

ISO/WD 10303-3456.2**Product data representation and exchange: Documentation methods: LaTeX package files for ISO 10303: User manual**

COPYRIGHT NOTICE: This ISO document is a working draft or committee draft and is copyright protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by Participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purposes of selling it should be addressed as shown below (*via* the ISO TC184/SC4 Secretariat's member body) or to ISO's member body in the country of the requester.

Copyright Manager
ANSI
11 West 42nd Street
New York, New York 10036
USA
phone: +1-212-642-4900
fax: +1-212-398-0023

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement. Violators may be prosecuted.

ABSTRACT:

This document describes and illustrates the LaTeX macros for typesetting ISO 10303. The International Organisation for Standardisation (ISO) has specified editorial directives for all international standards published by them. The LaTeX macros described here were developed to meet additional editorial directives for ISO 10303.

KEYWORDS: LaTeX, document preparation, typesetting ISO standards

COMMENTS TO READER:

This document serves two purposes. Firstly, it provides a description of the current LaTeX macros for ISO 10303. Secondly, the source can be used as an example of using the LaTeX commands. Although the document is written as though it were a standard, it is not, and is not intended to become, a standard.

Project Leader: Peter R. Wilson

Address: NIST
Bldg. 220, Room A127
Gaithersburg, MD 20899
USA

Telephone: +1 (301) 975-2976

Telefacsimile:

Electronic mail: pwilson@cme.nist.gov

Project Editor: Peter R. Wilson

Address: Boeing Commercial Airplane
Group
PO Box 3701
MS 6H-AF
Seattle, WA 98124-2207
USA

Telephone: +1 (425) 237-3506

Telefacsimile: +1 (425) 237-3428

Electronic mail: peter.r.wilson@boeing.com

Contents		Page
1	Scope	1
2	Normative references	1
3	Terms, definitions, and abbreviations	2
3.1	Terms defined in ISO 10303-1	2
3.2	Other definitions	2
3.3	Abbreviations	3
4	Conformance requirements	3
5	Fundamental concepts and assumptions	3
6	The step package facility	5
6.1	Preamble commands	5
6.2	Cover page	5
6.3	Heading commands	7
6.3.1	The Foreword commands	7
6.3.2	The Introduction environment	12
6.3.3	The stepparttitle command	14
6.3.4	Other headings	14
6.4	Miscellaneous commands	15
6.4.1	Standard reference commands	16
6.5	Commands for documenting EXPRESS code	18
6.5.1	Environments ecode, eicode and excode	18
6.5.2	Environment attrlist	19
6.5.3	Environment fproplist	19
6.5.4	Other listing environments	19
6.5.5	Documentation tagging	20
6.6	Commands producing boilerplate text	22
6.6.1	Major subdivision listing	22
6.6.2	Short names of entities	22
6.6.3	Registration commands	22
6.6.4	Computer interpretable listings	24
7	The ir package facility	25
7.1	Boilerplate commands	25
7.1.1	Schema introduction	25
7.1.2	Integrated resource EXPRESS-G	25
8	The ap package facility	26
8.1	Preamble commands	26
8.1.1	The Aptitle command	26
8.1.2	The Aicinaptrue command	26
8.2	Heading commands	26
8.3	Boilerplate commands	26

8.3.1	AP introduction	26
8.3.2	AP scope	28
8.3.3	AP information requirements	28
8.3.4	AP UoF	28
8.3.5	AP assertions	29
8.3.6	AP application objects	29
8.3.7	AP mapping table	30
8.3.8	AIM short EXPRESS listing	31
8.3.9	AP conformance	32
8.3.10	EXPRESS expanded listing	32
8.3.11	AIM short names	33
8.3.12	Requirements on exchange structure	33
8.3.13	AP PICS	33
8.3.14	AAM annex	34
8.3.15	AP AAM definitions	34
8.3.16	AAM figures annex	34
8.3.17	ARM figures annex	35
8.3.18	AIM EXPRESS-G annex	35
8.3.19	AIM EXPRESS listing	35
9	The aic package facility	36
9.1	Heading commands	36
9.2	Boilerplate commands	36
9.2.1	Introduction text	36
9.2.2	Definition of AIC	36
9.2.3	Short EXPRESS listing	37
9.2.4	EXPRESS-G figures	37
10	The ats package facility	38
10.1	Preamble commands	38
10.1.1	The APnumber command	38
10.1.2	The APtitle command	38
10.2	Heading commands	38
10.3	Keyword commands	40
10.4	Boilerplate commands	40
10.4.1	ATS introduction	40
10.4.2	ATS scope	41
10.4.3	Test purpose	42
10.4.4	Application element test purposes	42
10.4.5	AIM test purposes	43
10.4.6	General test purposes and verdict criteria	44
10.4.7	General test purposes	44
10.4.8	General verdict criteria	45
10.4.9	General verdict criteria for preprocessor	45
10.4.10	General verdict criteria for postprocessor	45
10.4.11	Abstract test cases	46

10.4.12	Conformance class	47
10.4.13	Postprocessor input specification file names	47
Annex A (normative)	Additional commands	49
A.1	STEP configuration commands	49
A.1.1	The foreword	49
A.1.2	AP boilerplate	49
A.1.3	ATS boilerplate	49
A.1.4	Other boilerplate files	49
Annex B (normative)	Ordering of LaTeX commands	50
B.1	Body of a resource document	51
B.2	Body of an application protocol	52
B.3	Body of an AIC	53
B.4	Body of an ATS document	54
Annex C (normative)	Information object registration	56
Annex D (informative)	Deprecated, deleted, new and modified commands	57
D.1	New commands	57
D.2	Modified commands	58
D.3	Deleted commands	58
Annex E (informative)	LaTeX, the Web, and *ML	60
Annex F (informative)	Obtaining LaTeX and friends	62
F.1	CTAN	62
F.2	SOLIS	62
F.2.1	Anonymous ftp	63
F.2.2	World Wide Web	63
Bibliography	64
Index	66

Tables

1	File versions current at publication time	4
2	STEP package parameterless heading commands	14
3	STEP package parameterized heading commands	15
4	Commands for common references to standards	17
5	AP package parameterless heading commands	27
6	AP package parameterized heading commands	27
7	AIC package parameterless heading commands	36
8	ATS package parameterless heading commands	39
9	ATS package parameterized heading commands	39
10	ATS package keyword commands	40

Foreword

ISO (the International Organization for Standardization) 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 the 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*.

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 methods. The series are described in ISO 10301-1.

A complete list of parts of ISO 10303 is available from the Internet:

<http://www.nist.gov/sc4/editing/step/titles/>

This part of ISO 10303 is a member of the documentation methods series

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

Major subdivisions of this International Standard are:

- the `step` package facility;
- the `ir` package facility;
- the `ap` package facility;
- the `aic` package facility;
- the `ats` package facility.

This part of ISO 10303 is intended to be used in conjunction with *LaTeX for standards: The LaTeX package files user manual* which is based in part upon material in the ISO Directives Part 3 (*Drafting and presentation of International Standard*). The LaTeX facilities described here are based as well upon the specifications given in ISO TC 184/SC4 N858 (*Supplementary directives for the drafting and presentation of ISO 10303*).

Overview

This document describes a set of LaTeX macro files for use within ISO 10303, commonly called STEP (STandard for the Exchange of Product model data). The electronic source of this document also provides an example of the use of these files.

The current set of macro files have been developed by Peter Wilson (`peter.r.wilson@boeing.com`) from a macro file developed by Kent Reed (NIST) for LaTeX v2.09. In turn, this was a revision of files originally created by Phil Spiby (CADDETC), based on earlier work by Phil Kennicott (GE).¹⁾

NOTE It is important to remember that these macro files are only compatible with LaTeX2e.

Documents produced with the LaTeX files have been twice reviewed by the ISO Editorial Board in Geneva for conformance to their typographical requirements. The first review was of a set of

¹⁾In mid 1994 LaTeX was upgraded from version 2.09 to what is called LaTeX2e. The files described in this document are only applicable to LaTeX2e (support for LaTeX v2.09 was dropped in September 1997).

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 (ISO 10303:1994). 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 was 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.

The 1997 baseline release was also designed to cater for the fact that a major update of LaTeX to LaTeX2e took place during 1994. LaTeX2e is the only officially supported version of LaTeX.

Because ISO standard documents have a very structured layout, the `iso` class and the package files described here 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 packages described here are written for the English language, but the language-specific elements have been parameterized for easy modification for document publication in other official ISO languages, such as French.

With ISO’s move toward accepting documents in HTML, the advent of second editions of some of the STEP parts, and the second edition of the STEP Supplementary Directives, and not forgetting the new millenium, a 2000 baseline release has been developed and is documented here.

Industrial automation systems and integration — Product data representation and exchange — Part 3456 : Documentation methods: LaTeX package files for ISO 10303: User manual

1 Scope

This part of ISO 10303 describes a set of LaTeX facilities for typesetting documents according to the ISO Directives Part 3, together with the Supplementary Directives for drafting and presentation of ISO 10303.

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 normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

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

ISO TC 184/SC4 N858, *Supplementary directives for the drafting and presentation of ISO 10303, edition 2*.

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

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

ISO/TR 10303-12:1997, *Industrial automation systems and integration — Product data representation and exchange — Part 12: Description method: The EXPRESS-I language reference manual.*

ISO/IEC 8824-1:1995, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

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

3 Terms, definitions, and abbreviations

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 and which has been agreed to be present in a specific type of document.

3.2.2

style file

A set of LaTeX macros assembled into a single file with an extension `.sty`.

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. The macros may be defined in either a style file or a package file.

²⁾To be published.

3.3 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
IS	International Standard
ISOD	ISO directives Part 3
SD	Supplementary Directives — <i>Supplementary directives for the drafting and presentation of ISO 10303, edition 2</i>
IS-REVIEW	The ISO Editorial Board review (September 1994) of twelve IS documents for conformance to ISO typographical and layout requirements.

