LaTeX Package Files for ISO 10303: Source code*

Peter Wilson
Catholic University of America
Now at peter.r.wilson@boeing.com

2002/01/10

Contents

*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 Committe [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 DOCSTRIP program.

```
⟨*driver⟩
\documentclass{ltxdoc}
```

We want an index, using linenumbers, but not update information.

```
\EnableCrossrefs
\CodelineIndex
%%% \RecordChanges
```

We use so many `docstrip` modules that we set the `StandardModuleDepth` counter to 1.

```
\setcounter{StandardModuleDepth}{1}
```

Define some commonly used abbreviations

```
\newcommand*{\Lopt}[1]{\textsf {#1}}
\newcommand*{\file}[1]{\texttt {#1}}
\newcommand*{\Lcount}[1]{\textsl {\small#1}}
\newcommand*{\pstyle}[1]{\textsl {#1}}
```

We also want the full details printed.

```
\begin{document}
\DocInput{stepe.dtx}
\PrintIndex
%%% \PrintChanges
\end{document}
⟨/driver⟩
```

3 Identification

These packages can only be used with LaTeX2e.

```
⟨*step | ir | ap | ats | aic | am⟩
```

Announce the Package name and its version:

```
⟨*step⟩
\ProvidesPackage{stepv13}[2002/01/10 v1.3.2 STEP general package]
⟨/step⟩
⟨*ir⟩
```

```
\ProvidesPackage{irv12}[2002/01/10 v1.2.2 STEP IR package]
⟨/ir⟩
⟨*ap⟩
\ProvidesPackage{apv12}[2002/01/10 v1.2.2 STEP AP package]
⟨/ap⟩
⟨*ats⟩
\ProvidesPackage{atsv11}[2002/01/10 v1.1.2 STEP ATS package]
⟨/ats⟩
⟨*aic⟩
\ProvidesPackage{aicv1}[2002/01/10 v1.0.2 STEP AIC package]
⟨/aic⟩
⟨*am⟩
\ProvidesPackage{amv1}[2002/01/10 v1.0 STEP AM package]
⟨/am⟩
⟨/step | ir | ap | ats | aic | am⟩
```

The `step` package is the main documentation style for STEP. Some of the other packages require this to be loaded.

```
⟨*ir | ap | ats | aic | am⟩
    \RequirePackage{stepv13}[2002/01/10]

⟨/ir | ap | ats | aic | am⟩
```

4 Initial Code

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

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

```
⟨step⟩ \let\stepemptystring\isoemptystring
```

5 The STEP package

This section defines the facilities available in the STEP package.

```
⟨*step⟩
```

5.1 Preamble commands

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

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

```
\gdef\thespartno{}
\newcommand{\partno}[1]{\gdef\thespartno{#1}}
```

\series \series{⟨*series name*⟩} specifies the particular series name for this Part of ISO
\theseries 10303. Internally, it is referred to by \theseries.
\Theseries

```
\gdef\theseries{}
\gdef\Theseries{}
\newcommand{\series}[1]{\gdef\Theseries{#1}
                        \gdef\theseries{\MakeLowercase{#1}}}
```

\doctitle \doctitle{⟨*informal title*⟩} specifies the informal title of the document to be
\thed@ctitle placed on the cover sheet. Internally, it is referred to by \thed@ctitle.
\st@pn@me

```
\gdef\thed@ctitle{}
\newcommand{\doctitle}[1]{\gdef\thed@ctitle{#1}}
\newcommand{\st@pn@me}{Product data representation and exchange}
```

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

```
\newcounter{b@cyc}
\newcommand{\ballotcycle}[1]{\setcounter{b@cyc}{#1}}
```

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

```
\newif\ifanir
  \anirfalse

```

\ifhaspatents TRUE if the document has identified patents.

```
\newif\ifhaspatents
  \haspatentsfalse

```

\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).

```
\newif\ifmapspec
  \mapspecfalse
```

5.2 Indexing style commands

We make sure that the index style commands are appropriate.

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

```
\renewcommand{\indexfill}{\dotfill}
\renewcommand{\sindexfill}{\dotfill}
```

\ssindexfill

```
\renewcommand{\ssindexfill}{\dotfill}
```

\alphaindexspace No extra vertical spacing between blocks of index entries,
\otherindexspace

```
\renewcommand{\alphaindexspace}[1]{}
\renewcommand{\otherindexspace}[1]{}
```

\indexsee
\indexseealso
Formatting of *see* and *see also*.

```
\renewcommand{\indexsee}[1]{\par \hspace*{2em} {\em see} #1}
\renewcommand{\indexseealso}[1]{\par \hspace*{2em} {\em see also} #1}
```

\ix
Both print and index a word or phrase.

```
\newcommand{\ix}[1]{#1\index{#1}}
```

5.3 Miscellaneous commands

5.3.1 Font changes

\B
\E
\BG
`\B{`⟨*text*⟩`}` prints ⟨*text*⟩ in bold while `\E{`⟨*text*⟩`}` prints it *emphasized*. `\BG{`⟨*mathsymbol*⟩`}` prints ⟨*mathsymbol*⟩ in bold.

```
\newcommand{\B}[1]{{\bf #1}}
\newcommand{\E}[1]{{\em #1}}
\newcommand{\BG}[1]{{\mbox{\boldmath $#1$}}}
```

