Contents

1 Introduction ........................................ 1

2 A driver for this document ......................... 2

3 Identification ....................................... 2

4 Initial Code ......................................... 3

5 The STEP package .................................... 3
   5.1 Preamble commands ............................. 3
   5.2 Indexing style commands .................... 4
   5.3 Miscellaneous commands ..................... 5
      5.3.1 Font changes ............................ 5
      5.3.2 Logos .................................. 5
      5.3.3 EXPRESS code symbols ................ 5
      5.3.4 marginal notes ......................... 6
   5.4 EXPRESS code documentation ................. 6
      5.4.1 environments .......................... 6
      5.4.2 Indexing ................................ 8
   5.5 STEP part title ................................ 9
   5.6 Headings and boilerplate .................... 9
      5.6.1 Foreword elements ...................... 9
      5.6.2 The introduction ....................... 11
      5.6.3 Miscellaneous headings ................. 12
      5.6.4 Miscellaneous boilerplate ............. 14
   5.7 Common references ........................... 16
   5.8 Cover sheet ................................... 18

*This file has version number v1.5, last revised 2002/01/10.
1 Introduction

This document provides the commented source for the LATEX package files designed for the typesetting of documents according to the rules for ISO international standards, and specifically for ISO 10303 Product data representation and exchange commonly referred to as ‘STEP’ (STandard for the Exchange of Product model data). A separate document provides the user manual [Wil96c]. This manual is typeset according to the conventions of the LATEX docstrip utility which enables the automatic extraction of the LATEX package files [GMS94].

ISO (the International Organization for Standardisation) specify their document layout requirements in ISO Directives [ISO97]. Unfortunately these Directives do not completely define the document layout, leaving several aspects open to interpretation by the document editor and re-interpretation by the ISO editorial board. In the case of STEP an additional set of informal ‘Supplementary Directives’ have been established by the ISO TC184/SC4 Editing Committee [Sec97b]. The packages defined herein provide extensions to the general package files [Wil96b, Wil96a] and meet the requirements of both of these Directives. Elsewhere there is a set of package files for the general typesetting of ISO documents [Wil96b, Wil96a].

Some of the STEP standard documents have been published by ISO from camera ready copy derived from electronic sources (this also means that ISO has not objected to the typographical conventions supported by these packages). Within ISO there are proposals to maintain and publish directly from SGML tagged electronic sources. The packages have been designed to simplify the conversion from LATEX to SGML tagging. Thus, there are more document structural elements defined than is usual with LATEX.
As already noted, the macros described later are based on the STEP Supplementary Directives. If in the future the Directives are modified or extended, then it may be necessary to modify or extend the macros. Essentially, this manual is provided as a service for maintainers of the \LaTeX{} packages. It is assumed that any package maintainer is \LaTeX{} literate and accustomed to supporting a \LaTeX{} system [GMS94].

2 A driver for this document

The next bit of code contains the documentation driver file for \LaTeX{}, i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the \texttt{docstrip} program.

1  (*driver)
2 \documentclass{ltxdoc}

   We want an index, using line numbers, but not update information.
3 \EnableCrossrefs
4 \CodelineIndex
5%%% \RecordChanges

We use so many \texttt{docstrip} modules that we set the \texttt{StandardModuleDepth} counter to 1.
6 \setcounter{StandardModuleDepth}{1}

Define some commonly used abbreviations
7 \newcommand*{\Lopt}[1]{\textsf{#1}}
8 \newcommand*{\file}[1]{\texttt{#1}}
9 \newcommand*{\Lcount}[1]{\textsl{\small#1}}
10 \newcommand*{\pstyle}[1]{\textsl{#1}}

We also want the full details printed.
11 \begin{document}
12 \DocInput{stepe.dtx}
13 \PrintIndex
14%%% \PrintChanges
15 \end{document}
16 (/driver)

3 Identification

These packages can only be used with \LaTeX{}2e.
17 (*step | ir | ap | ats | aic | am)

   Announce the Package name and its version:
18 (*step)
19 \ProvidesPackage{stepv13}[2002/01/10 v1.3.2 STEP general package]
20 (/step)
21 (*ir)
The step package is the main documentation style for STEP. Some of the other packages require this to be loaded.

\RequirePackage{stepv13}[2002/01/10]

\section{Initial Code}

In this part we define a few commands that are used later on.

\texttt{\stepemptystring} \smallskip
This is an alias for the \texttt{\isoemptystring} command (for the purposes of upwards compatibility). We use it in testing for an empty parameter.

\section{The STEP package}

This section defines the facilities available in the STEP package.

\subsection{Preamble commands}

The commands defined in this section should, if required, be placed in the document preamble.

\texttt{\partno\{part number\}} specifies the part number for ISO 10303. Internally, it is referred to by \texttt{\thespartno}.

\texttt{\gdef\thespartno{}\newcommand{\partno}[1]{\gdef\thespartno{#1}}}
\series \series{\langle series name \rangle} specifies the particular series name for this Part of ISO 10303. Internally, it is referred to by \theseries.

\Theseries 45 \gdef\theseries{} 46 \gdef\Theseries{} 47 \newcommand{\series}{\gdef\Theseries{#1}} 48 \gdef\theseries{\MakeLowercase{#1}}

\doctitle \doctitle{\langle informal title \rangle} specifies the informal title of the document to be placed on the cover sheet. Internally, it is referred to by \thed@ctitle.

\thed@ctitle 49 \gdef\thed@ctitle{} 50 \newcommand{\doctitle}{\gdef\thed@ctitle{#1}} 51 \newcommand{\st@pn@me}{Product data representation and exchange}

\ballotcycle \ballotcycle{\langle ballot cycle number \rangle} specifies the ballot cycle number for the document (i.e., 0, 1, 2, \ldots). The command sets the b@cyc counter appropriately.

\b@cyc 52 \newcounter{b@cyc} 53 \newcommand{\ballotcycle}{\setcounter{b@cyc}{#1}}

\ifanir \ifanir{TRUE if the document is an IR (generic or application).}

\anirfalse 54 \newif\ifanir 55 \anirfalse

\ifhaspatents \ifhaspatents{TRUE if the document has identified patents.}

\haspatentsfalse 57 \newif\ifhaspatents 58 \haspatentsfalse

\ifmapspec \ifmapspec{Set up for use Mapping specification (TRUE) or table (FALSE) in an AP. Initialise to FALSE (i.e., requires no change to an existing AP).}

\mapspecfalse 60 \newif\ifmapspec 61 \mapspecfalse

5.2 Indexing style commands

We make sure that the index style commands are appropriate.

\indexfill \indexfill{Dotted lines between an index entry and the page number.}

\r@newcommand{\indexfill}{\dotfill} 62 \r@newcommand{\ssindexfill}{\dotfill} 63 \r@newcommand{\alphaindexspace}{\dotfill} 64 \r@newcommand{\otherindexspace}{\dotfill}

\alphaindexspace 65 \r@newcommand{\alphaindexspace}{\dotfill} 66 \r@newcommand{\otherindexspace}{\dotfill}
Formatting of \texttt{see} and \texttt{see also}.
\begin{verbatim}
\renewcommand{\indexsee}[1]{\par \hspace*{2em} \textit{see} #1}
\renewcommand{\indexseealso}[1]{\par \hspace*{2em} \textit{see also} #1}
\end{verbatim}

Both print and index a word or phrase.
\begin{verbatim}
\newcommand{\ix}[1]{\#1\index{#1}}
\end{verbatim}

5.3 Miscellaneous commands

5.3.1 Font changes
\begin{verbatim}
\B{⟨text⟩} \texttt{prints} ⟨text⟩ \texttt{in bold while} \E{⟨text⟩} \texttt{prints it \textit{emphasized}.} \BG{⟨mathsymbol⟩} \texttt{prints} ⟨mathsymbol⟩ \texttt{in bold.}
\end{verbatim}

5.3.2 Logos
\begin{verbatim}
\Express\ The commands print the logos for the EXPRESS family of information modeling languages. (Note: In Part 11 the macros were specified as \{	exttt{\small\sc EX-PRESS}\}, etc. but the STEP Editing Committee ignored the wishes of the authors of EX-
\ExpressG PRESS leading to the definitions below.)
\ExpressI \newcommand{\Express}{\textsc{EX-PRESS}}
\ExpressG \newcommand{\ExpressG}{\textsc{EX-PRESS-G}}
\ExpressI \newcommand{\ExpressI}{\textsc{EX-PRESS-I}}
\ExpressX \newcommand{\ExpressX}{\textsc{EX-PRESS-X}}
\end{verbatim}

5.3.3 EXPRESS code symbols
\begin{verbatim}
\nexp \texttt{Highlight an EXPRESS-defined name.}
\nexp \newcommand{\nexp}[1]{\textbf{#1}}
\HASH \texttt{Various symbols used within EXPRESS.}
\LT \newcommand{\HASH}{\texttt{\small \#}}
\LE \newcommand{\LT}{\texttt{\small <}}
\INE \newcommand{\LE}{\texttt{\small <=}}
\GE \newcommand{\INE}{\texttt{\small :<>:}}
\GT \newcommand{\GE}{\texttt{\small >=}}
\end{verbatim}
More EXPRESS symbols.