4 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.

There shall be no author specified `\label{...}` commands where the first two characters of the label are `;`s (semicolon and ‘s’); the creation of labels starting with these characters is reserved to the maintainer of the facility files.

NOTE For conformance to the `iso` class, author specified labels starting with the characters `;`i (semicolon and ‘i’) are prohibited.

5 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` class and associated facilities described in *LaTeX for standards: The LaTeX package files user manual*.

NOTE 1 Reference [1] describes the LaTeX system.

Table 1 – File versions current at publication time

Facility	File
step	stepv13.sty
ir	irv12.sty
ap	apv12.sty
aic	aicv1.sty
ats	atsv11.sty

The reader is also assumed to be familiar with the ISO Directives Part 3 and the Supplementary Directives for the drafting and presentation of ISO 10303.

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

The packages described herein have been designed to be used with the iso document class. It is highly unlikely that the packages will perform at all with any other LaTeX document class.

Because of many revisions over the years to the packages described herein, a naming convention has been adopted for the package files. 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 been 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 At the time of publication of this document, any references to ir.sty should be read as actually referring to irv12.sty.

NOTE 3 This document is not, and is never intended to become, a standard, although it has been laid out in a similar, but not necessarily identical, manner.

6 The step package facility

The `step` package facility provides commands and environments applicable to all the ISO 10303 series of documents.

6.1 Preamble commands

Certain commands shall be put in the preamble of any document.

The command `\partno{<number>}` is used to specify the Part number of the ISO 10303 standard.

The command `\series{<series title>}` is used to specify the name of the ISO 10303 series of which the Part is a member.

The command `\doctitle{<informal title>}` is used to specify the title to be used on the cover sheet.

The command `\ballotcycle{<number>}` is used to specify the ballot cycle number for the document.

6.2 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
- `\docdate{<date>}` date of publication e.g., 1993/07/03
- `\oldwg{<working group>}` superseded working group e.g., WG 1
- `\olddocnumber{<number>}` number of previous document e.g., 107
- `\abstract{<text>}` an abstract of the document
- `\keywords{<text>}` for listing relevant keywords
- `\owner{<text>}` name of the project leader
- `\address{<text>}` address of the project leaderwner
- `\telephone{<number>}` the project leader's telephone number

ISO/WD 10303-3456:1999(E)

- `\fax{<number>}` the project leader's fax number
- `\email{<text>}` Email address of the project leader
- `\altowner{<text>}` name of the editor of the document
- `\altaddress{<text>}` the editor's address
- `\alttelephone{<number>}` the editor's telephone number
- `\altfax{<number>}` the editor's fax number
- `\altemail{<text>}` the editor's Email address
- `\comread{<text>}` comments to the reader

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

EXAMPLE 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{1994/08/19}
\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}
\owner{Peter R Wilson}
\address{NIST\newline
         Bldg. 220, Room A127 \newline
         Gaithersburg, MD 20899 \newline
         USA }
\telephone{+1 (301) 975-2976}
\email{\texttt{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
```

Note the use of the `\newline` command instead of `\\` in the argument of the `\address` command to indicate a new line. The `\newline` is needed to ensure satisfactory conversion to HTML.

6.3 Heading commands

The commands described in this subclause specify various ‘standard’ clause headings.

6.3.1 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 1 The LaTeX source for the Foreword for one version this document was:

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

The `\ fwdpartslist` command produces boilerplate text for inclusion in the foreword describing the parts of STEP³⁾

EXAMPLE 2 The command `\ fwdpartslist` prints:

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 methods: The EXPRESS language reference manual;
- Part 12, Description method: The EXPRESS-I language reference manual;

³⁾This command is not required by the current SD.

ISO/WD 10303-3456:1999(E)

- Part 21, Implementation methods: 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 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 resources: Fundamentals of product description and support;
- Part 42, Integrated generic resources: Geometric and topological representation;
- Part 43, Integrated generic resources: Representation structures;
- Part 44, Integrated generic resources: Product structure configuration;
- Part 45, Integrated generic resource: Materials;
- Part 46, Integrated generic resources: 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 104, Integrated application resource: Finite element analysis;
- Part 105, Integrated application resource: Kinematics;
- Part 106, Integrated application resource: Building construction core model;
- Part 107, Engineering Analysis Core Application reference model (EA C-ARM);
- 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: Electronic assembly, interconnect, and packaging design;
- 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 mechanical 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 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 casting parts;
- Part 224, Application protocol: Mechanical product definition for process plans using machining features;
- 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 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 232, Application Protocol: Technical data packaging core information and exchange;

ISO/WD 10303-3456:1999(E)

- 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: Electronic assembly, interconnect, and packaging design;
- 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 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 casting parts;
- Part 324, Abstract test suite: Mechanical product definition for process plans using machining 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 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 332, Abstract test suite: Technical data packaging core information and exchange;
- 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 517, Application interpreted construct: Mechanical design geometric presentation;
- Part 518, Application interpreted construct: Mechanical design shaded representation.
- Part 519, Application interpreted construct: Geometric tolerances;
- Part 520, Application interpreted construct: Associative draughting.

The `\fwddivslst` command produces boilerplate text for inclusion in the foreword describing the STEP series⁴⁾

EXAMPLE 3 The command `\fwddivslst` prints:

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

⁴⁾This command is not required by the current SD.

ISO/WD 10303-3456:1999(E)

- Parts 11 to 12 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 232 specify the application protocols,
- Parts 301 to 332 specify the abstract test suites, and
- Parts 501 to 518 specify the application interpreted constructs.

The `\fwdshortlist` command produces boilerplate text for inclusion in the foreword referencing the STEP parts and series. This command replaces the `\fwdpartslist` and `\fwddivslst` commands.

EXAMPLE 4 The command `\fwdshortlist` prints:

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 methods. The series are described in ISO 10301-1.

A complete list of parts of ISO 10303 is available from the Internet:

<http://www.nist.gov/sc4/editing/step/titles/>

The `\steptrid` command produces boilerplate text for inclusion in the foreword describing the creators of a STEP Technical Report.

EXAMPLE 5 The LaTeX command `\steptrid` in this document prints:

ISO/TR 10303-3456, which is a Technical Report of type 2, was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

6.3.2 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 1 The following LaTeX source was used to specify the Introduction to this document.

```
\begin{Introduction}{descriptive methods}
```

This part of ISO 10303 specifies the `\latex{}` facilities specifically designed for use in preparing the various parts of this standard.

```
\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 is intended to be used ...

```
\sclause*{Overview}
```

This document describes a set of `\latex{}` files for use within ISO~10303 ...

```
\end{Introduction}
```

The SD require that explanatory boilerplate about typographical conventions be included as the final paragraphs in the Introduction. The command `\introend` generates the appropriate text. This command should be used immediately before the end of the `Introduction` environment.

EXAMPLE 2 The command `\introend` prints:

In this International Standard the same English language words may be used to refer to an object in the real world or to a concept, and as the name of an EXPRESS data type that represents this object or concept. The following typographical convention is used to distinguish between these. If a word or phrase occurs in the same typeface as narrative text, the referent is the object or concept. If the word or phrase occurs in a bold typeface, the referent is the EXPRESS data type. Names of EXPRESS schemas also occur in a bold typeface.

The name of an EXPRESS data type may be used to refer to the data type itself, or to an instance of the data type. The distinction between these uses is normally clear from the context. If there is a likelihood of ambiguity, the phrase ‘entity data type’ or ‘instance(s) of’ is included in the text.

Quotation marks “ ” are used to denote text that is copied from another document. Inverted commas ‘ ’ are used to denote particular string values.

Several components of this part of ISO 10303 are available in electronic form. This access is provided through the specification of Universal Resource Locators (URLs) that identify the location of these files on the Internet. If there is difficulty accessing these sites contact the ISO Central Secretariat or the ISO TC184/SC4 Secretariat directly at: sc4@cme.nist.gov.

Table 2 – STEP package parameterless heading commands

Command	Clause	Default text	Label
<code>\partidefhead</code>	SC	Terms defined in ISO 10303-1	
<code>\otherdefhead</code>	SC	Other definitions	
<code>\introsubhead</code>	SC	Introduction	
<code>\fcandasubhead</code>	SC	Fundamental concepts and assumptions	
<code>\shortnamehead</code>	NA	Short names of entities	<code>;ssne</code>
<code>\picshead</code>	NA	Protocol Implementation Conformance Statement (PICS) proforma	<code>;spics</code>
<code>\objreghead</code>	NA	Information object registration	<code>;sior</code>
<code>\docidhead</code>	SC	Document identification	
<code>\schemaidhead</code>	SC	Schema identification	
<code>\expresshead</code>	IA	EXPRESS listing	
<code>\listingshead</code>	IA	Computer interpretable listings	<code>;scil</code>
<code>\expressghead</code>	IA	EXPRESS-G diagrams	<code>;seg</code>
<code>\modelscopehead</code>	IA	Model scope	<code>;sms</code>
<code>\techdishead</code>	IA	Technical discussions	<code>;std</code>

6.3.3 The `\stepparttitle` command

The `\stepparttitle{<part title>}` command produces the title for an ISO 10303 part, where `<part title>` is the title of the part.

EXAMPLE The title for this document was produced using:

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

6.3.4 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. Some of these headings commands have predefined labels, which are also listed in the table.

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. These commands take either one or two parameters. The parameters are denoted in the column headed ‘Parameterized title’.