5.3.2 Logos

\Express
\ExpressG
\ExpressI
\ExpressX
The commands print the logos for the EXPRESS family of information modeling languages. (Note: In Part 11 the macros were specified as `{{\small\sl EX\-PRESS}}`, etc. but the STEP Editing Committee ignored the wishes of the authors of EXPRESS leading to the definitions below.)

```
\newcommand{\Express}{{\sc EX\-PRESS}}
\newcommand{\ExpressG}{{\sc EX\-PRESS-G}}
\newcommand{\ExpressI}{{\sc EX\-PRESS-I}}
\newcommand{\ExpressX}{{\sc EX\-PRESS-X}}
```

5.3.3 EXPRESS code symbols

\nexp
Highlight an EXPRESS-defined name.

```
\newcommand{\nexp}[1]{\textbf{#1}}
```

\HASH
\LT
\LE
\NE
\INE
\GE
\GT
Various symbols used within EXPRESS.

```
\newcommand{\HASH}{\texttt{\small \#}}
\newcommand{\LT}{\texttt{\small <}}
\newcommand{\LE}{\texttt{\small <=}}
\newcommand{\NE}{\texttt{\small <>}}
\newcommand{\INE}{\texttt{\small :<>:}}
\newcommand{\GE}{\texttt{\small >=}}
\newcommand{\GT}{\texttt{\small >}}
```

\CAT More EXPRESS symbols.
\HAT
\QUES
\BS
\IEQ
\INEQ

```
\newcommand{\CAT}{\texttt{\small ||}}
\newcommand{\HAT}{\texttt{\small ^}}
\newcommand{\QUES}{\texttt{\small ?}}
\newcommand{\BS}{\texttt{\small \\}}
\newcommand{\IEQ}{\texttt{\small :=:}}
\newcommand{\INEQ}{\texttt{\small :<>:}}
```

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

```
\newcommand{\xword}[1]{\textsc{\lowercase{#1}}}
```

5.3.4 marginal notes

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

```
\newcommand{\mnote}[1]{\ifdr@ftd@c
                         \marginpar{\raggedright\tiny #1}
                       \fi}
```

5.4 EXPRESS code documentation

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

5.4.1 environments

specific@tion An environment to tag the body of a specification.

```
\newenvironment{specific@tion}[1]{}{}
```

espec Environments for tagging the bodies of entity, function, rule, schema and type
fspec specifications.
rspec
sspec
tspec

```
\newenvironment{espec}[1]{}{}
\newenvironment{fspec}[1]{}{}
\newenvironment{rspec}[1]{}{}
\newenvironment{sspec}[1]{}{}
\newenvironment{tspec}[1]{}{}
```

dtext An environment to tag descriptive text.

```
\newenvironment{dtext}{}{}
```

\pbre@k
\nopbre@k
Internal commands to encourage page breaking before a list heading and discourage after the heading.

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

\ehe@d
\ehe@dmark
An internal command for (underlined) headings. `\ehe@dmark` is required otherwise 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]{}
```

ecode
Environment for writing EXPRESS code.

```
\newenvironment{ecode}{%
     \ehe@d*{{\underline{\protect\Express{} specification}}:}
     \begin{Efont}}%
    {\end{Efont}}
```

eicode
Environment for writing EXPRESS-I code.

```
\newenvironment{eicode}{%
     \ehe@d*{{\underline{\protect\ExpressI{} specification}}:}
     \begin{Efont}}%
    {\end{Efont}}
```

excode
Environment for writing EXPRESS-X code.

```
\newenvironment{excode}{%
     \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}
```

attrlist Listing of attribute descriptions.

```
\newenvironment{attrlist}{%
      \ehe@d*{{\underline{Attribute definitions}}:}
      \begin{expdesc}}%
    {\end{expdesc}}
```

fproplist Listing of formal propositions.

```
\newenvironment{fproplist}{%
      \ehe@d*{{\underline{Formal propositions}}:}
      \begin{expdesc}}%
    {\end{expdesc}}
```

iproplist Listing of informal propositions.

```
\newenvironment{iproplist}{%
      \ehe@d*{{\underline{Informal propositions}}:}
      \begin{expdesc}}%
    {\end{expdesc}}
```

enumlist Listing of enumerated items.

```
\newenvironment{enumlist}{%
      \ehe@d*{{\underline{Enumerated item definitions}}:}
      \begin{expdesc}}%
    {\end{expdesc}}
```

arglist Listing of argument definitions.

```
\newenvironment{arglist}{%
      \ehe@d*{{\underline{Argument definitions}}:}
      \begin{expdesc}}%
    {\end{expdesc}}
```

5.4.2 Indexing

\ixent \ixenum \ixfun \ixproc \ixrule \ixsc \ixschema \ixselect \ixtype Macros for indexing EXPRESS definitions.

