopher menu. Just select either the `step` directory or the `howto` directory and proceed. When you have found a file you would like to receive, there are two ways to download it.

a) Email the file. For this method, you must be viewing the file that you want to receive. Type `m` and a `Mail current document to:` window will appear. Enter your Email address, press the `<Enter>` key, and the file will be sent to you.

b) Save the file to your local machine. For this method, you must either be viewing the file or have the file selected in the gopher menu. Type `s` and a `Save in file:` window will appear. You can enter the name you want for the file or use the default provided by the gopher client, and then press the `<Enter>` key.

F.2.4.2 World Wide Web

For a World Wide Web (WWW) browser, such as Mosaic, use one of the following URLs:

– `gopher://elib.cme.nist.gov` or

– `http://elib.cme.nist.gov:70`

You are now in SOLIS, using the gopher menu. Just click on either the `step` directory or the `howto` directory and proceed. When you have found a file you would like to receive, there are two ways to download it.

NOTE – These instructions are specific to Mosaic for X Windows. The procedures for Mosaic for Microsoft Windows and Mosaic for Macintosh will differ slightly.

a) When you are viewing the file that you want, open the `File` pull down menu, select the `Mail to` option and fill in your Email address. The file will be Emailed to you.

b) You must be at the gopher menu that lists the file you want. Pull down the `Options` menu and click on `Load to Local Disk`. Select the file that you want (ASCII, WordPerfect, or Postscript) from the gopher menu.

The `Save Binary File to Local Disk` window will appear. Select the directory where you want to save the file and fill in the `Name for binary file on local disk:`.

Annex G
(informative)

Bibliography

[1] LAMPORT, L., *LaTeX — A Document Preparation System,* Addison-Wesley Publishing Co., 2nd edition, 1994

[2] WILSON, P.R., *The LaTeX isorot.sty file,* ISO TC184/SC4/EC N42, August, 1994.

[3] GOOSENS, M., MITTELBACH, F. and SAMARIN, A., *The LaTeX Companion,* Addison-Wesley Publishing Co., 1994

[4] CHEN, P. and HARRISON, M.A., *Index preparation and processing,* Software–Practice and Experience, 19(9):897–915, September 1988.

[5] KOPKA, H. and DALY, P.W., *A Guide to LaTeX,* Addison-Wesley Publishing Co., 1993.

[6] WALSH, N., *Making TeX Work,* O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472. 1994.

[7] ISO 8879:1986, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML)*

[8] GOLDFARB, C.F., *The SGML Handbook,* Oxford University Press, 1990.

[9] BRYAN, M., *SGML — An Author's Guide to the Standard Generalized Markup Language,* Addison-Wesley Publishing Co., 1988.

[10] *IDEF0 (ICAM Definition Language 0),* Federal Information Processing Standards Publication 183, Integration Definition for Information Modeling (IDEF0), FIPS PUB 183, National Institute for Standards and Technology, December 1993.

[11] *IDEF1X (ICAM Definition Language 1 Extended),* Federal Information Processing Standards Publication 184, Integration Definition for Information Modeling (IDEF1X), FIPS PUB 184, National Institute for Standards and Technology, December 1993.

[12] RESSLER, S., *The National PDES Testbed Mail Server User's Guide,* NSTIR 4508, National Institute of Standards and Technology, Gaithersburg, MD 20899. January 1991.

[13] RINAUDOT, G.R., *STEP On Line Information Service (SOLIS),* Draft NSTIR, National Institute of Standards and Technology, Gaithersburg, MD 20899. October 1994.

[14] KROL, E., *The Whole Internet — User's Guide & Catalog,* O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472. 1993.

Index