SD N200 says that EXPRESS reserved words in the text should be written in
smallcaps. Use as \textsc{⟨word⟩}, where ⟨word⟩ is an EXPRESS (-I, -X) word
in any case.

5.3.4 marginal notes

Put a note into the document margin. This is only operative when the draft option
is in effect.

5.4 EXPRESS code documentation

The commands and environments in this section are for documenting EXPRESS
code.

5.4.1 environments

An environment to tag the body of a specification.

Environments for tagging the bodies of entity, function, rule, schema and type
specifications.

An environment to tag descriptive text.
\pbre@k  Internal commands to encourage page breaking before a list heading and discour-

\nopbre@k age after the heading.

\newcommand{\pbre@k}{\pagebreak[2]}
\newcommand{\nopbre@k}{\nopagebreak}

\ehe@d  An internal command for (underlined) headings. \ehe@dmark is required otherwise
\ehe@dmark the title is printed twice!

\newcommand{\ehe@d}{\@startsection{ehe@d}{20}
{\z@} % indent
{\-\baselineskip} % beforeskip
{0.5\baselineskip} % afterskip
{} % normal body text style for heading
\newcounter{ehe@d}
\newcommand{\ehe@dmark}[1]{}

\ehe@d Environment for writing EXPRESS code.
\newenvironment{ehe@d}{%\ehe@d*{\underline{\protect{Express}} specification}:}{\begin{Efont}}{%\end{Efont}

\ehe@d Environment for writing EXPRESS-I code.
\newenvironment{ehe@d}{%\ehe@d*{\underline{\protect{ExpressI}} specification}:}{\begin{Efont}}{%\end{Efont}

\ehe@d Environment for writing EXPRESS-X code.
\newenvironment{ehe@d}{%\ehe@d*{\underline{\protect{ExpressX}} specification}:}{\begin{Efont}}{%\end{Efont}

\expdesc  A non-indented description environment.
\expdesclabel  The label for the description list. Note that it includes a colon.
\newcommand{\expdesclabel}[1]{\bf #1:}
\newenvironment{expdesc}{\list{}%  \setlength{\leftmargin}{\z@}  \setlength{\labelsep}{0.5em}
\setlength{\itemindent}{\labelsep}  \setlength{\labelwidth}{\z@}
\setlength{\itemsep}{\z@  \plus 0.2ex  \minus 0.1ex}
\setlength{\parsep}{0.5\baselineskip}
\let\makelabel\expdesclabel}{\endlist}

\end{document}
attrlist Listing of attribute descriptions.
145 \newenvironment{attrlist}{% 
146 \underline{Attribute definitions}:
147 \begin{expdesc}
148 \end{expdesc}
149
fproplist Listing of formal propositions.
150 \newenvironment{fproplist}{% 
151 \underline{Formal propositions}:
152 \begin{expdesc}
153 \end{expdesc}
154
iproplist Listing of informal propositions.
155 \newenvironment{iproplist}{% 
156 \underline{Informal propositions}:
157 \begin{expdesc}
158 \end{expdesc}
159 enumlist Listing of enumerated items.
160 \newenvironment{enumlist}{% 
161 \underline{Enumerated item definitions}:
162 \begin{expdesc}
163 \end{expdesc}
164 arglist Listing of argument definitions.
165 \newenvironment{arglist}{% 
166 \underline{Argument definitions}:
167 \begin{expdesc}
168 \end{expdesc}
169
5.4.2 Indexing
\ixent Macros for indexing EXPRESS definitions.
\ixenum \newcommand{\ixent}[1]\{\index{#1 (entity)}\}
\ixfun \newcommand{\ixfun}[1]\{\index{#1 (enumeration)}\}
\ixrule \newcommand{\ixrule}[1]\{\index{#1 (rule)}\}
\ixsc \newcommand{\ixsc}[1]\{\index{#1 (subtype\_constraint)}\}
\ixschema \newcommand{\ixschema}[1]\{\index{#1 (schema)}\}
\ixselect \newcommand{\ixselect}[1]\{\index{#1 (select)}\}
\ixtype \newcommand{\ixtype}[1]\{\index{#1 (type)}\}
5.5 **STEP part title**

A special title command for STEP parts.

\stepparttitle{\langle Part title \rangle}

It is implemented in the same manner as the general ISO \title command but using specific title wording.

\gdef\thestepparttitle{}

\newcommand{\scivm@in}{Industrial automation systems and integration --- \newline}
\newcommand{\stepc@mp}{Product data representation and exchange --- \newline}
\newcommand{\thisp@rtno}{\thespartno \#1 \newline}
\newcommand{\sptitle}[1]{#1 \par}
\newcommand{\stepparttitle}[1]{
\cleardoublepage\pagenumbering{arabic}
\%\% \setcounter{section}{0}
\%\% \setcounter{clause}{0}
\ifotherdoc \else
\protect\thispagestyle{isotitlehead}
\fi
\gdef\thestepparttitle{\Tfont\bf \scivm@in \stepc@mp
\thisp@rtno{\thespartno} \sptitle{#1}}
\if@twocolumn
\twocolumn[\vspace*{2\baselineskip}\vbox to 35mm{\thestepparttitle}]
\else
\vspace*{2\baselineskip}\vbox to 35mm{\thestepparttitle}
\fi}

5.6 **Headings and boilerplate**

There are certain elements within a standard that are predetermined.

5.6.1 **Foreword elements**

This command introduces the Foreword for ISO 10303.

\newcommand{\Foreword}{\%
\begin{foreword}
\input{isofwdbp}
\fwdbp
\ifhaspatents\else\fwdnopatents\fi
\iftechspec
ISO/TS\thespartno\newline
\else
ISO/PAS\thespartno\newline
\fi
\ifpaspec
ISO/PAS\thespartno\newline
\else
ISO\thespartno\newline
\fi
\fi}

The command for ending the STEP Foreword. Use as:

```
\def\endForeword#1#2{
\par
A complete list of parts of ISO 10303 is available from the Internet:\
\centerline{\url{http://www.nist.gov/sc4/editing/step/titles/}}
\par
\% Don’t talk about annexes if relevant argument is empty.
\if\stepemptystring{#1} \else\%
\#1 a normative part of this part of ISO 10303. \fi%
\% an integral part of this part of ISO 10303. \fi%
\if\stepemptystring{#2} \else\%
\#2 for information only. \fi
\end{foreword}
```

Boilerplate for the foreword describing the creators of a TR.

```
\newcommand{\steptrid}{%
ISO/TR 10303--\thespartno, 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*.
}%
```

These commands typeset the list of STEP parts and the list of STEP documentation divisions, respectively.

```
\newcommand{\fwdshortlist}{\input{stppdlst}}
```

This International Standard is organized as a series of parts, each published separately. The structure of this International
Standard is described in ISO-10303-1.

Each part of this International Standard is a member of one of the following series:
- description methods,
- implementation methods,
- conformance testing methodology and framework,
- integrated generic resources,
- integrated application resources,
- application protocols,
- abstract test suites,
- application interpreted constructs,
- application modules.

This part is a member of the \textit{series}{} series.
\texttt{\textbackslash ifanir}\ The integrated generic resources and the integrated application resources specify a single conceptual product data model.\texttt{\textbackslash fi}

\texttt{\textbackslash fwd4)
\texttt{\textasteriskcentered bpfs1}
\texttt{\textbackslash ProvidesFile{bpfs1.tex}[2001/07/16 STEP Intro boilerplate]}
\texttt{\textbackslash typeout{bpfs1.tex [2001/07/16 STEP Intro boilerplate]}}

ISO 10303 is an International Standard for the computer-interpretable representation of product information and for the exchange of product data. The objective is to provide a neutral mechanism capable of describing products throughout their life cycle. This mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases, and as a basis for archiving.
This environment provides boilerplate text and an itemized listing for major subdivisions of the standard.

```latex
\newenvironment{majorsublist}{% 
  \majorsubname \begin{itemize}\end{itemize}}{}
```

Boilerplate for introduction to major subdivision listing.

```latex
\newcommand{\majorsubname}{% 
  Major subdivisions of this part of ISO~10303 are:}
```

### 5.6.3 Miscellaneous headings

Here we define the commands to produce `standard` clause headings, and in some cases the introductory boilerplate. Some of these are general in nature while others are specific to IR parts.

- **\partidefhead** Starts a ‘Terms defined in ISO 10303-1’ subclause
  ```latex
  \newcommand{\partidefhead}{\sclause{Terms defined in ISO~10303-1}}
  ```

- **\refdefhead** Starts a ‘Terms defined in ’ subclause
  ```latex
  \newcommand{\refdefhead}[1]{\sclause{Terms defined in #1}}
  ```

- **\otherdefhead** Starts a ‘Other definitions’ subclause
  ```latex
  \newcommand{\otherdefhead}{\sclause{Other terms and definitions}}
  ```

- **\schemahead** Identification of a clause describing an EXPRESS schema, and the introductory boilerplate.
  ```latex
  \let\schemahead=\clause 
  \newcommand{\schemaintro}[1]{% 
  The following \Express{} declaration begins the \nexp{#1}
  and identifies the necessary external references.\par}
  ```

- **\introsubhead** Starts an ‘Introduction’ subclause.
  ```latex
  \newcommand{\introsubhead}{\sclause{\introductionname}}
  ```

- **\fcandasubhead** Starts a ‘Fundamental concepts and assumptions’ subclause.
  ```latex
  \newcommand{\fcandasubhead}{\sclause{\fcandaname}}
  ```

- **\singletypehead** Starts a ‘type definition’ or ‘type definitions’ subclause.
  ```latex
  \newcommand{\singletypehead}[2]{\sclause{\#1 type definition: \#2}}
  \newcommand{\typehead}[1]{\sclause{\#1 type definitions}}
  ```
\typehead  Starts a ‘type definition’ subsubclause.