```
\newcommand{\ixent}[1]{\index{#1 (entity)}}
\newcommand{\ixenum}[1]{\index{#1 (enumeration)}}
\newcommand{\ixfun}[1]{\index{#1 (function)}}
\newcommand{\ixproc}[1]{\index{#1 (procedure)}}
\newcommand{\ixrule}[1]{\index{#1 (rule)}}
\newcommand{\ixsc}[1]{\index{#1 (subtype\_constraint)}}
\newcommand{\ixschema}[1]{\index{#1 (schema)}}
\newcommand{\ixselect}[1]{\index{#1 (select)}}
\newcommand{\ixtype}[1]{\index{#1 (type)}}
```

5.5 STEP part title

\stepparttitle A special title command for STEP parts.
\stepparttitle{⟨*Part title*⟩}
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}[1]{Part #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

\Foreword This command introduces the Foreword for ISO 10303.

```
\newcommand{\Foreword}{%
    \begin{foreword}
%%%    \input{isofwdbp}
    \fwdbp

    \ifhaspatents\else\fwdnopatents\fi

    \iftechspec
      ISO/TS~10303--\thespartno\
    \else
      \ifpaspec
        ISO/PAS~10303--\thespartno\
      \else
        ISO~10303--\thespartno\
      \fi
```

```
    \fi
    was prepared by Technical Committee
    ISO/TC~184, \textit{Industrial automation systems and integration},
    Subcommittee SC4, \textit{Industrial data}.
}
```

`\endForeword` The command for ending the STEP Foreword. Use as:
`\endForeword{`⟨*normannexes*⟩`}{`⟨*infannexes*⟩`}`

```
\gdef\endForeword#1#2{%
\par
    A complete list of parts of ISO~10303 is available from the Internet:\\
\centerline{\isourl{http://www.nist.gov/sc4/editing/step/titles/}}
\par

% Don't talk about annexes if relevent 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}
}

```

`\steptrid` 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, \textit{Industrial automation systems and integration,}
    Subcommittee SC4, \textit{Industrial data.}

}

```

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

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

```

The following is the contents of the file `stppdlst.tex`. The wording is based on the SD edition 2.

```
⟨/step⟩
⟨*fwd4⟩
\ProvidesFile{stppdlst.tex}[2001/07/16 STEP parts and divisions URL]
\typeout{stppdlst.tex [2001/07/16 STEP parts and divisions URL]}

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


⟨/fwd4⟩
⟨∗step⟩
```

5.6.2 The introduction

`Introduction` Starts a new 'introduction' clause, together with initial STEP boilerplate.

```
\newenvironment{Introduction}{%
\clearpage
\begin{introduction}
\input{bpfs1}

}%
{\end{introduction}}

```

Here is the text maintained in file `bpfs1.tex`.

```
⟨/step⟩
⟨∗bpfs1⟩
\ProvidesFile{bpfs1.tex}[2001/07/16 STEP Intro boilerplate]
\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.
```

```
296
297 ⟨/bpfs1⟩
298 ⟨*step⟩
```

`majorsublist` This environment provides boilerplate text and an itemized listing for major subdivisions of the standard.

```
299 \newenvironment{majorsublist}{%
300 \majorsubname
301 \begin{itemize}}{\end{itemize}}
302
```

`\majorsubname` Boilerplate for introduction to major subdivision listing.

```
303 \newcommand{\majorsubname}{%
304   Major subdivisions of this part of ISO~10303 are:}
305
```

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 defind in ISO 10303-1' subclause

```
306 \newcommand{\partidefhead}{\sclause{Terms defined in ISO~10303-1}}
```

`\refdefhead` Starts a 'Terms defined in ' subclause

```
307 \newcommand{\refdefhead}[1]{\sclause{Terms defined in #1}}
```

`\otherdefhead` Starts a 'Other definitions' subclause

```
308 \newcommand{\otherdefhead}{\sclause{Other terms and definitions}}
```

`\schemahead` `\schemaintro` Identification of a clause describing an EXPRESS schema, and the introductory boilerplate.

```
309 \let\schemahead=\clause
310 \newcommand{\schemaintro}[1]{%
311   The following \Express{} declaration begins the \nexp{#1}
312   and identifies the necessary external references.\par}
313
```

`\introsubhead` Starts an 'Introduction' subclause.

```
314 \newcommand{\introsubhead}{\sclause{\introductionname}}
```

`\fcandasubhead` Starts a 'Fundamental concepts and assumptions' subclause.

```
315 \newcommand{\fcandasubhead}{\sclause{\fcandaname}}
```

`\singletypehead` `\typehead` Starts a 'type definition' or 'type definitions' subclause.

```
316 \newcommand{\singletypehead}[2]{\sclause{#1 type definition: #2}}
317 \newcommand{\typehead}[1]{\sclause{#1 type definitions}}
```

`\atypehead` Starts a 'type definition' subsubclause.

```
\newcommand{\atypehead}[1]{\ssclause{#1}}
```

`\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.

```
\newcommand{\singleentityhead}[2]{\sclause{#1 entity definition: #2}}
\newcommand{\entityhead}[2]{%
  \if\stepemptystring{#2}
     \sclause{#1 entity definitions}
  \else
     \sclause{#1 entity definitions: #2}
  \fi
}
```

`\anentityhead` Starts an 'entity definition' subsubclause.

```
\newcommand{\anentityhead}[1]{\ssclause{#1}}
```

`\singlerulehead` `\rulehead` Starts a 'rule definition' or 'rule definitions' subclause.