Table 3 – STEP package parameterized heading commands

Command	Clause	Parameterized title
<code>\refdefhead</code>	SC	Terms defined in <i><ISO ref></i>
<code>\schemahead</code>	C	<i><schema name></i>
<code>\singletypehead</code>	SC	<i><schema name></i> type definition: <i><type name></i>
<code>\typehead</code>	SC	<i><schema name></i> type definitions
<code>\atypehead</code>	SSC	<i><type name></i>
<code>\singleentityhead</code>	SC	<i><schema name></i> entity definition: <i><entity name></i>
<code>\entityhead</code>	SC	<i><schema name></i> entity definitions <i><group></i>
<code>\anentityhead</code>	SSC	<i><entity name></i>
<code>\singlerulehead</code>	SC	<i><schema name></i> rule definition: <i><rule name></i>
<code>\rulehead</code>	SC	<i><schema name></i> rule definitions
<code>\arulehead</code>	SSC	<i><rule name></i>
<code>\singlefunctionhead</code>	SC	<i><schema name></i> function definition: <i><function name></i>
<code>\functionhead</code>	SC	<i><schema name></i> function definitions
<code>\afunctionhead</code>	SSC	<i><function name></i>
<code>\aschemaidhead</code>	SSC	<i><schema name></i> identification

6.4 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
- The command `\ExpressG` prints EXPRESS-G
- The command `\ExpressI` prints EXPRESS-I
- The command `\ExpressX` prints EXPRESS-X
- 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 `iso class draft` option is used. Marginal notes are not allowed by ISO.

6.4.1 Standard reference commands

Many parts of STEP use the same normative or informative references. The most common of these are provided via commands. The currently available commands are listed in Table 4.

The naming convention used for references to parts of ISO 10303 is to end the command name with the number of the part expressed in lower case Roman numerals. Should further references to parts of ISO 10303 be added later, the same naming convention will be used.

EXAMPLE 1 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 N-432}{Supplementary directives for the
drafting and presentation of ISO~10303.}
\nrefparti
\nrefpartxi
\nrefpartxii
\nrefasni
\disref{P. R. WILSON:---}{LaTeX for standards: The LaTeX package files
user manual.}
\end{nreferences}
```

Table 4 – Commands for common references to standards

Standard	Command
ISO/IEC 8824-1	<code>\nrefasni</code>
ISO 10303-1	<code>\nrefparti</code>
ISO 10303-11	<code>\nrefpartxi</code>
ISO 10303-12	<code>\nrefpartxii</code>
ISO 10303-21	<code>\nrefpartxxi</code>
ISO 10303-22	<code>\nrefpartxxii</code>
ISO 10303-31	<code>\nrefpartxxxii</code>
ISO 10303-32	<code>\nrefpartxxxiii</code>
ISO 10303-41	<code>\nrefpartxli</code>
ISO 10303-42	<code>\nrefpartxlii</code>
ISO 10303-43	<code>\nrefpartxliii</code>

NOTE For the commands providing references to STEP parts, the part number is denoted by lowercase Roman numerals. Should further reference commands be provided for other STEP parts, then the same naming scheme will be used.

Some informative bibliographic reference commands are also provided. The commands `\bibidefo` and `\bibidefix` produce the references to the IDEF0 and IDEF1X documents, which are entries [13] and [14] 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.

EXAMPLE 2 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}
```

EXAMPLE 3 The source for one of the sentences above was:

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

6.5 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.

NOTE This document uses the `hyphenat` package which enables automatic hyphenation of ‘words’ containing the underscore character command (`_.`). Such words would normally have to be coded as `long_word` to ensure potential hyphenation at the position of the underscore. When using the `hyphenat` package it is an error to put the `\-` discretionary hyphen command after the underscore command as this then stops further hyphenation.

6.5.1 Environments `ecode`, `ecode` and `excode`

The `ecode` environment is for tagging EXPRESS code. It prints the appropriate title and sets up the relevant fonts.

EXAMPLE The following LaTeX source code:

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

produces:

EXPRESS specification:

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

Similarly, the `ecode` and `excode` environments are for tagging EXPRESS-I and EXPRESS-X code and setting up the relevant titles and fonts.

6.5.2 Environment `attrlist`

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

EXAMPLE The following LaTeX source code:

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

produces:

Attribute definitions:

attr_1: The ...

attr_2: This ...

6.5.3 Environment `fproplist`

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

EXAMPLE The following LaTeX source code:

```
\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.

6.5.4 Other listing environments

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

6.5.5 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`, `sspec`, and `tspec` — for functions, rules, schemas and types respectively.

The `dtext` environment may be used to anonymously enclose descriptive text.

EXAMPLE 1 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{verbatim} % read verbatim 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{verbatim} % read verbatim 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 \next{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 2 The code in example 1 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.

6.6 Commands producing boilerplate text

The following commands produce boilerplate text as specified by the Supplementary Directives.

NOTE In the examples, the parameters of those commands that take them have been specified in *this font style* so their effects can be seen in the resulting printed text.

6.6.1 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 1.

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

EXAMPLE The command `\majorsubname` command prints:

Major subdivisions of this International Standard are:

6.6.2 Short names of entities

The command `\shortnames` produces the boilerplate for the introduction to the annex listing short names.

EXAMPLE The command `\shortnames` 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.

6.6.3 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>` which is the version number.⁵⁾

EXAMPLE 1 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.

⁵⁾The SD say that the version number should be 1 for an IS, 0 for an FDIS or DIS, and -1 for CD document (presumably the number is either -1 or less for a WD document). The version number is incremented by one for each corrigenda, amendment or new edition.

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;
- 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 2 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;
- 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 long form schema;
- f) `<p6>` The number identifying the short form schema.

EXAMPLE 3 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.

6.6.4 Computer interpretable listings

The command `\expurls{<short>}{<express>}` produces the boilerplate for the introduction to the annex listing short names and EXPRESS, where `<short>` is the URL for the short names and `<express>` is the URL for the EXPRESS.

EXAMPLE The command `\expurls{http://www.short/}{http://www.express/}` prints:

This annex references a listing of the EXPRESS entity names and corresponding short names as specified in this part of ISO 10303. It also references a listing of each EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text. These listings are available in computer-interpretable form and can be found at the following URLs:

Short names: `<http://www.short/>`

EXPRESS: `<http://www.express/>`

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: `sc4sec@cme.nist.gov`.

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

7 The `ir` package facility

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

Use of this package requires the use of the `step` package.

7.1 Boilerplate commands

The following commands produce boilerplate text as specified by the SD.

NOTE In the examples, the parameters of those commands that take them have been specified in *this font style* so their effects can be seen in the resulting printed text.

7.1.1 Schema introduction

The command `\irschemaintro{<schema name>}` produces the boilerplate for the introduction to an EXPRESS schema clause.

EXAMPLE The command `\irschemaintro{this_schema}` prints:

The following EXPRESS declaration begins the *this_schema* and identifies the necessary external references.

7.1.2 Integrated resource EXPRESS-G

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

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

The diagrams in this annex correspond to the EXPRESS schemas specified in this part of ISO 10303. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

8 The **ap** package facility

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

Use of this package requires the use of the **step** package.

8.1 Preamble commands

8.1.1 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 For the purposes of later examples, the command `\aptitle{implicit drawing}` has been put in the preamble of this document.

8.1.2 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.

8.2 Heading commands

These commands start document clauses with particular titles. The commands that take no parameters are listed in Table 5. Some of these commands have predefined labels, which are also listed in the table.

The commands listed in Table 6 take parameters.

8.3 Boilerplate commands

The following commands produce boilerplate text as specified by the SD.

NOTE In the examples, the parameters of those commands that take them have been specified in *this font style* so their effects can be seen in the resulting printed text.

8.3.1 **AP** introduction

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

EXAMPLE 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.

Table 5 – AP package parameterless heading commands

Command	Clause	Default text	Label
<code>\inforeqhead</code>	C	Information requirements	<code>;sireq</code>
<code>\uofhead</code>	SC	Units of functionality	<code>;suof</code>
<code>\applobjhead</code>	SC	Application objects	<code>;sao</code>
<code>\applasserthead</code>	SC	Application assertions	<code>;saa</code>
<code>\aimhead</code>	C	Application interpreted model	<code>;saim</code>
<code>\mactableheadhead</code>	SC	Mapping table	<code>;smap</code>
<code>\aimshortexphead</code>	SC	AIM EXPRESS short listing	<code>;saesl</code>
<code>\confreqheadhead</code>	C	Conformance requirements	<code>;scr</code>
<code>\aimlongexphead</code>	NA	AIM EXPRESS expanded listing	<code>;saeel</code>
<code>\aimshortnameshead</code>	NA	AIM short names	<code>;sasn</code>
<code>\impreqhead</code>	NA	Implementation method specific requirements	<code>;simreq</code>
<code>\aamhead</code>	IA	Application activity model	<code>;saam</code>
<code>\aamdefhead</code>	SC	Application activity model definitions and abbreviations	
<code>\aamfighead</code>	SC	Application activity model diagrams	
<code>\armhead</code>	IA	Application reference model	<code>;sarm</code>
<code>\aimexpressghead</code>	IA	AIM EXPRESS-G	<code>;saeg</code>
<code>\aimexpresshead</code>	IA	AIM EXPRESS listing	
<code>\apusagehead</code>	IA	Application protocol usage guide	<code>;sapug</code>

Table 6 – AP package parameterized heading commands

Command	Clause	Parameterized title
<code>\auofhead</code>	SSC	<code><UoF></code>

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 requirements 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 C.

8.3.2 AP scope

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

EXAMPLE 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.

8.3.3 AP information requirements

The command `\apinforeq{<AP purpose>}` produces the boilerplate for the AP clause on information requirements.

EXAMPLE 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.

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

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

NOTE 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.

8.3.4 AP UoF

The environment `\begin{apuof}<item list>\end{apuof}` produces the boilerplate for the introduction to the AP UoF clause.

EXAMPLE The command

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

prints:

This subclause specifies the units of functionality for the implicit drawing 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 part of the printed boilerplate.
Remember that `\aptitle` has been set to `implicit drawing`.

8.3.5 AP assertions

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

EXAMPLE The command `\apassert` prints:

This subclause specifies the application assertions for the implicit drawing 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 part of the printed boilerplate.
Remember that `\aptitle` has been set to `implicit drawing`.

8.3.6 AP application objects

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

EXAMPLE The command `\apapplobj` prints:

This subclause specifies the application objects for the implicit drawing 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 part of the printed boilerplate.
Remember that `\aptitle` has been set to `implicit drawing`.

8.3.7 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 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 or the application interpreted constructs, 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. Entities or types that are defined within the integrated resources have an AIC as the source reference if the use of the entity or type for the mapping is within the scope of the AIC.

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 application object, 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) `[]` : enclosed section constrains multiple AIM elements or sections of the reference path are required to satisfy an information requirement;
- b) `()` : enclosed section constrains 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) `<>` : enclosed section constrains at one or more required reference path;
 - e) `||` : enclosed section constrains the supertype entity;
 - f) `->` : attribute references the entity or select type given in the following row;
 - g) `<-` : entity or select type is referenced by the attribute in the following row;
 - h) `[i]` : attribute is an aggregation of which a single member is given in the following row;
 - i) `[n]` : attribute is an aggregation of which member `n` is given in the following row;
 - j) `=>` : entity is a supertype of the entity given in the following row;
 - k) `<=` : entity is a subtype of the entity given in the following row;
 - l) `=` : the string, select, or enumeration type is constrained to a choice or value;
 - m) `\` : the reference path expression continues on the next line.

8.3.8 AIM short EXPRESS listing

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

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

EXAMPLE 1 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 select list items and subtypes apply exclusively to those items that are imported into the AIM.

EXAMPLE 2 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 select list items and subtypes apply exclusively to those items that are imported into the AIM.

8.3.9 AP conformance

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

The environment `\begin{apconformclasses}<item list>\end{apconformclasses}` provides some additional boilerplate.

EXAMPLE 1 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 2 The command

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

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.

8.3.10 EXPRESS expanded listing

The command `\aimlongexp` produces the boilerplate for the introduction to an AIM's expanded EXPRESS listing.

EXAMPLE 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.

8.3.11 AIM short names

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

EXAMPLE 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.

8.3.12 Requirements on exchange structure

The command `\apimpreq{<schema name>}` produces the boilerplate for the requirements on exchange structure.

EXAMPLE 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*'.

8.3.13 AP PICS

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

EXAMPLE 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.

8.3.14 AAM annex

The command `\aamfigrange{<figure range>}` is used to store the activity model diagram figure range for later use by the `\aamfigures` and `\apaamintro` commands.

EXAMPLE 1 For the purposes of this document we set