\singleentityhead  \entityhead  Starts an ‘entity definition’ subclause or an ‘entity definitions’ subclause. Use the latter as:
\entityhead\{(schema)\}{⟨group⟩} where (schema) is the name of the schema and ⟨group⟩ is a possibly blank grouping identifier.

\singlefunctionhead  \functionhead  Starts a ‘function definition’ or a ‘function definitions’ subclause.

\shortnamehead  Starts a ‘Short names of entities’ normative annex

\objreghead  Starts a ‘Information object registration’ normative annex.

\docidhead  Starts a ‘Document identification’ subclause.

\schemidhead  Starts a ‘Schema identification’ subclause

\aschemidhead  Starts a ‘Schema identification’ subsubclause
\expresshead \newcommand{\expresshead}{\infannex{EXPRESS listing}}

\listingshead \newcommand{\listingshead}{\infannex{Computer interpretable listings}\label{;scil}}

\expressghead \newcommand{\expressghead}{\infannex{EXPRESS-G diagrams}\label{;seg}}

\picshead \newcommand{\picshead}{\normannex{Protocol Implementation Conformance Statement (PICS) proforma}\label{;spics}}

\techdischead \newcommand{\techdischead}{\infannex{Technical discussions}\label{;std}}

\exampleshead \newcommand{\exampleshead}{\infannex{Examples}\label{;sex}}

5.6.4 Miscellaneous boilerplate

\expressgdef \newcommand{\expressgdef}{\ExpressG{} is defined in annex~D of ISO 10303-11}

\maptableorspec Depending on \ifmapspec, prints either ‘table’ or ‘specification’.
\DeclareRobustCommand{\maptableorspec}{% 
\ifmapspec specification\else table\fi}

\shortnames Boilerplate for Short Name annex.
\newcommand{\shortnames}{\input{bpfir1}}

\ProvidesFile{bpfir1.tex}[1997/09/30 short names annex boilerplate]
\typeout{bpfir1.tex [1997/09/30 short names annex boilerplate]}

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.

Here is the text of file bpfir1.tex.
To provide for unambiguous identification of an information object in an open system, the object identifier
\begin{center}
\{~iso standard 10303 part(\thespartno) version(#1)~\}
\end{center}
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.

To provide for unambiguous identification of the schema-name \#2 in an open information system, the object identifier
\begin{center}
\{~iso standard 10303 part(\thespartno) version(#1) schema(#3) #4(#5)~\}
\end{center}
is assigned to the \texttt{\exp{#2}} schema (see \#6). The meaning of this value is defined in ISO/IEC~8824-1, and is described in ISO~10303-1.

The command \expurls{(short)}{(express)} prints the boilerplate for an annex of short names and EXPRESS schemas, where \texttt{(short)} is the URL of the short names and \texttt{(express)} is the URL of the EXPRESS code.

Here is the text of file \texttt{bpfir2.tex}
5.7 Common references

Many of the STEP parts use the same ‘standard’ references.

These macros specify some standard normative references.

\newcommand{\nrefasni}{\isref{ISO/IEC 8824-1:1998}{Information technology --- Abstract Syntax Notation One (ASN.1): Specification of basic notation.}}
\newcommand{\nrefparti}{\isref{ISO 10303-1:1994}{Industrial automation systems and integration --- Part 1: Overview and fundamental principles.}}
\newcommand{\nrefpartxi}{\isref{ISO 10303-11:1994}{Industrial automation systems and integration --- The EXPRESS language reference manual.}}
\newcommand{\nrefpartxxii}{\disref{ISO 10303-22:---}{Clear text encoding of the exchange structure.}}
Industrial automation systems and integration ---
Product data representation and exchange ---
Part 22: Implementation method:
Standard data access interface specification.}
\newcommand{\nrefpartxxxi}{\isref{ISO 10303-31:1994}{Industrial automation systems and integration ---
Product data representation and exchange ---
Part 31: Conformance testing methodology and framework:
General concepts.}}
\newcommand{\nrefpartxxxii}{\disref{ISO 10303-32:---}{Industrial automation systems and integration ---
Product data representation and exchange ---
Part 32: Conformance testing methodology and framework:
Requirements on testing laboratories and clients.}}
\newcommand{\nrefpartxli}{\isref{ISO 10303-41:1994}{Industrial automation systems and integration ---
Product data representation and exchange ---
Part 41: Integrated generic resources:
Fundamentals of product description and support.}}
\newcommand{\nrefpartxlia}{\isref{ISO 10303-41:2001}{Industrial automation systems and integration ---
Product data representation and exchange ---
Part 41: Integrated generic resources:
Fundamentals of product description and support.}}
\newcommand{\nrefpartxlii}{\isref{ISO 10303-42:1994}{Industrial automation systems and integration ---
Product data representation and exchange ---
Part 42: Integrated generic resources:
Geometric and topological representation.}}
\newcommand{\nrefpartxliia}{\isref{ISO 10303-42:2001}{Industrial automation systems and integration ---
Product data representation and exchange ---
Part 42: Integrated generic resources:
Geometric and topological representation.}}
\newcommand{\nrefpartxliii}{\isref{ISO 10303-43:1994}{Industrial automation systems and integration ---
Product data representation and exchange ---
Part 43: Integrated generic resources:
Representation structures.}}
\newcommand{\nrefpartxliiiia}{\isref{ISO 10303-43:2001}{Industrial automation systems and integration ---
Product data representation and exchange ---
Part 43: Integrated generic resources:
Representation structures.}}
\bibidefo These macros specify some bibliographic references and the associated commands to cite them in the text.
\brefidfo \bibidefix \bibidefix \bibieeidefix \brefidefix
5.8 Cover sheet

STEP documents require a cover sheet for tracking purposes.

First we set up some internal commands depending on the type of ISO document being produced. The information is typically taken from the options used in the ISO class.

\thest@tus holds the ISO suffix indicating the type of ISO document.

\gdef\thest@tus{}
\iffdisstandard \gdef\thest@tus{/FDIS}\fi
\iffdisstandard \gdef\thest@tus{/DIS}\fi
\iffcdstandard \gdef\thest@tus{/CD}\fi
\iffwdstandard \gdef\thest@tus{/WD}\fi
\ifftechrep \gdef\thest@tus{/TR}\fi
\ifftechspec \gdef\thest@tus{/TS}\fi
\iffpaspec \gdef\thest@tus{/PAS}\fi
The cover sheet is typeset by clever use of the \texttt{picture} environment. First define some commands that place text at particular places in a picture.

\begin{itemize}
\item \texttt{\@wg} The Working Group number. Use as \texttt{\@wg\{the\_number\}}.
\item \texttt{\@docnumber} Document number. Use as \texttt{\@docnumber\{1234\}}.
\item \texttt{\@docdate} Document date. Use as \texttt{\@docdate\{yyyy/mm/dd\}}.
\item \texttt{\@oldwg} Developers of the immediately prior version of the document.
\item \texttt{\@olddocnumber} The number of the immediately prior version of the document.
\item \texttt{\@abstract} Document abstract. Use as \texttt{\abstract\{text\}}.
\item \texttt{\@keywords} Document keywords. Use as \texttt{\keywords\{text\}}.
\item \texttt{\@comread} Document comments to the reader. Use as \texttt{\comread\{text\}}.
\end{itemize}
\@owner Contact information for the document’s project leader. Each of these commands takes a single text argument (e.g., \address{(text)}).

\address\newcommand{\@address}{\def\@address{\put(22,40){\parbox[t]{59mm}{#1}}}}
\telephone\newcommand{\@telephone}{\def\@telephone{\put(25,11){#1}}}
\fax\newcommand{\@fax}{\def\@fax{\put(25,6){#1}}}
\email\newcommand{\@email}{\def\@email{\put(22,1){#1}}}

\@altowner Contact information for the document’s editor.

\altowner\newcommand{\@altowner}{\def\@altowner{\put(117.5,45){#1}}}
\altaddress\newcommand{\@altaddress}{\def\@altaddress{\put(104.5,40){\parbox[t]{59mm}{#1}}}}
\alttelephone\newcommand{\@alttelephone}{\def\@alttelephone{\put(107.5,11){#1}}}
\altfax\newcommand{\@altfax}{\def\@altfax{\put(107.5,6){#1}}}
\altemail\newcommand{\@altemail}{\def\@altemail{\put(104.5,1){#1}}}

\STEPcover The cover sheet is implemented by clever use of the picture environment and by using a multitude of internal commands. Use as \STEPcover{(commands)}.

\newcommand{\STEPcover}{%\makebox{}
\ifc@pyright\@copyrighttext\fi\newpage}

\drawcoversheet
Put a copyright notice at the bottom of the next page.