```
\newcommand{\singlerulehead}[2]{\sclause{#1 rule definition: #2}}
\newcommand{\rulehead}[1]{\sclause{#1 rule definitions}}
```

`\arulehead` Starts a 'rule definition' subsubclause.

```
\newcommand{\arulehead}[1]{\ssclause{#1}}
```

`\singlefunctionhead` `\functionhead` Starts a 'function definition' or a 'function definitions' subclause.

```
\newcommand{\singlefunctionhead}[2]{\sclause{#1 function definition: #2}}
\newcommand{\functionhead}[1]{\sclause{#1 function definitions}}
```

`\afunctionhead` Starts a 'function definition' subsubclause.

```
\newcommand{\afunctionhead}[1]{\ssclause{#1}}
```

`\shortnamehead` Starts a 'Short names of entities' normative annex

```
\newcommand{\shortnamehead}{\normannex{Short names of entities}\label{;ssne}}
```

`\objreghead` Starts a 'Information object registration' normative annex.

```
\newcommand{\objreghead}{\normannex{Information object registration}\label{;sior}}
```

`\docidhead` Starts a 'Document identification' subclause.

```
\newcommand{\docidhead}{\sclause{Document identification}}
```

`\schemidhead` Starts a 'Schema identification' subclause

```
\newcommand{\schemaidhead}{\sclause{Schema identification}}
```

`\aschemidhead` Starts a 'Schema identification' subsubclause

```
\newcommand{\aschemaidhead}[1]{\ssclause{#1 identification}}
```

`\expresshead` Starts an 'EXPRESS listing' informative annex

```
\newcommand{\expresshead}{\infannex{EXPRESS listing}}
```

`\listingshead` Starts a 'Computer interpretable listings' informative annex.

```
\newcommand{\listingshead}{\infannex{Computer interpretable listings}\lable{;scil}}
```

`\expressghead` Starts a 'EXPRESS-G diagrams' informative annex

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

`\picshead` Starts a 'Protocol Implementation Conformance Statement (PICS) proforma' normative annex

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

`\techdischead` Starts a 'Technical discussions' informative annex.

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

`\exampleshead` Starts an 'Examples' informative annex

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

```

5.6.4 Miscellaneous boilerplate

`\expressgdef` Where EXPRESS-G is defined.

```
\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}}
```

Here is the text of file `bpfir1.tex`.

```
⟨/step⟩
⟨*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.

⟨/bpfir1⟩
⟨*step⟩
```

`\docreg` Boilerplate for document registration annex. Use as:
`\docreg{`⟨*version no*⟩`}`

```
\newcommand{\docreg}[1]{%
  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.
}

```

`\schemareg` Boilerplate for EXPRESS schema registration. Use as:
`\schemareg{`⟨*version no*⟩`}{` ⟨*underscored schema*⟩`}{`⟨*schema no*⟩`}{` ⟨*hyphenated schema*⟩`}{`⟨*schema-name no*⟩`}{` ⟨*clause/annex no*⟩`}`

```
\newcommand{\schemareg}[6]{%
  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 \nexp{#2} schema (see #6). The meaning of this
  value is defined in ISO/IEC~8824-1, and is described in ISO~10303-1.
}

```

`\expurls` The command `\expurls{`⟨*short*⟩`}{`⟨*express*⟩`}` prints the boilerplate for an 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.

```
\newcommand{\expurls}[2]{\input{bpfir2}
  Short names: \isourl{#1} \\
  \Express: \isourl{#2}
  \input{bpfir3}}

```

Here is the text of file `bpfir2.tex`

```
⟨/step⟩
⟨*bpfir2⟩
\ProvidesFile{bpfir2.tex}[2002/01/22 IR short names and EXPRESS annex initial boilerplate]
\typeout{bpfir2.tex [2002/01/22 IR short names and EXPRESS annex initial boilerplate]}

  This annex references a listing of the \Express{} entity data type
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:
```

```
400
401 ⟨/bpfir2⟩
402 %
403 % Here is the text of \file{bpfir3.tex}.
404 % \changes{v1.3}{1999/02/15}{Added file bpfir3.tex}
405 ⟨*bpfir3⟩
406 \ProvidesFile{bpfir3.tex}[1999/02/15 IR short names and EXPRESS annex ending boilerplate]
407 \typeout{bpfir3.tex [1999/02/15 IR short names and EXPRESS annex ending boilerplate]}
408
409     If there is difficulty accessing these sites contact ISO Central
410 Secretariat or contact the ISO TC~184/SC4 Secretariat directly at:
411 \url{sc4sec@cme.nist.gov}.
412
413 \begin{anote}The information provided in computer-interpretable form at
414       the above URLs is informative. The information that is contained
415       in the body of this part of ISO~10303 is normative.
416 \end{anote}
417
418 ⟨/bpfir3⟩
419 ⟨*step⟩
```

5.7 Common references

Many of the STEP parts use the same 'standard' references.