```
\aamfigrange{figures F.1 through F.n}
```

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

EXAMPLE 2 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 figures that contain the activity diagrams and a set of definitions of the activities and their data. The application activity model is given in *figures F.1 through F.n*. Activities and data flows that are out of scope are marked with an asterisk.

NOTE This note is not part of the printed boilerplate.
Remember that `\aamfigrange{figures F.1 through F.n}` has been set previously.

8.3.15 AP AAM definitions

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

EXAMPLE 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.

8.3.16 AAM figures annex

The command `\aamfigures` produces the boilerplate for the introduction to an APs AAM figure subclause.

EXAMPLE The command `\aamfigures` 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 IDEF0 activity modelling format. Activities and data flows that are out of scope are marked with asterisks.

NOTE This note is not part of the printed boilerplate.
Remember that `\aamfigrange{figures F.1 through F.n}` has been set previously.

8.3.17 ARM figures annex

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

EXAMPLE 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.

8.3.18 AIM EXPRESS-G annex

The command `\aimexpressg{<Figure range>}` produces the boilerplate for the introduction to an AP's AIM EXPRESS-G model.

EXAMPLE 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.

8.3.19 AIM EXPRESS listing

The command `\apexpurls{<short>}{<express>}` produces the boilerplate for the introduction to the AP annex listing short names and EXPRESS, where *<short>* is the URL for the short names and *<express>* is the URL for the EXPRESS.

EXAMPLE The command `\apexpurls{http://www.short/}{http://www.express/}` prints:

This annex references a listing of the EXPRESS entity names and corresponding short names as specified in the AIM this part of ISO 10303. It also references a listing of each EXPRESS schema specified in the AIM of this part of ISO 10303, without comments or other explanatory text. These listings are available in computer-interpretable form and can be found at the following URLs:

Short names: <http://www.short/>
 EXPRESS: <http://www.express/>

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@cme.nist.gov.

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Table 7 – AIC package parameterless heading commands

Command	Clause	Default text	Label
<code>\aicshortexphead</code>	C	EXPRESS short listing	<code>;sesl</code>

9 The aic package facility

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

The use of this package requires the use of the `step` package.

NOTE 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.

9.1 Heading commands

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

The commands that take no parameters are listed in Table 7.

9.2 Boilerplate commands

The following commands produce boilerplate text as specified by the Supplementary Directives.

NOTE 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.

9.2.1 Introduction text

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

EXAMPLE The command `\aicextraintro` prints:

This part of ISO 10303 is a member of the application interpreted construct series. 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.

9.2.2 Definition of AIC

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

EXAMPLE The commands:

```

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

```

produce:

9.2.2.1

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.

9.2.3 Short EXPRESS listing

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

EXAMPLE 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 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. In some cases, all items of the select list are eliminated. Because AICs are intended to be implemented in the context of an application protocol, the items of the select list will be defined by the scope of the application protocol.

9.2.4 EXPRESS-G figures

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

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

The diagrams in this annex are generated from the short listing given in clause 4 and correspond to the EXPRESS schemas specified in this part of ISO 10303. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

10 The `ats` package facility

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

The use of this package requires the use of the `step` package.

NOTE 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 N536, 1 March 1997.

10.1 Preamble commands

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

10.1.1 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 For the purposes of later examples, the command `\APnumber{299}` has been put in the preamble. of this document.

10.1.2 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 For the purposes of later examples, the command `\APtitle{abstract painting}` has been put in the preamble of this document.

10.2 Heading commands

These commands start document clauses with particular titles. The commands that take no parameters are listed in Table 8.

All the commands in Table 8 of the form `\...head` have an associated command of the form `\...name` which holds the text of the clause title.

The commands that take a parameter are listed in Table 9.

With the exception of the `\confclasshead` command, none of the `\...head` commands in Table 9 have an associated command of the form `\...name` holding part of the title text.

EXAMPLE The command `\gvcatcname` prints:

Table 8 – ATS package parameterless heading commands

Command	Clause	Default text
<code>\purposeshead</code>	C	Test purposes
<code>\domainpurposehead</code>	SC	Domain test purposes
<code>\aepurposehead</code>	SC	Application element test purposes
<code>\apasserthead</code>	SSC	Application assertions
<code>\aimpurposehead</code>	SC	AIM test purposes
<code>\extrefpurposehead</code>	SC	External reference test purposes
<code>\implementpurposehead</code>	SC	Implementation method test purposes
<code>\otherpurposehead</code>	SC	Other test purposes
<code>\gtpvchead</code>	C	General test purposes and verdict criteria
<code>\generalpurposehead</code>	SC	General test purposes
<code>\gvcatchead</code>	SC	General verdict criteria for all abstract test cases
<code>\gvcprehead</code>	SC	General verdict criteria for preprocessor abstract test cases
<code>\gvcposthead</code>	SC	General verdict criteria for postprocessor abstract test cases
<code>\atchead</code>	C	Abstract test cases
<code>\prephead</code>	SSC	Preprocessor
<code>\confclassannexhead</code>	NA	Conformance classes
<code>\postipfilehead</code>	NA	Postprocessor input specification file names
<code>\excludepurposehead</code>	NA	Excluded test purposes

Table 9 – ATS package parameterized heading commands

Command	Clause	Parameterized title
<code>\apobjhead</code>	SSC	<Application object <i>n</i> >
<code>\aimenthead</code>	SSC	<Entity name>
<code>\atctitlehead</code>	SC	<Title>
<code>\posthead</code>	SSC	<Postprocessor (<i>n</i>)>
<code>\confclasshead</code>	SC	Conformance class < <i>number</i> >

Table 10 – ATS package keyword commands

Command	Effect
<code>\atssummary</code>	<u>Test case summary:</u>
<code>\atscovered</code>	<u>Test purposes covered:</u>
<code>\atsinput</code>	<u>Input specification:</u>
<code>\atsconstraints</code>	<u>Constraints on values:</u>
<code>\atsverdict</code>	<u>Verdict criteria:</u>
<code>\atsexecution</code>	<u>Execution sequence:</u>
<code>\atsextra</code>	<u>Extra details:</u>

General verdict criteria for all abstract test cases

10.3 Keyword commands

Several keyword (headings) are used in documenting a test case. LaTeX commands for these keywords are given in Table 10.

10.4 Boilerplate commands

The following commands produce boilerplate text.

NOTE In the examples, the parameters of those commands that take them have been specified in *this font style* so that their effects can be seen in the printed text.

10.4.1 ATS introduction

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

EXAMPLE 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 abstract painting. The abstract test cases presented here are the basis for conformance testing of implementations 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 information requirements 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. This part of ISO 10303 only specifies test purposes and abstract test cases for a subset of such implementations.

For ISO 10303-21, two kinds of implementations, preprocessors and postprocessors, must be tested. Both of these are addressed in this abstract test suite.

For ISO 10303-22, a class of applications will possess the capability to upload and download AP-compliant SDAI-models or schema instances to and from applications that implement the SDAI. By providing test case data that correspond with SDAI-models, this abstract test suite addresses such applications in a single-schema scenario.

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble, and `\APtitle` was set to `abstract painting`.

10.4.2 ATS scope

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

EXAMPLE 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;

NOTE The verdict criteria are used to ascertain whether a test purpose has been sat-

isfactorily met by an implementation under test (IUT) within the context of a given test case.

- 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;
- test specifications for tests other than conformance testing such as interoperability or acceptance testing;
- other implementation methods.

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble.

10.4.3 Test purpose

The command `\atspurposebp{<stop or other>}{<sources>}` prints boilerplate for the introduction for the *Test purposes* clause.

`<stop or other>` shall be either a full stop (.) or a phrase starting ‘, and ...’ finishing with a full stop. The `<sources>` parameter is a (list of) sources, ending with a full stop.

EXAMPLE The command `\atspurposebp{, and others.}{sources.}` prints:

This clause specifies the test purposes for this part of ISO 10303. Test purposes in 4.1 and 4.2 are derived from the information requirements contained in clause 4 of ISO 10303-299 and the AIM EXPRESS schema in annex A of ISO 10303-299, *and others*. Each test purpose statement identifies some specific element from the AEs or the AIM. These test purpose statements implicitly require that the identified element, as specified in the test purpose statement, will be correctly instantiated by the implementation under test.

Implementation method test purposes in 4.3 are derived from ISO 10303-21. Domain test purposes in 4.4 are derived from implicit domain requirements *sources*. Implementation method and domain test purposes are individually identified by the prefix “other” in the test purpose number. These test purposes are statements of requirements that shall be met by a conforming implementation.

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble.

10.4.4 Application element test purposes

The command `\aetppb` prints boilerplate for the *Application element test purposes* clause.

EXAMPLE The command `\aetppb` prints:

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 given in the following statement:

The IUT shall preserve the semantic associated with the unique application element from which the test purpose was derived.

This implies that the semantics of the application element are preserved by the IUT between the input and output of a test, according to the reference path specified by the mapping table of the AP. 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:

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 (see 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) (see 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 (see 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 assertions (see 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 and assertion test purposes.

NOTE This note is not part of the printed boilerplate.
Remember that \APnumber was set to 299 in the preamble.

10.4.5 AIM test purposes

The command \aimtpbp prints boilerplate for the *AIM purposes* clause.

EXAMPLE The command \aimtpbp prints:

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 given in the following statement:

The postprocessor shall accept the input in accordance with the AIM EXPRESS structure corresponding to this test purpose.

This implies that the semantics of the application element represented by the AIM element are preserved by the IUT between the input and output of a test according to the reference path

specified in the mapping table of the AP. This also implies no violations of any constraints (e.g., where rules or global rules) that apply to the AIM element. 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. A test purpose derived from an AIM entity attribute is a statement of the AIM entity with a given attribute.

Each AIM entity test purpose is grouped with its attribute test purposes.

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble.

10.4.6 General test purposes and verdict criteria

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

EXAMPLE 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. Each general verdict criterion includes a reference to its associated test purpose.

10.4.7 General test purposes

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

EXAMPLE The command `\gtpbp` prints:

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

g1 The output of an IUT shall preserve all the semantics defined by the input model according to the reference paths specified in the mapping table defined in clause 5 of ISO 10303-299.

g2 The output of a preprocessor shall conform to the implementation method to which the IUT claims conformance.

g3 The instances in the output of a preprocessor shall be encoded according to the AIM EXPRESS long form and mapping table as defined in annex A and clause 5 of ISO 10303-299.

g4 A postprocessor shall accept input data which is encoded according to the implementation method to which the IUT claims conformance.

g5 A postprocessor shall accept input data structured according to the AIM EXPRESS long form

and mapping table as defined in annex A and clause 5 of ISO 10303-299.

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble.

10.4.8 General verdict criteria

The command `\gvatcbp` prints boilerplate for the *General verdict criteria* clause.

EXAMPLE The command `\gvatcbp` prints:

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

gvc1 The semantics of the input model are preserved in the output of the IUT according to the reference paths specified in the mapping table defined in clause 5 of ISO 10303-299 (g1).

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble.

10.4.9 General verdict criteria for preprocessor

The command `\gvcprebp` prints boilerplate for the *General verdict criteria for preprocessor* clause.

EXAMPLE The command `\gvcprebp` prints:

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

gvc2 The output of a preprocessor conforms to the implementation method to which the IUT claims conformance (g2).

gvc3 The instances in the output of a preprocessor are encoded according to the AIM EXPRESS long form and mapping table as defined in annex A and clause 5 of ISO 10303-299 (g3).

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble.

10.4.10 General verdict criteria for postprocessor

The command `\gvcpostbp` prints boilerplate for the *General verdict criteria for postprocessor* clause.

EXAMPLE The command `\gvcpostbp` prints:

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

gvc4 The postprocessor accepts input data which is encoded according to the implementation method

to which the IUT claims conformance (g4).

gvc5 The postprocessor accepts input data which is structured according to the AIM EXPRESS long form and mapping table as defined in annex A and clause 5 of ISO 10303-299 (g5).

NOTE This note is not part of the printed boilerplate.
Remember that \APnumber was set to 299 in the preamble.

10.4.11 Abstract test cases

The command `\atcbp` prints the first paragraph of the boilerplate for the *Abstract test cases* clause.

EXAMPLE 1 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. All the test purposes addressed by the test case are referenced either explicitly, in the test purposes covered sections, or indirectly, through the verdicted rows of the preprocessor input specification table.

The command `\atcbpii{<12 and/or 21>}` prints paragraphs 3 and onwards⁶⁾ of the boilerplate.

EXAMPLE 2 The command `\atcbpii{ISO 10303-12 and ISO 10303-21}` prints:

Each abstract test case has a subclause for the preprocessor test information and a subclause for the postprocessor test information. The preprocessor and postprocessor input specifications are mirror images of each other, i.e., they represent the same semantic information. The preprocessor input model is presented in the form of a table with five columns:

- The Id column is used to reference application objects for assertions and categorisations. It uses the same identifier as the test purpose associated with the application element in that row of the table;
- The V column specifies whether, or not, the element in that row of the table is verdicted in the test case. A blank indicates it is not verdicted. A “*” indicates that it is verdicted using a derived verdict criteria. The derived verdict criteria determine a number references a specific verdict criteria defined in the verdict criteria section that follows the preprocessor input specification table.
- The Application Elements column identifies the particular application element or categorisation instance that is being defined by the table. For assertions the role is specified in parenthesis.
- The Value column specifies a specific value for the application element. For application objects and attributes the value column defines the semantic value for that element’s instance in the input model. A “#<number>” in the column is a reference to an entity instance name in the postprocessor input specification where the corresponding value is specified. For assertions, this column holds a link to the related application object. For categorisations, the Value column

⁶⁾The boilerplate matches that specified in N536, but some of it makes little sense to me (PRW). See especially the description of the V column.

identifies the subtype application object. A “<not_present>” indicates that the application element or categorisation is not present in the input model.

- The Req column specifies whether the value in the Value column is mandatory (M), suggested (S) or constrained (Cn), where ‘n’ is an integer referencing a note that follows the table. A mandatory value may not be changed due to rules in EXPRESS, rules in the mapping table, or the requirements of the test purpose being verdicted. Each constrained value references a note labelled C<number> at the end of the preprocessor input model table and may be modified according to the specific constraints specified in it.

The postprocessor input specifications are defined using *ISO 10303-12 and ISO 10303-21*. The values in the postprocessor specifications are suggested unless declared mandatory or constrained by the preprocessor input table.

The abstract test case specifies all the verdict criteria which are used to assign a verdict during testing. Special verdict criteria for preprocessor and postprocessor testing are defined explicitly in each abstract test case subclause. The relevant derived verdict criteria for preprocessor and postprocessor testing are identified in the V column of the preprocessor input table.

10.4.12 Conformance class

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

EXAMPLE 1 The command `\atsnoclassesbp` prints:

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

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble.

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 2 The command `\confclassbp{27}` prints:

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

NOTE This note is not part of the printed boilerplate.
Remember that `\APnumber` was set to 299 in the preamble.

10.4.13 Postprocessor input specification file names

The command `\pisfbp` prints the boilerplate for a *Postprocessor input specification file names* annex.

EXAMPLE The command `\pisfbp` prints:

The postprocessor input specifications for each test case is supplied electronically on magnetic media (floppy diskette). Table C.1 lists the file name of the postprocessor input specification that is associated with the postprocessor subclause(s) of a test case.

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 package files.

A.1.1 The foreword

The `\endForeword` command in the `step` package 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 `\fwddivslst`. Both of these commands just `\input` a file — `stpplst.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.

NOTE The SD state that an up-to-date list of parts and divisions is maintained on SOLIS in the `sc4/step/editing` directory. The name of the relevant file is `titles`. It is provided in several formats which are indicated by the file extension (e.g., `titles.txt` for an ASCII text version).

A.1.2 AP boilerplate

Many of the boilerplate commands in the `ap` package `\input` the text from external files.

Two commands in particular are of note. 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

Many of the commands in the `ats` package `\input` boilerplate text files. The files all have the naming convention `bpfats...tex`.

A.1.4 Other boilerplate files

Apart from the files mentioned above, much other boilerplate text is maintained in external files. These typically are named `bpf...tex`.

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 identifier>}
\yearofedition{<year>}
\languageofedition{<parenthesized code letter>}
\partno{<part number>}
\series{<series title>}
\doctitle{<title on cover page>}
\ballotcycle{<number>}
\aptitle{<title of AP>}           % if doc is an AP
\aicinaptrue                 % if doc is an AP that uses AICs
\APnumber{<number>}          % if doc is an ATS
\APTtitle{<title>}           % if doc is an ATS
  % other preamble commands
\begin{document}
\STEPcover{< title commands >}
\Foreword                   % start Foreword & boilerplate
  % other text
  \fwdshortlist
\endForeword{<param1>}{<param2>} % end Foreword & boilerplate
\begin{Introduction}{}      % start Introduction & boilerplate
  \aicextraintro           % extra boilerplate for an AIC
  % your text
  \apintroend              % extra boilerplate at end of Intro to an AP
  \introend                % extra boilerplate at end of all docs
\end{Introduction}
\stepparttitle{<Part title>}
\scopeclause                 % Clause 1: Scope clause
  \apscope{<AP purpose>}   % boilerplate if an AP
  % text of scope
\normrefsclause             % Clause 2: Normative references
  \normrefbp{<document type>} % boilerplate
  \begin{nreferences}
    % \isref{}{} and/or \disref{}{} 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
  % bibliography listing
% the index
\end{document}

```

NOTE Remember that in the above the package names should be replaced by the current file names.