\clearpage
\thispagestyle{startpage}
\mbox{}
\ifc@pyright\@copyrighttext\fi
\newpage}
\drawcoversheet This draws the STEP cover sheet.

\newcommand{\drawcoversheet}{%
\protect\thispagestyle{nohead}
Start the picture. The actual size of the picture is (165,240) but need to fool \LaTeX
into thinking it is smaller so it fits onto a page without complaints. The origin
also needs adjustment to centre it in a reasonable fashion.
\setlength{\unitlength}{1mm}
\begin{picture}(165,200)(0,40) %% actual size is (165,240)
\thicklines
Revision notice for the cover sheet layout.
\put(165,-1){\makebox(0,0)[tr]\tiny revision 8, 1/02 (PRW)}
Project leader information. (Box at y=0, height 50)
\put(0,0){\framebox(82.5,50){}}
\put(2,1){\bf E-mail:}
\put(2,6){\bf Facsimile:}
\put(2,11){\bf Telephone:}
\put(2,40){\bf Address:}
\put(2,45){\bf Project Leader:}
Document editor information.
\put(82.5,0){\framebox(82.5,50){}}
\put(84.5,1){\bf E-mail:}
\put(84.5,6){\bf Facsimile:}
\put(84.5,11){\bf Telephone:}
\put(84.5,40){\bf Address:}
\put(84.5,45){\bf Project Editor:}
Comments to reader box. (Box at y=50, height 35, total height 85)
\put(0,50){\framebox(165,25){}}
%%% \put(0,80){\large \bf COMMENTS TO READER:}
%%% \put(0,85){\framebox(165,45){}}
%%% \put(2,87){\large \bf KEYWORDS:}
%%% \put(2,115){\large \bf ABSTRACT:}
\put(0,85){\framebox(165,35){}}
\put(2,87){\large \bf KEYWORDS:}
\put(2,125){\large \bf ABSTRACT:}
\put(2,70){\large \bf COMMENTS TO READER:}
Draw abstract and keyword headings. (Box at y=85, height 35, total 120)
\put(0,85){\framebox(165,35){}}
\put(2,87){\large \bf KEYWORDS:}
\put(2,115){\large \bf ABSTRACT:}
%%% \put(0,85){\framebox(165,45){}}
%%% \put(2,87){\large \bf KEYWORDS:}
%%% \put(2,125){\large \bf ABSTRACT:}
Do the copyright element. (Box at y=120, height 80, total 200)
\put(0,120){\framebox(165,80)[t]{}%}
%%% \put(0,130){\framebox(165,70)[t]{}%}
%%% \ifc@pyrighthotopt
\begin{minipage}{161mm}
\ifisstandard
22
\input{bpfs2} \%\ unknown at present
\fi
\iffdisstandard
\input{bpfs2}
\fi
\ifdisstandard
\input{bpfs2}
\fi
\ifcdstandard
\input{bpfs3}
\fi
\ifwdstandard
\input{bpfs3}
\fi
\iftechrep
\input{bpfs3} \%\ unknown at present
\fi
\end{minipage}
\else
%%% \put(2,195)\{{\large\bf COPYRIGHT NOTICE:}}
{\vspace*{\baselineskip}
\textbf{\large\space COPYRIGHT NOTICE}\hfill\vspace*{\fill}}
\fi}

Draw the STEP title. (y=215 and 210)
\put(0,215)\{%\n\ifnum\value{b@cyc} < 2
{\bf ISO\thest@tus\ 10303-\thespartno}
\else
{\bf ISO\thest@tus\ 10303-\thespartno.\theb@cyc}
\fi\}
\put(0,210)\{{\begin{minipage}\[t\]{165mm}
{\bf \st@pn@me: \Theseries: \thed@ctitle}
\end{minipage}}\%

Identify the slots for the superseded document information.
\put(0,227)\{{\bf Supersedes ISO TC 184/SC4/}\ % (y=227)
\put(67,226)\{{\line(1,0)(5)}\}
\put(73,227)\{{\bf N}\}
\put(78,226)\{{\line(1,0)(8)}\}

Draw the heading block
\put(0,235)\{{\Large{\bf ISO TC 184/SC4/}}\ % (y=235)
\put(58,234)\{{\line(1,0)(7)}\}
\put(67,235)\{{\Large{\bf N}}\}
\put(72,234)\{{\line(1,0)(11)}\}

Identify the date slot.
\put(135,235)\{{\bf Date:}
Finish off the picture. Note that this is where all the specific drawing commands are called.

\@wg \@docnumber \@docdate \@oldwg \@olddocnumber
\@abstract \@keywords \@comread
\@owner \@address \@telephone \@fax \@email
\@altowner \@altaddress \@alttelephone \@altfax \@altemail
\end{picture}
\setlength{\unitlength}{1pt}

Force printing of cover sheet, and remove the STEPcover internal commands as they are no longer needed.
\clearpage
\undef@covercmds

At last, this is the end of the definition of the \drawcoversheet command.
\}

\undef@covercmds Make the \STEPcover internal commands undefined to make space for later macros, if necessary.
\newcommand{\undef@covercmds}{%
\let\@wg\relax \let\wg\relax
\let\@docnumber\relax \let\docnumber\relax
\let\@docdate\relax \let\docdate\relax
\let\@oldwg\relax \let\oldwg\relax
\let\@olddocnumber\relax \let\olddocnumber\relax
\let\@abstract\relax \let\abstract\relax
\let\@keywords\relax \let\keywords\relax
\let\@comread\relax \let\comread\relax
\let\@owner\relax \let\owner\relax
\let\@address\relax \let\address\relax
\let\@telephone\relax \let\telephone\relax
\let\@fax\relax \let\fax\relax
\let\@email\relax \let\email\relax
\let\@altowner\relax \let\altowner\relax
\let\@altaddress\relax \let\altaddress\relax
\let\@alttelephone\relax \let\alttelephone\relax
\let\@altfax\relax \let\altfax\relax
\let\@altemail\relax \let\altemail\relax
%
\}

Here is the text of the file bpfs2.tex.

\ProvidesFile{bpfs2.tex}[2002/01/10 STEP cover DIS+ copyright boilerplate]
\typeout{bpfs2.tex [2002/01/10 STEP cover DIS+ copyright boilerplate]}
\vspace*{\baselineskip}
\textbf{\large COPYRIGHT NOTICE}
This ISO document is a Final Draft if disstandard, a Draft else.

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

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

ISO copyright office
Case postale 56. CH-1211 Geneva 20
Tel. +41 22 749 01 11
Fax +41 22 734 01 79
E-mail copyright@iso.ch

Reproduction for sales purposes for any of the above-mentioned documents may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

\begin{small}
\begin{center}
ISO copyright office
Case postale 56. CH-1211 Geneva 20
Tel. +41 22 749 01 11
Fax +41 22 734 01 79
E-mail copyright@iso.ch
\end{center}
\end{small}
6 The Integrated Resources package

This section defines the content of the package designed for use in documenting STEP Integrated Resources.
6.1 Boilerplate

This section defines the commands used to print boilerplate text.

The diagrams in this annex correspond to the Express{} schemas specified in this part of ISO~10303. The diagrams use the ExpressG{} graphical notation for the Express{} language. \expressgdef.

The end of this package.

7 The Application Protocol package

This section defines the content of the package designed for use in documenting STEP Application Protocols.

\settocdepth{sclause}

7.1 Preamble commands

These commands, if used, should be placed in the document preamble.

\aptitle{⟨title of AP⟩} — the AP title to be used in running text.

\ifacinap

Set up for use of AIC’s in the AP. Initialize to no AIC used.
\ifmaptemplate Set up for use Mapping Template (TRUE). Initialise to FALSE (i.e., requires no change to an existing AP).
814 \newif\ifmaptemplate
815 \maptemplatefalse

\ifidefix Set up for using IDEF1X as the ARM graphical form (TRUE).
816 \newif\ifidefix
817 \idefixfalse
818

7.2 Heading commands
The commands in this section provide for the ‘standard’ clause headings in an AP.

\inforeqhead Starts a ‘Information requirements’ clause. N200 says that subsubclauses of this should be in the ToC.
819 \newcommand{\inforeqhead}{% 820 \settocdepth{ssclause}
821 \clause{Information requirements}\label{;sireq}}

\uofhead Starts a ‘Units of functionality’ subclause
822 \newcommand{\uofhead}{% 823 \sclause{Units of functionality}\label{;suof}}

\auofhead Starts a subsubclause for a UoF
824 \newcommand{\auofhead}[1]{\ssclause{#1}}

\applobjhead Starts a ‘Application objects’ subclause. N200 says this should revert to ToC subclause listing.
825 \newcommand{\applobjhead}{% 826 \settocdepth{sclause}
827 \sclause{Application objects}\label{;sao}}

\applasserthead Starts a ‘Application assertions’ subclause
828 \newcommand{\applasserthead}{% 829 \sclause{Application assertions}\label{;saa}}

\aimhead Starts a ‘Application interpreted model’ clause
830 \newcommand{\aimhead}{% 831 \clause{Application interpreted model}\label{;saim}}