\nrefasni
\nrefparti
\nrefpartxi
\nrefpartxii
\nrefpartxxi
\nrefpartxxii
\nrefpartxxxi
\nrefpartxxxii
\nrefpartxli
\nrefpartxlii
\nrefpartxliii

These macros specify some standard normative references.

```
420 \newcommand{\nrefasni}{\isref{ISO/IEC 8824-1:1998}{%
421           Information technology ---
422           Abstract Syntax Notation One (ASN.1):
423           Specification of basic notation.}}
424 \newcommand{\nrefparti}{\isref{ISO 10303-1:1994}{%
425           Industrial automation systems and integration ---
426           Product data representation and exchange ---
427           Part 1: Overview and fundamental principles.}}
428 \newcommand{\nrefpartxi}{\isref{ISO 10303-11:1994}{%
429           Industrial automation systems and integration ---
430           Product data representation and exchange ---
431           Part 11: Description methods:
432                    The EXPRESS language reference manual.}}
433 \newcommand{\nrefpartxii}{\isref{ISO/TR 10303-12:1997}{%
434           Industrial automation systems and integration ---
435           Product data representation and exchange ---
436           Part 12: Description method:
437                    The EXPRESS-I language reference manual.}}
438 \newcommand{\nrefpartxxi}{\isref{ISO 10303-21:1994}{%
439           Industrial automation systems and integration ---
440           Product data representation and exchange ---
441           Part 21: Implementation methods:
442                    Clear text encoding of the exchange structure.}}
443 \newcommand{\nrefpartxxii}{\disref{ISO 10303-22:---}{%
```

```
444           Industrial automation systems and integration ---
445           Product data representation and exchange ---
446           Part 22: Implementation method:
447                   Standard data access interface specification.}}
448 \newcommand{\nrefpartxxxi}{\isref{ISO 10303-31:1994}{%
449           Industrial automation systems and integration ---
450           Product data representation and exchange ---
451           Part 31: Conformance testing methodology and framework:
452                   General concepts.}}
453 \newcommand{\nrefpartxxxii}{\disref{ISO 10303-32:---}{%
454           Industrial automation systems and integration ---
455           Product data representation and exchange ---
456           Part 32: Conformance testing methodology and framework:
457                   Requirements on testing laboratories and clients.}}
458 \newcommand{\nrefpartxli}{\isref{ISO 10303-41:1994}{%
459           Industrial automation systems and integration ---
460           Product data representation and exchange ---
461           Part 41: Integrated generic resources:
462                   Fundamentals of product description and support.}}
463 \newcommand{\nrefpartxlia}{\isref{ISO 10303-41:2001}{%
464           Industrial automation systems and integration ---
465           Product data representation and exchange ---
466           Part 41: Integrated generic resources:
467                   Fundamentals of product description and support.}}
468 \newcommand{\nrefpartxlii}{\isref{ISO 10303-42:1994}{%
469           Industrial automation systems and integration ---
470           Product data representation and exchange ---
471           Part 42: Integrated generic resources:
472                   Geometric and topological representation.}}
473 \newcommand{\nrefpartxliia}{\isref{ISO 10303-42:2001}{%
474           Industrial automation systems and integration ---
475           Product data representation and exchange ---
476           Part 42: Integrated generic resources:
477                   Geometric and topological representation.}}
478 \newcommand{\nrefpartxliii}{\isref{ISO 10303-43:1994}{%
479           Industrial automation systems and integration ---
480           Product data representation and exchange ---
481           Part 43: Integrated generic resources:
482                   Representation structures.}}
483 \newcommand{\nrefpartxliiia}{\isref{ISO 10303-43:2001}{%
484           Industrial automation systems and integration ---
485           Product data representation and exchange ---
486           Part 43: Integrated generic resources:
487                   Representation structures.}}
488
```

\bibidefo
\brefidfo
\bibidefix
\bibieeeidefix
\brefidefix

These macros specify some bibliographic references and the associated commands to cite them in the text.

```
489 \newcommand{\bibidefo}{\reference{}{%
490           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.}\label{bibidefo}}
\newcommand{\brefidefo}{\bref{bibidefo}}
\newcommand{\bibidefix}{\reference{}{%
            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.}\label{bibidefix}}
\newcommand{\bibieeeidefix}{\reference{IEEE Std 1320.2--1998,}{%
            Standard for Conceptual Modeling Language ---
            Syntax and Semantics for IDEF1X.}{}\label{bibidefix}}
\newcommand{\brefidefix}{\bref{bibidefix}}

```

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

```
\gdef\thest@tus{}
\ifisstandard
  \gdef\thest@tus{}
\fi
\iffdisstandard
  \gdef\thest@tus{/FDIS}
\fi
\ifdisstandard
  \gdef\thest@tus{/DIS}
\fi
\ifcdstandard
  \gdef\thest@tus{/CD}
\fi
\ifwdstandard
  \gdef\thest@tus{/WD}
\fi
\iftechrep
  \gdef\thest@tus{/TR}
\fi
\iftechspec
   \gdef\thest@tus{/TS}
\fi
\ifpaspec
   \gdef\thest@tus{/PAS}
```

```
\fi
\ifotherdoc
  \gdef\thest@tus{}
\fi

```

The cover sheet is typeset by clever use of the `picture` environment. First define some commands that place text at particular places in a picture.