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>}
\listingshead % Annex C: Computer interpretable listings
  \expurls{<short>}{<express>} % boilerplate
\expressghead % Annex D: EXPRESS-G figures
  \irexpressg{} % 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 % Clause 4: Information requirements
  \apinforeq{<param1>} % boilerplate
  \uofhead % Clause 4.1: Units of functionality
    \begin{apuof} % boilerplate
      % \item list of UoFs
    \end{apuof}
  \auofhead{<UoF1>} % repeat for each UoF
    % text
  \applobjhead % Clause 4.2: Application objects
    \apappobj % boilerplate
    % text
  \applasserthead % Clause 4.3: Application assertions
    \apassert % boilerplate
    % text
\aimhead % Clause 5: Application interpreted model
  \mactablehead % Clause 5.1: Mapping table
  \apmappingtable % boilerplate
  % mapping table
  \aimshortexphead % Clause 5.2: AIM EXPRESS short listing
  \apshortexpress % boilerplate
  % text
\confreqhead % Clause 6: Conformance requirements
  \apconformance{<param1>} % boilerplate
  \begin{apconformclasses} % optional boilerplate
    % \item list
  \end{apconformclasses}
  % text
\aimlongexphead % Annex A: AIM EXPRESS expanded listing
  \aimlongexp % boilerplate
  % text
\aimshortnameshead % Annex B: AIM short names
  \apshortnames % boilerplate

```

```

    % text
\impreqhead           % Annex C: Impl. specific reqs
  \apimpreq{<schema name>} % boilerplate
\picshead            % Annex D: PICS
  \picsannex         % boilerplate
  % text
\objreghead          % Annex E: Information object registration
  \docidhead         % Annex E.1: Document identification
    \docreg{<param1>} % boilerplate
  \schemaidhead      % Annex E.2: Schema identification
    \apschemareg{<6 params>} % boilerplate
\amhead              % Annex F: Application activity model
  \amfigrange{<figure range>} % Figure range for AAM diagrams
  \apaamintro        % boilerplate
  % text
  \amdefhead         % Annex F.1: AAM defs and abbreviations
    \apaamdefs       % boilerplate
    % text
  \amfighead         % Annex F.2: AAM diagrams
    \amfigures       % boilerplate
    % IDEF0 diagrams
\armhead             % Annex G: Application reference model
  \armfigures{<p1>}{<p2>} % boilerplate
  % ARM figures
\aimexpressghead     % Annex H: AIM EXPRESS-G
  \aimexpressg{<p1>} % boilerplate
  % AIM figures
\listingshead        % Annex J: Computer interpretable listings
  \apexpurls{<short>}{<express>} % boilerplate
\apusagehead         % optional Annex: AP usage
  % text
\techdishead         % optional Annex: Technical discussions
  % text

```

B.3 Body of an AIC

The body of an AIC has the following structure:

```

\aicshortexphead     % Clause 4: EXPRESS short listing
  \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

```

<code>\aicshortnameshead</code>	% Annex A: Short names of entities
<code>\aicshortnames</code>	% boilerplate
<code>\objreghead</code>	% Annex B: Information object registration
<code>\docidhead</code>	% Annex B.1: Document identification
<code>\docreg{<version no>}</code>	% boilerplate
<code>\schemaidhead</code>	% Annex B.2: Schema identification
<code>\schemareg{<6 parameters>}</code>	% boilerplate
<code>\aicexpressghead</code>	% Annex C: EXPRESS-G diagrams
<code>\aicexpressg{}</code>	% boilerplate
<code>\listingshead</code>	% Annex D: Computer interpretable listings
<code>\expurls</code>	% boilerplate
<code>\techdishead</code>	% optional Annex: Technical discussions

B.4 Body of an ATS document

According to *ISO TC184/SC4 N536* the body of an Abstract Test Suite document has the following⁷⁾ structure:

<code>\purposeshead</code>	% Clause 4: Test purposes
<code>\atspurposebp{<p1>}{<p2>}</code>	% boilerplate
<code>\domainpurposehead</code>	% 4.1 Domain test purposes
<code>\aepurposehead</code>	% 4.2 Application element test purposes
<code>\aetpbp</code>	% boilerplate
<code>\apobjhead{<object>}</code>	% 4.2.n
...	
<code>\aimpurposehead</code>	% 4.3 AIM test purposes
<code>\aimtpbp</code>	% boilerplate
<code>\aimenthead{<entity>}</code>	% 4.3.n
...	
<code>\extrafpurposehead</code>	% (optional) External reference t.p
<code>\implementpurposehead</code>	% (optional) Implementation method t.p
<code>\otherpurposehead</code>	% (optional) Other test purposes
<code>\gtpvchhead</code>	% Clause 5: General t.p and verdict criteria
<code>\atsgtpvcbp</code>	% boilerplate
<code>\generalpurposehead</code>	% 5.1 General test purposes
<code>\gtpbp</code>	% boilerplate
...	
<code>\gvcatchhead</code>	% 5.2 General verdict criteria for all ATC
<code>\gvatcbp</code>	% boilerplate
...	
<code>\gvcprephead</code>	% 5.3 General verdict criteria for preprocessor
<code>\gvcprepbp</code>	% boilerplate
...	
<code>\gvcposthead</code>	% 5.4 General verdict criteria for postprocessor

⁷⁾N536 includes *Domain test purposes* but provides no explanation of its contents. I (PRW) think that perhaps it should not be there at all.