\mappinghead Starts a ‘Mapping table’ or ‘Mapping specification’ subclause
832 \newcommand{\mappinghead}{% 833 \sclause{Mapping \maptableorspec}\label{;smap}}

\templateshead Starts a ‘Mapping templates’ subsubclause.
834 \newcommand{\templateshead}{% 835 \ssclause{Mapping templates}\label{;stemps}}
\mapuofhead  Starts a UoF mapping subsubclause.
836 \newcommand{\mapuofhead}[1]{\ssclause{#1}}

\mapobjecthead Starts an application object mapping subsubsubclause.
837 \newcommand{\mapobjecthead}[1]{\ssssclause{#1}}

\mapattribhead Starts an application object attribute mapping subsubsubsubclause.
838 \newcommand{\mapattribhead}[1]{\ssssclause{#1}}

\aimshortexphead Starts a ‘AIM EXPRESS short listing’ subclause
839 \newcommand{\aimshortexphead}{%  
840    \sclause{AIM EXPRESS short listing}\label{;saesl}}

\confreqhead Starts a ‘Conformance requirements’ clause
841 \newcommand{\confreqhead}{%  
842    \clause{Conformance requirements}\label{;scr}}

\aimlongexphead Starts a ‘AIM EXPRESS expanded listing’ normative annex
843 \newcommand{\aimlongexphead}{%  
844    \normannex{AIM EXPRESS expanded listing}\label{;saeel}}

\aimshortnameshead Starts a ‘AIM short names’ normative annex
845 \newcommand{\aimshortnameshead}{%  
846    \normannex{AIM short names}\label{;sasn}}

\impreqhead Starts a ‘Implementation method specific requirements’ normative annex
847 \newcommand{\impreqhead}{%  
848    \normannex{Implementation method specific requirements}\label{;simreq}}

\aamhead Starts a ‘Application activity model’ informative annex
849 \newcommand{\aamhead}{%  
850    \infannex{Application activity model}\label{;saam}}

\aamdefhead Starts a ‘Application activity model definitions and abbreviations’ subclause.
N200 says this should not be in the ToC.
851 \newcommand{\aamdefhead}{%  
852    \settocdepth{clause}  
853    \sclause{Application activity model definitions and abbreviations}}

\aamfighead Starts a ‘Application activity model diagrams’ subclause N200 says this should not be in the ToC.
854 \newcommand{\aamfighead}{%  
855    \settocdepth{clause}  
856    \sclause{Application activity model diagrams}}

\armhead Starts a ‘Application reference model’ informative annex
857 \newcommand{\armhead}{%  
858    \settocdepth{clause}  
859    \infannex{Application reference model}\label{;sarm}}
7.2.1 Template headings

\signature The 'mapping signature' heading.
867 \newcommand{\signature}{\ehe@d*{\underline{Mapping signature}:}}

\parameters The 'parameter definitions' heading.
868 \newcommand{\parameters}{\ehe@d*{\underline{Parameter definitions}:}}

\body The 'template body' heading.
869 \newcommand{\body}{\ehe@d*{\underline{Template body}:}}

7.3 Boilerplate printing

\apextraintro Print boilerplate for end of AP introduction clause.
871 \newcommand{\apextraintro}{\input{apendint}}
Here is the text of apendint.tex.
872 (/ap)
873 (+apfl)
874 \ProvidesFile{apendint.tex}[1996/05/31 AP end intro boilerplate]
875 \typeout{apendint.tex [1996/05/31 AP end intro boilerplate]}
876
877 Application protocols provide the basis for developing
878 implementations of ISO°10303 and abstract test suites for
879 the conformance testing of AP implementations.
880
881 Clause\textendash{\ref{;i1}} defines the scope of the application protocol
882 and summarizes the functionality and data covered by the AP.
883 Clause\textendash{\ref{;i3}} lists the words defined in this part of ISO°10303 and
884 gives pointers to words defined elsewhere.
885 An application activity model that is the basis for the definition
886 of the scope is provided in \aref{;saam}. The information requirements
887 of the application are specified in \cref{;sireq} using terminology
888 appropriate to the application. A graphical representation of the
Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in \ref{;smap}, 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 \ref{;saesl}. Note that the definitions and \Express{} provided in 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 \ref{;saesl} contains the complete \Express{} for the AIM without annotation. A graphical representation of the AIM is given in \ref{;saeg}. Additional requirements for specific implementation methods are given in \ref{;simreq}. Additional requirements for specific implementation methods are given in \ref{;simreq}.

\begin{anote} The application activity model in \ref{;saam} provides a graphical representation of the processes and information flows that are the basis for the definition of the scope of this part of ISO\textsuperscript{10303}. \end{anote}
Here is the text for file bpfap2.tex.

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.

\begin{note}
A graphical representation of the information requirements is given in \aref{sarm}.
\end{note}

\begin{note}
The information requirements correspond to those of the activities identified as being within the scope of this application protocol in \aref{saam}.
\end{note}

\begin{note}
The mapping \maptableorspec\ specified in~\ref{smap} shows how the \ifaicinap and application interpreted constructs \fi integrated resources \ifaicinap and application interpreted constructs \fi are used to meet the information requirements of this application protocol. \end{note}

\begin{note}
Print boilerplate for UoF. \begin{apuof}
\begin{itemize}
\end{itemize}
\end{apuof}
\end{note}

Here is the text for file bpfap3.tex.
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 \ref{;sao}.

Here is the text for file bpfap4.tex

This subclause specifies the application objects for the \theap space 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.