`\@wg` `\wg` The Working Group number. Use as `\wg{WG the_number}`.

```
\newcommand{\@wg}{}
\newcommand{\wg}[1]{\def\@wg{\put(47,235){\Large\textbf{#1}}}}
```

`\@docnumber` `\docnumber` Document number. Use as `\docnumber{1234}`.

```
\newcommand{\@docnumber}{}
\newcommand{\docnumber}[1]{\def\@docnumber{\put(72,235){\Large\textbf{#1}}}}
```

`\@docdate` `\docdate` Document date. Use as `\docdate{yyyy/mm/dd}`.

```
\newcommand{\@docdate}{}
\newcommand{\docdate}[1]{\def\@docdate{\put(148,235){#1}}}
```

`\@oldwg` `\oldwg` Developers of the immediately prior version of the document.

```
\newcommand{\@oldwg}{}
\newcommand{\oldwg}[1]{\def\@oldwg{\put(58,227){\textbf{#1}}}}
```

`\@olddocnumber` `\olddocnumber` The number of the immediately prior version of the document.

```
\newcommand{\@olddocnumber}{}
\newcommand{\olddocnumber}[1]{\def\@olddocnumber{\put(78,227){\textbf{#1}}}}
```

`\@abstract` `\abstract` Document abstract. Use as `\abstract{`⟨*text*⟩`}`.

```
\newcommand{\@abstract}{}
\newcommand{\abstract}[1]{%
  \def\@abstract{\put(2,110){\parbox[t]{161mm}{#1}}}}
%%%  \def\@abstract{\put(2,120){\parbox[t]{161mm}{#1}}}}
```

`\@keywords` `\keywords` Document keywords. Use as `\keywords{`⟨*text*⟩`}`.

```
\newcommand{\@keywords}{}
%%% \newcommand{\keywords}[1]{\def\@keywords{\put(35,77){#1}}}
\newcommand{\keywords}[1]{\def\@keywords{\put(35,87){#1}}}
```

`\@comread` `\comread` Document comments to the reader. Use as `\comread{`⟨*text*⟩`}`.

```
\newcommand{\@comread}{}
\newcommand{\comread}[1]{%
%%%  \def\@comread{\put(2,65){\parbox[t]{161mm}{#1}}}}
  \def\@comread{\put(2,75){\parbox[t]{161mm}{#1}}}}

```

\@owner
\owner
\@address
\address
\@telephone
\telephone
\@fax
\fax
\@email
\email

Contact information for the document's project leader. Each of these commands takes a single text argument (e.g., `\address{`⟨*text*⟩`}`).

```
\newcommand{\@owner}{}
\newcommand{\owner}[1]{\def\@owner{\put(35,45){#1}}}
\newcommand{\@address}{}
\newcommand{\address}[1]{\def\@address{\put(22,40){\parbox[t]{59mm}{#1}}}}
\newcommand{\@telephone}{}
\newcommand{\telephone}[1]{\def\@telephone{\put(25,11){#1}}}
\newcommand{\@fax}{}
%%%\newcommand{\fax}[1]{\def\@fax{\put(30,6){#1}}}
\newcommand{\fax}[1]{\def\@fax{\put(25,6){#1}}}
\newcommand{\@email}{}
%%%\newcommand{\email}[1]{\def\@email{\put(35,1){#1}}}
\newcommand{\email}[1]{\def\@email{\put(22,1){#1}}}

```

\@altowner
\altowner
\@altaddress
\altaddress
\@alttelephone
\alttelephone
\@altfax
\altfax
\@altemail
\altemail

Contact information for the document's editor.

```
\newcommand{\@altowner}{}
\newcommand{\altowner}[1]{\def\@altowner{\put(117.5,45){#1}}}
\newcommand{\@altaddress}{}
\newcommand{\altaddress}[1]{\def\@altaddress{\put(104.5,40){\parbox[t]{59mm}{#1}}}}
\newcommand{\@alttelephone}{}
\newcommand{\alttelephone}[1]{\def\@alttelephone{\put(107.5,11){#1}}}
\newcommand{\@altfax}{}
%%%\newcommand{\altfax}[1]{\def\@altfax{\put(112.5,6){#1}}}
\newcommand{\altfax}[1]{\def\@altfax{\put(107.5,6){#1}}}
\newcommand{\@altemail}{}
%%%\newcommand{\altemail}[1]{\def\@altemail{\put(117.5,1){#1}}}
\newcommand{\altemail}[1]{\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}[1]{%
```

Make sure that the internal commands are picked up.

```
  #1
```

and call the routine to draw the picture.

```
  \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}{%
```

Make sure we have an empty page style.

```
\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(2,70){\large\bf COMMENTS TO READER:}
\put(0,50){\framebox(165,35){}}
\put(2,80){\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@pyrightopt
    \begin{minipage}{161mm}
      \ifisstandard
```

```
        \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){%
  \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`.

```
⟨/step⟩
⟨*bpfs2⟩
\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}
```

```
\begin{small}
This ISO document is
\iffdisstandard
  a Final Draft
\else
  \ifdisstandard
    a Draft
  \else
    an
  \fi
\fi
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:
\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 \texttt{copyright@iso.ch}
\end{center}
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.