```

    \gvcpostbp          % boilerplate
    ...
\atchead              % Clause 6: Abstract test cases
  \atcbp              % boilerplate (para 1)
  % your para 2
  \atcbpii{<12 and/or 21>} % boilerplate (paras 3+)
  \atctitlehead{<title>} % 6.n an abstract test case
    \atssummary{} ... % Test case summary
    \atsexecution{} ... % Execution sequence
    \atsextra{} ... % Extra details
  \prephead          % 6.n.1 Preprocessor
    \atscovered{} ... % Test purposes covered
    \atsinput{} ... % Input specification
    \atsconstraints{} ... % Constraints on values
    \atsverdict{} ... % Verdict criteria
    \atsexecution{} ... % Execution sequence
    \atsextra{} ... % Extra details
  \posthead{<title>} % 6.n.2 Postprocessor
    \atscovered{} ... % Test purposes covered
    \atsinput{} ... % Input specification
    \atsverdict{} ... % Verdict criteria
    \atsexecution{} ... % Execution sequence
    \atsextra{} ... % Extra details
\confclassannexhead % Annex A: Conformance classes
  \atsnoclassesbp % boilerplate if no conformance classes, else
  \confclasshead{<number>} % A.n Conformance class <number>
  \confclassbp{<number>} % boilerplate
  ...
\objreghead          % Annex B: Information object registration
  \docreg{<partno>} % registration boilerplate
\postipfilehead     % Annex C: Postprocessor input file names
  \pisfbp           % boilerplate
  ...
\excldepurposehead % Annex D: Excluded test purposes
  ...

```

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

This release has involved many internal changes to the LaTeX .sty files. In particular boilerplate text is, as far as possible, maintained in external .tex files in order to save memory space within the LaTeX processor.

The major externally visible change is to the `\STEPcover` commands due to the recent simplification of the cover sheet. For authors, this involves some additional preamble commands, and the removal of many of the commands previously used in specifying the cover sheet.

The file for the `step` package is now `stepv13.sty`, for the `ir` package is `irv12.sty` for the `ap` package is `apv12.sty`, for the `ats` package is `atsv11.sty`, and for the `aic` package is `aicv1.sty`.

D.1 New commands

The commands that are new in this release are:

- `\fwdshortlist` — boilerplate for the foreword;
- `\introend` — boilerplate for final paragraphs of the introduction;
- `\series` — a preamble command;
- `\ballotcycle` — a preamble command;
- `\fax` — a `\STEPcover` command;
- `\altfax` — a `\STEPcover` command;
- `\steptrid` — additional Foreword boilerplate for a STEP TR document.
- Commands `\nrefpartxii`, `\nrefpartxxii`, `\nrefpartxxxii`, and `\nrefpartxxxiii` for normatively referencing ISO 10303 parts;
- `\ExpressI` — prints EXPRESS-I;
- `\ExpressX` — prints EXPRESS-X;
- `eicode` — environment for writing EXPRESS-I code;
- `excode` — environment for writing EXPRESS-X code;
- `\expurlls` — Computer interpretable listings listings boilerplate;

ISO/WD 10303-3456:1999(E)

- `\apexpur1s` — AP AIM shortnames/EXPRESS listings boilerplate;
- `\aamfigrange` — holds the figure range for Activity Model diagrams;
- Commands `\otherpurposehead`, `\postipfilehead`, `\excludepurposehead`, `\aimenthead`, `\atctitlehead`, `\prephead`, and `\posthead` for ATS headings;
- Commands `\atssummary`, `\atscovered`, `\atsinput`, `\atsconstraints`, `\atsverdict`, `\atsexecution`, and `\atsextra` for ATS keywords;
- Commands `\aetbp`, `\aimtpbp`, `\gtpbp`, `\atcbpii`, and `\pisfbp` for ATS boilerplate.

D.2 Modified commands

The commands that have been modified in this release are:

- `\doctitle` — is now a preamble command instead of a `\STEPcover` command;
- `\aamfigures` — now has no arguments rather than two;
- `\Foreword` — no longer lists STEP parts and series;
- `\Introduction` — the argument is unused (for now);
- `\irexpressg` — the argument is unused (for now);
- `\aicexpressg` — the argument is unused (for now);
- `\atspurposebp` — now has two arguments rather than three;
- Many of the heading commands include ‘built-in’ `\label{}`s and, where appropriate, boilerplate that references clauses, etc., uses `\ref{}`s to the labels instead of numeric values

D.3 Deleted commands

The commands that have been deleted in this release are:

- `\scivnumber` — was a `\STEPcover` command;
- `\projnumber` — was a `\STEPcover` command;
- `\oldprojnumber` — was a `\STEPcover` command;
- `\partnumber` — was a `\STEPcover` command;
- `\status` — was a `\STEPcover` command;

- `\primcont` — was a `\STEPcover` command;
- `\issue` — was a `\STEPcover` command;
- `\altprop` — was a `\STEPcover` command;
- `\partialcont` — was a `\STEPcover` command;
- `\dateproj` — was a `\STEPcover` command;
- `\datewg` — was a `\STEPcover` command;
- `\datequal` — was a `\STEPcover` command;
- `\dateint` — was a `\STEPcover` command;
- `\dateec` — was a `\STEPcover` command;
- `\dateprojo` — was a `\STEPcover` command;
- `\datewgo` — was a `\STEPcover` command;
- `\datequalo` — was a `\STEPcover` command;
- `\dateinto` — was a `\STEPcover` command;
- `\dateeco` — was a `\STEPcover` command;
- `\irexplisting` — was for IR shortnames/EXPRESS listing boilerplate;
- `\aimexplisting` — was for AP AIM shortnames/EXPRESS listing boilerplate;
- `\rulepurposehead` — was an ATS heading command;
- `\generalpurposebp` — was an ATS boilerplate command;

Annex E (informative) LaTeX, the Web, and *ML

ISO are becoming more interested in electronic sources for their standards as well as the traditional camera-ready copy. Acronyms like PDF, HTML, SGML and XML have been bandied about. Fortunately documents written using LaTeX are well placed to be provided in a variety of electronic formats. A comprehensive treatment of LaTeX with respect to this topic is provided by Goossens and Rahtz [4].

SGML (Standard Generalized Markup Language) is a document tagging language that is described in ISO 8879 [8] and whose usage is described in [10], among others. The principal mover behind SGML is Charles Goldfarb from IBM, who has authored a detailed handbook [9] 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 migrating towards electronic versions of its standard documents and, naturally, would prefer these to be SGML tagged. Like LaTeX, SGML has a concept of style files, which are termed DTDs, and both systems support powerful macro-like capabilities. SGML provides for logical document markup and not typesetting — commercial SGML systems often use TeX or LaTeX as their printing engine, as does the NIST SGML environment for ISO 10303 [11].

NIST have SGML tagged some STEP documents using manual methods, which are time consuming and expensive. In about 1997 there was a NIST effort underway to develop an auto-tagger that would (semi-) automatically convert a LaTeX tagged document to one with SGML tags. This tool assumed a fixed set of LaTeX macros and a fixed DTD. 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. 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. There is a program called `ltx2x` available from SOLIS, which replaces LaTeX commands within a document with user-defined text strings [12]. This can be used as a basis for a LaTeX to whatever auto-tagger, provided the LaTeX commands are not too exotic.

HTML is a simple markup language, based on SGML, and is used for the publication of many documents on the Web. XML is a subset of SGML and appears to be taken up by every man and his dog as *the* document markup language. HTML is being recast in terms of XML instead of SGML. PDF is a page description language that is a popular format for display of documents on the Web.

LaTeX documents can be output in PDF by using pdfLaTeX. Instead of a .dvi file being produced a .pdf file is output directly. The best results are obtained when PostScript fonts rather than Knuth's cm fonts are used. Noting that the `isov2` class provides an `\ifpdf` command, a general form for documents to be processed by either LaTeX or pdfLaTeX is

```

\documentclass{isov2}
\usepackage{times}      % PostScript fonts Times, Courier, Helvetica
\ifpdf
  \pdfoutput=1          % request PDF output
  \usepackage[pdftex]{graphicx}
\else
  \usepackage{graphicx}
\fi
...

```

There are several converters available to transform a LaTeX document into an HTML document, but like `ltx2x` they generally do their own parsing of the source file, and unlike `ltx2x` are typically limited to only generating HTML. Eitan Gurari's `TeX4ht` suite is a notable exception (see Chapter 4 and Appendix B of [4]). It uses the `.dvi` file as input, so that all the parsing is done by TeX, and can be configured to generate a wide variety of output formats. A set of `TeX4ht` configuration files are available for converting STEP LaTeX documents into HTML⁸⁾.

It is highly recommended that for the purposes of ISO 10303, 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` class, the `step` package and/or appropriate other packages.

Some other points to watch when writing LaTeX documents that will assist in translations into *ML 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.

⁸⁾Later, configuration files for XML output will be developed.

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 [17].

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 two sites and several mirrors; pick the one nearest to you.

- `ftp.dante.de` CTAN in Germany;
- `ftp.tex.ac.uk` CTAN in the UK;
- `ctan.tug.org` CTAN in the USA;
- `ftp.cdrom.com/pub/tex/ctan` a mirror in the USA.

The top level CTAN 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.

NOTE 1 CTAN is maintained by the TeX Users Group (TUG). Their homepage <http://www.tug.org> should be consulted for the current list of CTAN sites and mirrors.

NOTE 2 The master copies of the LaTeX files described in this document are maintained on SOLIS. They may be mirrored from SOLIS to the CTAN.

World Wide Web (hypertext) interfaces to the archives is available from:

<http://www.cdrom.com/pub/tex/ctan>
<http://jasper.ora.com/CTAN/ctan.html>

Walsh [7] provides instructions on various non-ftp means of accessing CTAN and on building a LaTeX system.

F.2 SOLIS

SOLIS is the *SC4 On Line Information Service*. It contains many electronic sources of STEP related documents. The relevant top level directory is `pub/subject/sc4`. In particular, SOLIS contains the source for this document and the `.sty` files, as well as other LaTeX related files. The LaTeX root directory is `sc4/editing/latex`. The latest versions of the LaTeX related files are

kept in the sub-directory `latex/current`. Some LaTeX related programs are also available in the `latex/programs` sub-directory.

There are several ways of accessing SOLIS. These are detailed by Ressler [15] and Rinaudot [16]. Copies of these reports may be obtained by telephoning the IPO Office at +1 (301) 975-3983. Abbreviated instructions for the most popular means of access are given below and are current as of September 1997.

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/subject/sc4
```

to get to the top level directory.

F.2.2 World Wide Web

For a World Wide Web (WWW) browser use the following URL:

<<http://www.nist.gov/sc4>>

Bibliography

- [1] LAMPORT, L., *LaTeX — A Document Preparation System*, Addison-Wesley Publishing Co., 2nd edition, 1994
- [2] WILSON, P. R., *LaTeX for standards: The LaTeX package files user manual*, NISTIR, National Institute of Standards and Technology, Gaithersburg, MD 20899. June 1996.
- [3] GOOSENS, M., MITTELBACH, F. and SAMARIN, A., *The LaTeX Companion*, Addison-Wesley Publishing Co., 1994
- [4] GOOSENS, M. and RAHTZ, S., *The LaTeX Web Companion — Integrating TeX, HTML, and XML*, Addison-Wesley Publishing Co., 1999
- [5] CHEN, P. and HARRISON, M.A., *Index preparation and processing*, Software-Practice and Experience, 19(9):897–915, September 1988.
- [6] KOPKA, H. and DALY, P.W., *A Guide to LaTeX*, Addison-Wesley Publishing Co., 1993.
- [7] WALSH, N., *Making TeX Work*, O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472. 1994.
- [8] ISO 8879:1986, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML)*
- [9] GOLDFARB, C.F., *The SGML Handbook*, Oxford University Press, 1990.
- [10] BRYAN, M., *SGML — An Author's Guide to the Standard Generalized Markup Language*, Addison-Wesley Publishing Co., 1988.
- [11] PHILLIPS, L., and LUBELL, J., *An SGML Environment for STEP*, NISTIR 5515, National Institute of Standards and Technology, Gaithersburg, MD 20899. November 1994.
- [12] WILSON, P. R., *LTX2X: A LaTeX to X Auto-tagger*, NISTIR, National Institute of Standards and Technology, Gaithersburg, MD 20899. June 1996.
- [13] *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.
- [14] *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.
- [15] RESSLER, S., *The National PDES Testbed Mail Server User's Guide*, NSTIR 4508, National Institute of Standards and Technology, Gaithersburg, MD 20899. January 1991.

[16] RINAUDOT, G. R., *STEP On Line Information Service (SOLIS)*, NISTIR 5511, National Institute of Standards and Technology, Gaithersburg, MD 20899. October 1994.