Here is the text for file bpfap5.tex

This subclause specifies the application assertions for the \theap space 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.
Here is the contents of the `apmptbl.tex` file.

```latex
⟨/ap⟩ AP: boilerplate;

\ProvidesFile{apmptbl.tex}[2002/01/22 AP mapping table boilerplate]
\typeout{apmptbl.tex [2002/01/22 STEP AP mapping table boilerplate]}

This clause contains the mapping table that shows how each UoF and application object of this part of ISO-10303 (see \cref{;sireq}) maps to one or more AIM constructs (see \aref{;saeel}).

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 \ref{;sao}. 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 lowercase.

Column 2) AIM element: Name of an AIM element as it appears in the AIM (see \aref{;saeel}), 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:
\item \verb|*| : used in conjunction with braces to indicate that any number of relationship entity data types may be assembled in a relationship tree structure.
\end{enumerate}

\apmappingspec
\newcommand{\apmappingspec}{\input{apmpspec}}

Here is the contents of the apmpspec.tex file.

\ProvidesFile{apmpspec.tex}[2001/07/16 AP mapping spec boilerplate]
\typeout{apmpspec.tex [2001/07/16 STEP AP mapping spec boilerplate]}

This clause contains the mapping specification that shows how each UoF and application object of this part of ISO\textsuperscript{10303} (see \cref{;sireq}) maps to one or more AIM constructs (see \aref{;saeel}).

Each mapping specifies up to five elements.

\begin{description}
\item[Application element] The mapping for each application element is specified in a separate subclause below.
\item[AIM element] The name of one or more AIM entity data types (see \aref{;saeel}), the term \texttt{``IDENTICAL MAPPING''}, or the term \texttt{``PATH''}.
\item[Source] For those AIM elements that are interpreted from any common resource, this is the ISO standard number and part number in which the resource is defined.
\end{description}
For those AIM elements that are created for the purpose of this part of ISO\textsuperscript{10303}, this is ‘‘ISO\textsuperscript{10303--}’’ followed by the number of this part.

\item[Rules] One or more global rules may be specified that apply to the population of the AIM entity data types specified as the AIM element or in the 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. A reference to a global rule may be accompanied by a reference to the subclause in which the rule is defined.

\item[Reference path] To describe fully the mapping of an application object, it may be necessary to specify a reference path involving several related AIM elements. Each line in the reference path documents the role of an AIM element relative to the AIM element in the line following 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\textsuperscript{10303}, a reference path to its supertype from an integrated resource is specified. For the expression of reference paths and the relationships between AIM elements the following notational conventions apply:

\begin{itemize}
\item [] enclosed section constrains multiple AIM elements or sections of the reference path are required to satisfy an information requirement;
\item () enclosed section constrains multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;
\item {} enclosed section constrains the reference path to satisfy an information requirement;
\item <> enclosed section constrains at one or more required reference path;
\item [[]] enclosed section constrains the supertype entity;
\item {-->} attribute references the entity or select type given in the following row;
\item {-->} entity or select type is referenced by the attribute in the following row;
\item {[i]} attribute is an aggregation of which a single member is given in the following row;
\item {[n]} attribute is an aggregation of which member \texttt{[n]} is given in the following row;
\item {=>} entity is a supertype of the entity given in the
\end{itemize}
following row;
\item \[\texttt{<=}\] entity is a subtype of the entity given in
the following row;
\item \[\texttt{=}\] the string, select, or enumeration type is
constrained to a choice or value;
\item \[\texttt{/textbackslash}\] the reference path expression continues on
the next line;
\item \[\texttt{*}\] used in conjunction with braces to indicate that
any number of relationship entity data types may be assembled
in a relationship tree structure;
\item \ifmaptemplate
\item \[\texttt{/}\] enclosed section is an application of one of the
mapping templates defined in \ref{;stemps} below;
\fi
\item \[\texttt{-}\] the text following is a comment
(normally a clause reference).
end{itemize}
\end{description}
\langle
\apmmaptemplate
Print boilerplate for start of AP mapping template subsubclause.
\newcommand{\apmmaptemplate}{\input{apmptempl}}
Here is the contents of the \apmptempl.tex file.
\langle
\apmmaptemplate
\ProvidesFile{apmptempl.tex}[2001/07/16 AP mapping template boilerplate]
\typeout{apmptempl.tex \[2001/07/16 STEP AP mapping template boilerplate\]}
\begin{itemize}
\item the template signature that specifies the name of the template
and may also specify the names and the order of the formal parameters
of the template;
\item descriptions of the formal parameters of the template, if any;
\item the template body that defines the reusable portion of a reference
path and may indicate, through the use of the formal parameter
names included in the template signature, the points at which
the value parameters are supplied in each template application.
\end{itemize}
Each mapping template is used at least once in the reference paths specified in \ref{uof1} to \ref{uoflast}. Each such template application is a reference to the template definition, based on the pattern established by the template signature, and supplies the value parameters that are to be substituted for the formal parameters specified in the template definition. The full reference path can be derived by replacing any formal parameters in the template body by the value parameters specified in the template application and then substituting the completed template body for the template application.

The following is an example of a template application that invokes and supplies parameters for the \texttt{GROUPS} mapping template.

\begin{anexample}
\texttt{/GROUPS(shape\_aspect, 'boundary index 1')/}
\end{anexample}

The non-blank characters following the first `/` define the name of the mapping template. The name of the mapping template is given in upper case. The name of the template is followed by a list of parameter values, separated by commas, enclosed in parentheses. Parameter values are given in lower case except in the case that the value parameter is a string literal that includes upper case characters.

The following notational conventions apply to the definitions and applications of templates:

\begin{itemize}
\item `/` marks the beginning and end of a template signature or a template application;
\item `\&` prefixes the name of a formal parameter within the definition of a template body;
\item `{}` enclose the formal parameters in a template signature or the value parameters in a template application;
\item `[,]` separates formal parameters in a template signature or value parameters in a template application;
\item `' ' ` denotes a string literal that is used as a value parameter in a template application.
\end{itemize}

Value parameters that are not enclosed by quotes are \texttt{\Express{}} data type identifiers.

This part of ISO-10303 uses the templates that are specified in the following subclauses.

39
A macro for the boilerplate text for SUBTYPE and SUPERTYPE templates.
\newcommand{\sstemplates}{\input{apsstempl}}

Here is the text for the file `apsstempl.tex`.

\ProvidesFile{apsstempl.tex}[2001/07/16 AP SUP/SUB templates boilerplate]
\typeout{apsstempl.tex [2001/07/16 AP SUP/SUB templates boilerplate]}

\signature
\parameters
application\_object: the application object that is a subtype of the current supertype application object and that has the entire or a part of the mapping specification of this supertype.

The SUBTYPE mapping template specifies a reference to the mapping of a subtype of the current application object. Several such references may be included for one supertype application object.
\begin{anote} This template definition only consists of a template signature, there is no matching template body. The template is included to ease the automatic processing of the mapping specification. \end{anote}

\signature
\parameters
application\_object: the application object that is a supertype of the current application object. Several such references may be included for the subtype application object.
\begin{anote} This template only consists of a signature, there is no matching body. The template is included to ease the automatic processing of the mapping specification. \end{anote}
parameters

application\_object: the application object that is a supertype of the current
subtype application object and that has the entire
or a part of the mapping specification of this
subtype.

\apshortexpress Print boilerplate for AP AIM EXPRESS short listing.
\newcommand{\apshortexpress}{\input{bpfap6}}

Here is the text of file bpfap6.tex

\apconformance Print boilerplate for AP conformance.
\apconformance{\textit{implementation methods}}
\newcommand{\apconformance}[1]{%}
\input{bpfap7}

An implementation shall support at least one of the following
implementation methods: #1.
\input{bpfap8}
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.

Requirements with respect to implementation methods-specific requirements are specified in \ref{simreq}.

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 \ref{spics}.

\begin{apconformclasses}
\item
\end{apconformclasses}

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

\begin{itemize}
\end{itemize}

\end{apconformclasses}

\newenvironment{apconformclasses}{% 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:
\begin{itemize}
\end{itemize}
\end{apconformclasses}

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

\begin{itemize}
\end{itemize}

\end{apconformclasses}

\apshortnames
\newcommand\apshortnames{\input{bpfap9}}
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.

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.

The command \aamfigrange{figure range} stores the figure range for the AAM activity model diagrams. Use as:

\aamfigrange{figure F.1 through F.n}

where F.n is the last of n figures.

Internally, the value of \aamfigrange is kept in \aamfigs which is given an initial value just in case the user forgets to call \aamfigrange. The value of \aamfigs is used in later boilerplate.
\apaamintro  Print boilerplate for AAM annex intro.

\newcommand{\apaamintro}{\input{bpfap11}}

Here is the contents of \texttt{bpfap11.tex}. Note the use of the \texttt{\aamfigrs} command.

\apaamdefs  Print boilerplate for AAM definitions.

\newcommand{\apaamdefs}{\input{bpfap12}}

Here is the text of file \texttt{bpfap12.tex}

\aamfigures  Print boilerplate for AAM figures. \texttt{\aamfigures}

\newcommand{\aamfigures}{\input{bpfap15}}

Here is the contents of \texttt{bpfap15.tex}. 
The application activity model diagrams are given in \aamfigrs. The graphical form of the application activity model is presented in the IDEF0 activity modelling format \brefidefo. Activities and data flows that are out of scope are marked with asterisks.

\armintro Print boilerplate for ARM introduction.
\aimexpressg Print boilerplate for AIM EXPRESS-G.
\apexpurls The command \apexpurls\{⟨short⟩\}\{⟨express⟩\} prints the boilerplate for an AP annex of short names and EXPRESS schemas, where ⟨short⟩ is the URL of the short names and ⟨express⟩ is the URL of the EXPRESS code.