\end{small}

⟨/bpfs2⟩
⟨*step⟩

⟨/step⟩
⟨*bpfs3⟩
\ProvidesFile{bpfs3.tex}[2002/01/10 STEP cover WD/CD copyright boilerplate]
\typeout{bpfs3.tex [2002/01/10 STEP cover WD/CD copyright boilerplate]}

\vspace*{\baselineskip}
\textbf{\large COPYRIGHT NOTICE}

```

```
\begin{small}
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 TC 184/SC4 Secretariat's member body)
or to ISO's member body in the
country of the requester:
\begin{center}
Copyright Manager \\
ANSI \\
11 West 42nd Street \\
New York, New York 10036 \\
USA \\
phone: +1--212--642--4900 \\
fax:   +1--212--398--0023
\end{center}
Reproduction for sales purposes may be subject to royalty payments
or a licensing agreement.

Violators may be prosecuted.

\end{small}

⟨/bpfs3⟩
⟨*step⟩
```

`\draftctr` Some boilerplate for 'Comments to Reader'.

```
\newcommand{\draftctr}{Recipients of this draft are invited to submit,
  with their comments, notification of any relevant patent rights of
  which they are aware and to provide supporting documentation. }

```

The end of this package.

```
⟨/step⟩
```

6 The Integrated Resources package

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

```
⟨*ir⟩
```

\anirtrue We are meant to be processing an IR.

```
\anirtrue
```

6.1 Boilerplate

This section defines the commands used to print boilerplate text.

\irexpressg Boilerplate for IR EXPRESS-G annex. Use as:
\irexpressg

```
\newcommand{\irexpressg}{%
  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.

⟨/ir⟩

7 The Application Protocol package

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

⟨∗ap⟩

\anirfalse If we are processing an AP then we are not processing an IR.

```
\anirfalse
```

In general, the ToC should contain subclauses.

```
\settocdepth{sclause}
```

7.1 Preamble commands

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

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

```
\gdef\theap{}
\newcommand{\aptitle}[1]{\gdef\theap{#1}}
```

\ifaicinap Set up for use of AIC's in the AP. Initialize to no AIC used.

```
\newif\ifaicinap
  \aicinapfalse
```

\ifmaptemplate Set up for use Mapping Template (TRUE). Initialise to FALSE (i.e., requires no change to an existing AP).

```
\newif\ifmaptemplate
  \maptemplatefalse
```

\ifidefix Set up for using IDEF1X as the ARM graphical form (TRUE).

```
\newif\ifidefix
  \idefixfalse

```

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.

```
\newcommand{\inforeqhead}{%
  \settocdepth{ssclause}
  \clause{Information requirements}\label{;sireq}}
```

\uofhead Starts a 'Units of functionality' subclause

```
\newcommand{\uofhead}{%
  \sclause{Units of functionality}\label{;suof}}
```

\auofhead Starts a subsubclause for a UoF

```
\newcommand{\auofhead}[1]{\ssclause{#1}}
```

\applobjhead Starts a 'Application objects' subclause. N200 says this should revert to ToC subclause listing.

```
\newcommand{\applobjhead}{%
  \settocdepth{sclause}
  \sclause{Application objects}\label{;sao}}
```

\applasserthead Starts a 'Application assertions' subclause

```
\newcommand{\applasserthead}{%
  \sclause{Application assertions}\label{;saa}}
```

\aimhead Starts a 'Application interpreted model' clause

```
\newcommand{\aimhead}{%
  \clause{Application interpreted model}\label{;saim}}
```

\mappinghead Starts a 'Mapping table' or 'Mapping specification' subclause

```
\newcommand{\mappinghead}{%
  \sclause{Mapping \maptableorspec}\label{;smap}}
```

\templateshead Starts a 'Mapping templates' subsubclause.

```
\newcommand{\templateshead}{%
  \ssclause{Mapping templates}\label{;stemps}}
```

\mapuofhead Starts a UoF mapping subsubclause.

```
\newcommand{\mapuofhead}[1]{\ssclause{#1}}
```

\mapobjecthead Starts an application object mapping subsubsubclause.

```
\newcommand{\mapobjecthead}[1]{\sssclause{#1}}
```

\mapattribhead Starts an application object attribute mapping subsubsubsubclause.

```
\newcommand{\mapattribhead}[1]{\ssssclause{#1}}
```

\aimshortexphead Starts a 'AIM EXPRESS short listing' subclause

```
\newcommand{\aimshortexphead}{%
  \sclause{AIM EXPRESS short listing}\label{;saesl}}
```

\confreqhead Starts a 'Conformance requirements' clause

```
\newcommand{\confreqhead}{%
  \clause{Conformance requirements}\label{;scr}}
```

\aimlongexphead Starts a 'AIM EXPRESS expanded listing' normative annex

```
\newcommand{\aimlongexphead}{%
  \normannex{AIM EXPRESS expanded listing}\label{;saeel}}
```

\aimshortnameshead Starts a 'AIM short names' normative annex

```
\newcommand{\aimshortnameshead}{%
  \normannex{AIM short names}\label{;sasn}}
```

\impreqhead Starts a 'Implementation method specific requirements' normative annex

```
\newcommand{\impreqhead}{%
  \normannex{Implementation method specific requirements}\label{;simreq}}
```

\aamhead Starts a 'Application activity model' informative annex

```
\newcommand{\aamhead}{%
  \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.