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

Index

\-	18
_	18
\aamdefhead	26
\aamfighead	26
\aamfigrange	34, 58
\aamfigures	34, 58
\aamhead	26
\abstract	5
\address	5
\aepurposehead	38
\aepurposename	38
\aetbp	58
\aetppb	42
\afunctionhead	14
AIC	3, 26, 53
aic (package)	4, 36
\aicdef	36
\aicexpressg	37, 58
\aicexpressghead	36
\aicextraintro	36
\aicinaptrue	26, 30–31
\aicshortexphead	36
\aicshortexpintro	37
aicv1.sty	4, 57
\aimenthead	38, 58
\aimexplisting	59
\aimexpressg	35
\aimexpressghead	26
\aimexpresshead	26
\aimhead	26
\aimlongexp	32
\aimlongexphead	26
\aimpurposehead	38
\aimpurposename	38
\aimshortexphead	26
\aimshortnameshead	26
\aimtbp	58
\aimtpbp	43
\altaddress	6
\altemail	6
\altfax	6, 57
\altowner	6
\altprop	59
\alttelephone	6
\anentityhead	14

AP	2–3, 26, 52
ap (package)	4, 26, 49
\apaamdefs	34
\apaamintro	34
\apapplobj	29
\apassert	29
\apasserthead	38
\apassertname	38
\apconformance	32
apconformclasses (environment)	32
apendint.tex	49
\apexpurls	35, 58
\apimpreq	33
\apinforeq	28
\apintroend	26, 49
\apmappingtable	30, 49
apmptbl.tex	49
\APnumber	38, 41–47
\apobjhead	38
\apobjname	38
\applasserthead	26
Application Protocol	2
\applobjhead	26
\apschemareg	23
\apscope	28
\apshortexpress	31
\apshortnames	33
\APTtitle	26, 29, 38, 41
apuof (environment)	28
\apusagehead	26
apv12.sty	4, 57
arglist (environment)	19
\armfigures	35
\armhead	26
\arulehead	14
\aschemaidhead	14
\atcbp	46
\atcbpii	46, 58
\atchead	38
\atcname	38
\atctitlehead	38, 58
ATS	54
ats (package)	4, 38, 49
\atsconstraints	40, 58
\atscovered	40, 58
\atsexecution	40, 58

<code>\atsextra</code>	40, 58
<code>\atsgtpvcbp</code>	44
<code>\atsinput</code>	40, 58
<code>\atsintroendbp</code>	40
<code>\atsnoclassesbp</code>	47
<code>\atspurpobp</code>	42, 58
<code>\atsscopebp</code>	41
<code>\atssummary</code>	40, 58
<code>atsv11.sty</code>	4, 57
<code>\atsverdict</code>	40, 58
<code>attrlist</code> (environment)	18–19
<code>\atypehead</code>	14
<code>\auofhead</code>	26
<code>\B</code>	14
<code>\ballotcycle</code>	5, 57
<code>\BG</code>	15
<code>\bibidefix</code>	17
<code>\bibidefo</code>	17
<code>boilerplate</code>	2, 7, 12, 21
<code>bpf...tex</code>	49
<code>bpfats...tex</code>	49
<code>\brefidefix</code>	17
<code>\brefidefo</code>	17
<code>\CAT</code>	15
<code>\comread</code>	6
<code>\confclassannexhead</code>	38
<code>\confclassannexname</code>	38
<code>\confclassbp</code>	47
<code>\confclasshead</code>	38
<code>\confclassname</code>	38
<code>\confreqhead</code>	26
<code>CTAN</code>	62
<code>\dateec</code>	59
<code>\dateeco</code>	59
<code>\dateint</code>	59
<code>\dateinto</code>	59
<code>\dateproj</code>	59
<code>\dateprojo</code>	59
<code>\datequal</code>	59
<code>\datequalo</code>	59
<code>\datewg</code>	59
<code>\datewgo</code>	59
<code>definitions</code> (environment)	36
<code>description</code> (environment)	18
<code>DIS</code>	3
<code>\docdate</code>	5

<code>\docidhead</code>	14
<code>\docnumber</code>	5
<code>\docreg</code>	22
<code>\doctitle</code>	5, 58
<code>\domainpurposehead</code>	38
<code>\domainpurposename</code>	38
<code>draft</code> (option)	16
<code>dtext</code> (environment)	19
<code>\E</code>	14
<code>ecode</code> (environment)	18
<code>eicode</code> (environment)	18, 57
<code>\email</code>	6
<code>\endForeword</code>	7, 49
<code>\entityhead</code>	14
<code>enumlist</code> (environment)	19
<code>espec</code> (environment)	19
<code>\excludepurposehead</code>	38, 58
<code>\excludepurposename</code>	38
<code>excode</code> (environment)	18, 57
<code>EXPRESS</code>	14, 17, 19
<code>\Express</code>	15
<code>EXPRESS-G</code>	25
<code>\ExpressG</code>	15
<code>\expressghead</code>	14
<code>\expresshead</code>	14
<code>\expressi</code>	15, 57
<code>\expressx</code>	15, 57
<code>\expurls</code>	23, 57
<code>\extrefpurposehead</code>	38
<code>\extrefpurposename</code>	38
<code>facility</code>	2
<code>\fax</code>	6, 57
<code>\fcandasubhead</code>	14
<code>\Foreword</code>	7, 58
<code>fproplist</code> (environment)	19
<code>fspec</code> (environment)	19
<code>ftp</code>	62–63
<code>\functionhead</code>	14
<code>\fwddivslst</code>	11, 49
<code>\fwdpartslst</code>	7, 49
<code>\fwdshortlst</code>	12, 57
<code>\gccatcname</code>	38
<code>\GE</code>	15
<code>\generalpurposebp</code>	59
<code>\generalpurposehead</code>	38
<code>\generalpurposename</code>	38

\GT	15
\gtpbp	44, 58
\gtpvthead	38
\gtpvcname	38
\gvatcbp	45
\gvcatchead	38
\gvcpostbp	45
\gvcpostthead	38
\gvcpostname	38
\gvcprebp	45
\gvcprethead	38
\gvcprename	38
\HASH	15
hypertext	62
hyphenat (package)	18
\IEQ	16
\implementpurposehead	38
\implementpurposename	38
\impreqhead	26
\index	61
\INE	15
\INEQ	16
\inforeqhead	26
integrated resource	2, 51
Internet	62
Introduction (environment)	12
\Introduction	58
\introend	13, 57
\introsubhead	14
iproplist (environment)	19
ir (package)	4, 25
\irexplisting	59
\irexpressg	25, 58
\irschemaintro	25
irv12.sty	4, 57
IS	3
IS-REVIEW	3
iso (class)	3–4, 16, 61
ISO Directives	3
ISOD	3–4
\issue	59
\ix	16
\keywords	5
\label	61
L ^A T _E X	1, 3
\LE	15

<code>\listingshead</code>	14
<code>\LT</code>	15
<code>ltx2x</code>	60
<code>majorsublist (environment)</code>	21
<code>\majorsubname</code>	21
<code>\mactablehead</code>	26
<code>\mnote</code>	16
<code>\modelscopehead</code>	14
<code>\NE</code>	15
<code>\nexp</code>	14, 16
<code>\nrefasni</code>	16
<code>\nrefparti</code>	16
<code>\nrefpartxi</code>	16
<code>\nrefpartxii</code>	16, 57
<code>\nrefpartxli</code>	16
<code>\nrefpartxlii</code>	16
<code>\nrefpartxlili</code>	16
<code>\nrefpartxxi</code>	16
<code>\nrefpartxxii</code>	16, 57
<code>\nrefpartxxxii</code>	16, 57
<code>\nrefpartxxxiii</code>	16, 57
<code>\objreghead</code>	14
<code>\olddocnumber</code>	5
<code>\oldprojnumber</code>	58
<code>\oldwg</code>	5
<code>\otherdefhead</code>	14
<code>\otherpurposehead</code>	38, 58
<code>\otherpurposename</code>	38
<code>\owner</code>	5
<code>package file</code>	2
<code>\partialcont</code>	59
<code>\partidefhead</code>	14
<code>\partno</code>	5
<code>\partnumber</code>	58
<code>\picsannex</code>	33
<code>\picshead</code>	14
<code>\pisfbp</code>	47, 58
<code>\posthead</code>	38, 58
<code>\postipfilehead</code>	38, 58
<code>\postipfilename</code>	38
<code>preamble</code>	3, 5, 26, 38, 57–58
<code>\prephead</code>	38, 58
<code>\prepname</code>	38
<code>\primcont</code>	59
<code>\projnumber</code>	58
<code>\purposeshead</code>	38

<code>\purposesname</code>	38
<code>\QUES</code>	15
<code>\refdefhead</code>	14
<code>\renewcommand</code>	3
<code>\renewenvironment</code>	3
<code>rspec</code> (environment)	19
<code>\rulehead</code>	14
<code>\rulepurposehead</code>	59
<code>\schemahead</code>	14
<code>\schemaidhead</code>	14
<code>\schemareg</code>	22
<code>\scivnumber</code>	58
<code>scope</code>	28
<code>SD</code>	3–4, 17, 21, 25–26
<code>\series</code>	5, 57
<code>SGML</code>	60
<code>\shortnamehead</code>	14
<code>\shortnames</code>	22
<code>\singleentityhead</code>	14
<code>\singlefunctionhead</code>	14
<code>\singlerulehead</code>	14
<code>\singletypehead</code>	14
<code>SOLIS</code>	49, 62
<code>sspec</code> (environment)	19
<code>\status</code>	58
<code>step</code> (package)	4–5, 25–26, 36, 38, 49, 61
<code>\STEPcover</code>	5, 57–58
<code>\stepparttitle</code>	13
<code>\steptrid</code>	12, 57
<code>stepv13.sty</code>	4, 57
<code>stpdv1st.tex</code>	49
<code>stppt1st.tex</code>	49
<code>style file</code>	2
<code>Supplementary Directives</code>	3
<code>\techdischead</code>	14
<code>\telephone</code>	5
<code>TeX4ht</code>	61
<code>titles.txt</code>	49
<code>TR</code>	57
<code>tspec</code> (environment)	19
<code>\typehead</code>	14
<code>\uofhead</code>	26
<code>\wg</code>	5
<code>WWW</code>	62–63