Here is the text of file bpfap13.tex

\ProvidesFile{bpfap13.tex}[2001/07/16 AP short names/EXPRESS listing boilerplate (1)]
\typeout{bpfap13.tex [2001/07/16 AP short names/EXPRESS listing boilerplate (1)]}

This annex provides a listing of the complete \Express{} schema specified in \aref{;saeel} of this part of ISO~10303 without comments or explanatory text. It also provides a listing of the \Express{} entity names and corresponding short names as specified in \aref{;sasn} of this part of ISO~10303. The content of this annex is available in computer-interpretable form and can be found at the following URLs:

\begin{anote}
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.
\end{anote}

\aimlongexp Print boilerplate for AIM EXPRESS expanded listing.
\newcommand{\aimlongexp}{\input{bpfap14}}

Here is the text of file bpfap14.tex

\ProvidesFile{bpfap14.tex}[1997/09/30 AP AIM EXPRESS expanded listing boilerplate]
\typeout{bpfap14.tex [1997/09/30 AP AIM EXPRESS expanded listing boilerplate]}

The following \Express{} is the expanded form of the short form schema given in \aref{;saesl}. In the event of any discrepancy between the short form and this expanded listing, the expanded listing shall be used.

\begin{anote}
[1997/09/30 AP AIM EXPRESS expanded listing boilerplate]
\end{anote}
\apimpreq Print boilerplate for AP requirements on exchange structure.
\apimpreq{\textit{schema name}{}).
\newcommand{\apimpreq}{%}

1583 The implementation method defines what types of exchange
behaviour are required with respect to this part of ISO\textsuperscript{\textregistered}10303.
Conformance to this part of ISO\textsuperscript{\textregistered}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\textsuperscript{\textregistered}10303-21 and in the AIM defined in \aref{;saeel} of this part
of ISO\textsuperscript{\textregistered}10303. The header of the exchange structure shall identify
use of this part of ISO\textsuperscript{\textregistered}10303 by the schema name \#1'.

1592
1593 

The end of this package.

1594 \textbackslash /ap

\section{The Application Interpreted Construct package}
This section defines the contents of the package designed for use in documenting
STEP AICs.
\begin{itemize}
\item If we are in an AIC we are not in an IR.
\end{itemize}
\newcommand{\anirfalse}{

8.1 Heading commands
The commands in this section provide for the specified clause headings in an AIC.
\aicshortexphead Starts an \textquote{EXPRESS short listing} clause
\newcommand{\aicshortexphead}{\clause{EXPRESS short listing}\label{;sesl}}

8.2 Boilerplate commands
\aicextraintro Print boilerplate for an extra AIC paragraph in the Introduction.
\newcommand{\aicextraintro}{%}

This part of ISO\textsuperscript{\textregistered}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.

\aicdef  
Boilerplate for the definition of ‘AIC’. Only to be used within the definitions environment.

\aicshortexpintro  
This environment provides the boilerplate for the introduction to the AIC EXPRESS short listing.

\aicexpressg  
Print boilerplate for AIC EXPRESS-G.
9 The Abstract Test Suite package

This section defines the contents of the package designed for use in documenting
STEP ATSs. The relevant text has been taken from [Sec97a].

If we are in an ATS then we are not in an IR.

9.1 Preamble commands

These commands must be put in the document preamble.

\theAPpartno \APnumber{⟨part number of AP⟩} — the part number (e.g. 203) of the AP of this
ATS. Internally it is referred to as \theAPpartno.

\APtitle{⟨part number of AP⟩}{⟨part number of AP⟩} — the part number (e.g. 203) of the AP of this
ATS. Internally it is referred to as \theAPpartno.

9.2 Keyword commands

The commands defined in this section implement the keywords specified for an
ATS document.

\atssummary These commands produce a set of underlined phrases.
\atscovered \atsinput \atsconstraints \atsverdict \atsexecution \atsextra

9.3 Heading commands

The commands in this section provide for the specified clause headings in an ATS.

\purposesname Command to start a ‘Test purposes’ clause.
\domainpurposehead Command to start a ‘Domain test purposes’ clause.
\aepurposehead Command to start a ‘Application element test purposes’ clause.
\newcommand{\aepurposehead}{\sclause{Application element test purposes}}

\apobjhead Command to start an application object clause. Use as \apobjhead{⟨Application object n⟩}.
\newcommand{\apobjhead}[1]{\ssclause{#1}}

\apasserthead Command to start an ‘Application assertions’ clause.
\newcommand{\apasserthead}{\ssclause{Application assertions}}

\aimpurposehead Command to start a ‘AIM test purposes’ clause.
\newcommand{\aimpurposehead}{\sclause{AIM test purposes}}

\aimenthead Command to start an AIM entity clause. Use as \aimenthead{(aim entity n)}.
\newcommand{\aimenthead}[1]{\ssclause{#1}}

\extrefpurposehead Command to start a ‘External reference test purposes’ clause.
\newcommand{\extrefpurposehead}{\sclause{External reference test purposes}}

\implementpurposehead Command to start a ‘Implementation method test purposes’ clause.
\newcommand{\implementpurposehead}{\sclause{Implementation method test purposes}}

\otherpurposehead Command to start an ‘Other test purposes’ clause.
\newcommand{\otherpurposehead}{\ssclause{Other test purposes}}

\gtpvchead Command to start a ‘General test purposes and verdict criteria’ clause.
\newcommand{\gtpvchead}{\sclause{General test purposes and verdict criteria}}

\generalpurposehead Commands to start a ‘General test purposes’ clause.
\newcommand{\generalpurposehead}{\sclause{General test purposes}}

\gvcatchead Commands to start a ‘General verdict criteria for all test cases’ clause.
\newcommand{\gvcatchead}{\sclause{General verdict criteria for all abstract test cases}}

\gvcprehead Commands to start a ‘General verdict criteria for preprocessor abstract test cases’ clause.
\newcommand{\gvcprehead}{\sclause{General verdict criteria for preprocessor abstract test cases}}

\gvcposthead Commands to start a ‘General verdict criteria for postprocessor abstract test cases’ clause.
\newcommand{\gvcposthead}{\sclause{General verdict criteria for postprocessor abstract test cases}}

\atchead Commands to start a ‘Abstract test cases’ clause.
\newcommand{\atchead}{\clause{Abstract test cases}}


\atctitlehead Command \atctitlehead{⟨title⟩} to start a particular test case clause.
\newcommand{\atctitlehead}[1]{\ssclause{#1}}

\prehead Commands to start a ‘Preprocessor’ clause.
\newcommand{\prehead}{\ssclause{Preprocessor}}

\posthead Command \posthead{⟨title⟩} to start a ‘Postprocessor’ clause.
\newcommand{\posthead}[1]{\ssclause{Postprocessor}}

\confclassannexhead Commands to start a ‘Conformance classes’ annex.
\newcommand{\confclassannexhead}{\normannex{Conformance classes}}

\confclasshead Commands to start a ‘Conformance class N’ clause. Use \confclasshead{⟨number⟩}.
\newcommand{\confclasshead}[1]{\ssclause{Conformance class #1}}

\postipfilehead Command to start a ‘Postprocessor input specification file names’ annex.
\newcommand{\postipfilehead}{\normannex{Postprocessor input specification file names}}

\atsusagehead Command to start an ‘ATS Usage scenarios’ annex.
\newcommand{\atsusagehead}{\infannex{Usage scenarios}}

9.4 Boilerplate printing
\atsintroendbp Print boilerplate for the end of ATS introduction clause.
\newcommand{\atsintroendbp}{% 
\input{bpfats1} 
}

Here is the text of bpfats1.tex.
\ProvidesFile{bpfats1.tex}[2001/07/16 ATS end intro boilerplate]
\typeout{bpfats1.tex [2001/07/16 ATS end intro boilerplate]}

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\textsuperscript{\textsuperscript{\textsuperscript{-10303}} specifies the abstract test suite for ISO 10303-\theAPpartno, application protocol \theAPtitle. The abstract test cases presented here are the basis for conformance testing of implementations of ISO 10303-\theAPpartno. This abstract test suite is made up of two major parts:
\begin{itemize}
\item the test purposes, the specific items to be covered by conformance testing;
\item the set of abstract test cases that meet those test purposes.
\end{itemize}

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:

\begin{itemize}
\item specifying the requirements for input data to be used when testing an implementation of the application protocol;
\item specifying the verdict criteria to be used when evaluating whether the implementation successfully converted the input data to a different form.
\end{itemize}

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.
\atsscopebp The boilerplate for the ATS scope clause.
\newcommand{\atsscopebp}{% 
\input{bpfats2}
\}

Here is the text of bpfats2.tex.

\ProvidesFile{bpfats2.tex}[1997/09/30 ATS scope boilerplate]
\typeout{bpfats2.tex [1997/09/30 ATS scope boilerplate]}

This part of ISO 10303 specifies the abstract test suite to be used in the conformance testing of implementations of ISO 10303-\theAPpartno.

The following are within the scope of this part of ISO 10303:
\begin{itemize}
\item the specification of the test purposes associated with ISO 10303-\theAPpartno;
\item the verdict criteria to be applied during conformance testing of an implementation of ISO 10303-\theAPpartno using ISO 10303-21 or ISO 10303-22;
\begin{anote}
The verdict criteria are used to ascertain whether a test purpose has been satisfactorily met by an implementation under test (IUT) within the context of a given test case.
\end{anote}
\item the abstract test cases to be used as the basis for the executable test cases for conformance testing.
\end{itemize}

The following are outside the scope of this part of ISO 10303:
\begin{itemize}
\item the creation of executable test cases;
\item test specifications for tests other than conformance testing such as interoperability or acceptance testing;
\item other implementation methods.
\end{itemize}

\atspurposebp The boilerplate for the introduction to the Test purposes clause.
\newcommand{\atspurposebp}{% 
This clause specifies the test purposes for this part of ISO 10303.
Clauses 4.1 and 4.2 are describe the source and meaning of test purposes that are derived from the information requirements defined in ISO 10303-\theAPpartno, clause 4, and the AIM \Express{} schema defined in ISO 10303-\theAPpartno, annex A. These test purposes are not repeated in this part of ISO-10303.
However, through reference in a test case each specific element from the application elements of the AIM implicitly requires that the identified element, as specified in the test purpose statement, will be correctly instantiated by the implementation under test.

\aetpbp

Prints the boilerplate for the introduction to the Application element test purposes clause.

And here is the text of file bpfats3.tex.

Application element (AE) test purposes are implicitly derived from the AP information requirements and are not explicitly documented here. AE test purposes apply to the input specifications of both preprocesser and postprocessor test cases. AE test purposes are implicitly derived from the AP information requirements as follows:

- Application objects (see ISO 10303-\theAPpartno, 4.2): a test purpose derived from an application object is a simple statement of the object's name;
- Application object attributes (see ISO 10303-\theAPpartno, 4.2): test purposes derived from application object attributes are statements of the application object name with a specific attribute name;
- Application assertions (see ISO 10303-\theAPpartno, 4.3): test purposes derived from application assertions are statements describing the relationships between two application objects. Application assertion test purposes address the directions of relationships as well as the number (cardinality) of relationships.

They 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 implicitly 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 `\maptableorspec` defined in ISO 10303-\theAPpartno, 5.1.

\par

`(bp fats3)

\texttt{\aimtpbp} A command to print the introductory boilerplate for an AIM test purpose clause.

\newcommand{\aimtpbp}{%
\input{bpfats4}
}

And here is the text of file \texttt{bpfats4.tex}.

\ProvidesFile{bpfats4.tex}[2002/01/23 ATS AIM test purpose intro boilerplate]
\typeout{bpfats4.tex [2002/01/23 ATS AIM test purpose intro boilerplate]}

Test purposes are implicitly derived from the AP AIM \texttt{\Express}, and are not explicitly documented here. AIM test purposes are implicitly derived from the expanded \texttt{\Express} listing contained in annex A of ISO 10303-\theAPpartno as follows:

\begin{itemize}
\item AIM entity data types: a test purpose derived from an AIM entity data type is a simple statement of the entity data type name;
\item AIM entity attributes: a test purpose derived from an AIM entity attribute is a statement of the AIM entity data type with a given attribute.
\end{itemize}

\begin{quotation}
the postprocessor shall accept the input in accordance with the AIM \texttt{\Express} structure corresponding to this test purpose.
\end{quotation}

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 \`\maptableorspec` of the AP. This also implies no violations of any constraints (local rules or global rules) that apply to the AIM element. AIM test purposes apply to the input specifications of postprocessor test cases only.

\par
\atsimtpbp  \atsimtpbp — the boilerplate for the introduction to the Implementation method test purposes clause.

\newcommand{\atsimtpbp}{\input{bpfats14}}

And here is the text of file bpfats14.tex.

The following test purpose is derived from requirements in ISO 10303-21 and applies to preprocessors only.

other1 The IUT correctly encodes the AIM schema name in the exchange structure.

The following test purposes are derived from requirements in ISO 10303-21 and apply to postprocessors only.

other2 The IUT interprets the ISO 10303-21 header section present in the exchange structure.

other3 The IUT interprets the ISO 10303-21 SCOPE and EXPORT constructs present in the exchange structure.

other4 The IUT interprets the ISO 10303-21 user-defined entity constructs present in the exchange structure.

other5 The IUT interprets various representations of numbers present in the exchange structure in accordance with ISO 10303-21.

other6 The IUT interprets various sequences of symbols present in the exchange structure in accordance with ISO 10303-21.

\atsgtpvc  \atsgtpvc — the boilerplate for the introduction to the General test purposes and verdict criteria clause.

\newcommand{\atsgtpvc}{%
Here is the text of file bpfats5.tex.

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.

\input{bpfats6}

Command to print the boilerplate introduction to General test purposes clause.

And here is the text of file bpfats6.tex

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 \maptableorspec{} defined in clause~5 of ISO 10303-\theAPpartno.

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 mapping \maptableorspec{} and the AIM \Express{} long form defined in 5.1 and annex A of ISO 10303-\theAPpartno.

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 mapping \maptableorspec{}}
and the AIM \Express{} long form
defined in 5.1 and annex A of ISO 10303-\theApPartno.
\par
1980 (\slash bpfats6)
1981 (\ast \text{ats})
\par
\gvatcbp Command to print the boilerplate introduction to \textit{General verdict criteria} clause.
\par
\newcommand{\gvatcbp}&view{bpfats7} \par
And here is the text of file \texttt{bpfats7.tex}
\par
\newcommand{\gvatcbp}&&\slash \texttt{bpfats7} \par
1986 \ProvidesFile{bpfats7.tex}[2001/07/16 ATS general verdict criteria boilerplate]
1987 \typeout{bpfats7.tex \ [2001/07/16 ATS general verdict criteria boilerplate]}
\par
The following verdict criteria apply to all abstract test cases contained in this part of ISO 10303:
\par
\textsc{gvcl} The semantics of the input model are preserved in the output of the IUT according to the reference paths specified in the mapping \maptableorspec{} defined in ISO 10303-\theApPartno, clause 5 (g1).
\par
\textsc{gvcl} The following verdict criteria apply to all abstract test cases contained in this part of ISO 10303:
\par
\textsc{gvcl} The output of a preprocessor conforms to the implementation method to which the IUT claims conformance (g2).
\par
\textsc{gvcl} The instances in the output of a preprocessor are encoded according
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 \maptableorspec{} defined in ISO 10303-\theAPpartno, annex A and 5.1 (g5).

\atcbp Commands to print boilerplate for Abstract test cases clause. \atcbp prints the first paragraph.

\atcbpii is for printing the major portion of the boilerplate (paragraphs 3 onwards).
And here is the text of files bpfats10.tex and bpfats11.tex.

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

- **Id column**: contains an identifier for the application object instantiated in a particular row. The identifier may be referenced as the value of an application assertion. The identifier is the lowest-level subclause number from ISO 10303-\theAPpartno, 4.2 where the application element that appears in that row of the table is specified.
- **V column**: specifies whether or not the element in that row of the table is assigned a verdict in this test case. A blank indicates that it is not assigned a verdict in this test case. A '*' indicates that it is assigned a verdict using a derived verdict criteria. The derived verdict criteria determine whether the semantics associated with the application element are preserved in the output of the IUT according to the reference paths specified in the mapping table defined in ISO 10303-\theAPpartno, 5.1. A number in the V column references a specific verdict criterion defined in the verdict criteria section that follows the preprocessor input specification table.
- **Application Elements column**: identifies the particular application element instance that is being defined by the table. For assertions the role is specified in parenthesis.
- **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. A '"$<$not_present$>$' indicates that the application element 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 (C\textless n\textgreater), where ‘n’ is an integer referencing a note that follows the table.

A suggested value may be changed by the test realizer. A mandatory value may not be changed due to rules in \Express, rules in the mapping \maptableorspec, or the requirements of the test purpose being assigned a verdict. Each constrained value references a note labelled C\textless number\textgreater at the end of the preprocessor input model table and may be modified according to specific constraints specified in it.

The postprocessor input specifications are defined using ISO 10303-\theAPpartno. 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 that 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.

\atcpretpc \atcpretpc prints the boilerplate for the Preprocessor Test Purposes Covered subclause.

\newcommand{\atcpretpc}{%\input{bpfats11}}

Here is the text of bpfats11.tex.

In the preprocessor input specification table of this test case, the numbers in the Id column (ignoring the part beyond the decimal point, if any) whose rows are not empty in the V column identify the application objects that are covered by this test case. These Id numbers refer directly to the subclause numbers within ISO 10303-\theAPpartno, 4.2, where the application object is defined.
\atcposttpc \atcposttpc prints boilerplate for the Postprocessor Test Purposes Covered sub-clause.

\newcommand{\atcposttpc}{% 
  \input{bpfats12}
  }

Here is the text of \texttt{bpfats12.tex}.

\input{bpfats12}
\ProvidesFile{bpfats12.tex}[2001/07/16 ATS postprocessor purposes covered boilerplate]
\typeout{bpfats12.tex [2001/07/16 ATS postprocessor purposes covered boilerplate]}

In the postprocessor input specification table of this test case, the numbers in the Id column (ignoring the part beyond the decimal point, if any) whose rows are not empty in the V column identify the application objects that are covered by this test case. These Id numbers refer directly to the subclause numbers within ISO 10303-\the\APpartno, 4.2, where the application object is defined.

\confclassbp \confclassbp{⟨\number\rangle} prints the boilerplate for the start of a Conformance class clause.

\atsnoclassesbp — the boilerplate for the Conformance class annex when the AP has no conformance classes.

\newcommand{\atsnoclassesbp}{% 
  To conform to conformance class #1 of ISO 10303-\the\APpartno, an implementation shall pass executable versions of the following abstract test cases: }

\pisfbp Prints the boilerplate for the start of a Postprocessor input specification file names annex.

\newcommand{\pisfbp}[3]{\par
  This annex references a listing of the postprocessor input specifications for this part of ISO’10303 without comments or other explanatory text. These specifications are documented using ISO 10303-#1. These specifications are available in computer-interpretable form and can be found at the following URL:
  \isourl{#2}}
If there is difficulty accessing this site contact the ISO Central Secretariat or contact the ISO TC184/SC4 Secretariat directly at: \url{sc4sec@cme.nist.gov}.

The postprocessor input specifications for each test case is supplied electronically via the Internet. Table \#3 lists the file name of the postprocessor input specification that is associated with the postprocessor subclause(s) of a test case.

Here is the text of bpfats13.tex.

\ProvidesFile{bpfats13.tex}[2001/07/16 ATS postprocessor annex (B) boilerplate]
\typeout{bpfats13.tex [2001/07/16 ATS postprocessor annex (B) boilerplate]}

This annex references a listing of the postprocessor input specifications for this part of ISO\textsuperscript{10303} without comments or other explanatory text. These specifications are documented using ISO 10303-\texttt{atstempa}. These specifications are available in computer-interpretable form and can be found at the following URL:

\begin{center}
\isourl{atstempb}
\end{center}

If there is difficulty accessing this site contact the ISO Central Secretariat or contact the ISO TC184/SC4 Secretariat directly at: \url{sc4sec@cme.nist.gov}.

The postprocessor input specifications for each test case is supplied electronically via the Internet. Table \texttt{atstempc} lists the file name of the postprocessor input specification that is associated with the postprocessor subclause(s) of a test case.

The end of this package.
References


Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
<th>Code Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>82, 2091</td>
<td>1108</td>
</tr>
<tr>
<td>&amp;</td>
<td>1285</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>76–79</td>
<td></td>
</tr>
<tr>
<td>@abstract</td>
<td>546, 670, 685</td>
<td></td>
</tr>
<tr>
<td>@address</td>
<td>558, 671, 689</td>
<td>1082</td>
</tr>
<tr>
<td>@altaddress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@altfax</td>
<td>571, 672, 694</td>
<td></td>
</tr>
<tr>
<td>@altowner</td>
<td>571, 672, 693</td>
<td></td>
</tr>
<tr>
<td>@alttelephone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@comread</td>
<td>553, 670, 687</td>
<td></td>
</tr>
<tr>
<td>@copyright</td>
<td>590</td>
<td></td>
</tr>
<tr>
<td>@docdate</td>
<td>540, 669, 682</td>
<td></td>
</tr>
<tr>
<td>@docnumber</td>
<td>538, 669, 681</td>
<td></td>
</tr>
<tr>
<td>@email</td>
<td>558, 671, 692</td>
<td></td>
</tr>
</tbody>
</table>