```
\newcommand{\aamdefhead}{%
  \settocdepth{clause}
  \sclause{Application activity model definitions and abbreviations}}
```

\aamfighead Starts a 'Application activity model diagrams' subclause N200 says this should not be in the ToC.

```
\newcommand{\aamfighead}{%
  \settocdepth{clause}
  \sclause{Application activity model diagrams}}
```

\armhead Starts a 'Application reference model' informative annex

```
\newcommand{\armhead}{%
  \settocdepth{sclause}
  \infannex{Application reference model}\label{;sarm}}
```

\aimexpressghead Starts a 'AIM EXPRESS-G' informative annex

```
\newcommand{\aimexpressghead}{%
  \infannex{AIM EXPRESS-G}\label{;saeg}}
```

\aimexpresshead Starts a 'AIM EXPRESS listing' informative annex

```
\newcommand{\aimexpresshead}{%
  \infannex{AIM EXPRESS listing}}
```

\apusagehead Starts a 'Application protocol usage guide' informative annex

```
\newcommand{\apusagehead}{%
  \infannex{Application protocol usage guide}\label{;sapug}}

```

7.2.1 Template headings

\signature The 'mapping signature' heading.

```
\newcommand{\signature}{\ehe@d*{\underline{Mapping signature}:}}
```

\parameters The 'parameter definitions' heading.

```
\newcommand{\parameters}{\ehe@d*{\underline{Parameter definitions}:}}
```

\body The 'template body' heading.

```
\newcommand{\body}{\ehe@d*{\underline{Template body}:}}

```

7.3 Boilerplate printing

\apextraintro Print boilerplate for end of AP introduction clause.

```
\newcommand{\apextraintro}{\input{apendint}}
```

Here is the text of `apendint.tex`.

```
⟨/ap⟩
⟨*apf1⟩
\ProvidesFile{apendint.tex}[1996/05/31 AP end intro boilerplate]
\typeout{apendint.tex [1996/05/31 AP end intro boilerplate]}

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

    Clause~\ref{;i1} defines the scope of the application protocol
and summarizes  the functionality and data covered by the AP.
Clause~\ref{;i3} 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 \aref{;saam}. The information requirements
of the application are specified in \cref{;sireq} using terminology
appropriate to the application. A graphical representation of the
```

```
information requirements, referred to as the application reference
model, is given in \aref{;sarm}.

    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 \aref{;saeel}
contains the complete \Express{} for the AIM without annotation. A
graphical representation of the AIM is given in \aref{;saeg}. Additional
requirements for specific implementation methods are given in
\aref{;simreq}.

⟨/apf1⟩
⟨*ap⟩
```

`\apscope` Print boilerplate for start of AP scope clause.
`\apscope{`⟨*application purpose and context*⟩`}`

```
\newcommand{\apscope}[1]{%
  This part of ISO 10303 specifies the use of the integrated
resources necessary for the scope and information requirements
for #1

\input{bpfap1}

}
```

Here is the text for file `bpfap1.tex`

```
⟨/ap⟩
⟨*bpfap1⟩
\ProvidesFile{bpfap1.tex}[2001/07/16 AP start scope clause boilerplate]
\typeout{bpfap1.tex [2001/07/16 AP start scope clause boilerplate]}

\begin{anote}The application activity model in \aref{;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~10303.\end{anote}

⟨/bpfap1⟩
⟨*ap⟩
```

`\apinforeq` Print boilerplate for start of AP clause on information requirements.
`\apinforeq{`⟨*AP purpose*⟩`}`

```
\newcommand{\apinforeq}[1]{%
  This clause specifies the information required for #1

```

```
\input{bpfap2}

}
```

Here is the text for file `bpfap2.tex`.

```
⟨/ap⟩
⟨*bpfap2⟩
\ProvidesFile{bpfap2.tex}[2001/07/16 AP info boilerplate]
\typeout{bpfap2.tex [2001/07/16 AP info boilerplate]}

  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
      integrated  resources
      \ifaicinap and application interpreted constructs \fi
      are used to meet the information requirements of this
      application protocol. \end{note}

⟨/bpfap2⟩
⟨*ap⟩
```

`apuof` Print boilerplate for UoF.
`\begin{apuof}`⟨*UoF list*⟩`\end{apuof}` where ⟨*UoF list*⟩ is a list of UoF names in `\item` format.

```
\newenvironment{apuof}{%
  This subclause specifies the units of functionality for the
\theap\space application protocol. This part of ISO~10303
specifies the following units of functionality:
\begin{itemize}}{%
\end{itemize}

\input{bpfap3}

}
```

Here is the text for file `bpfap3.tex`.

```
⟨/ap⟩
⟨*bpfap3⟩
```

```
\ProvidesFile{bpfap3.tex}[1997/09/30 AP uof boilerplate]
\typeout{bpfap3.tex [1997/09/30 AP uof boilerplate]}

  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}.

⟨/bpfap3⟩
⟨*ap⟩
```

`\apapplobj` Print boilerplate for Application objects.

```
\newcommand{\apapplobj}{\input{bpfap4}}
```

Here is the text for file `bpfap4.tex`

```
⟨/ap⟩
⟨*bpfap4⟩
\ProvidesFile{bpfap4.tex}[1997/09/30 AP application objects boilerplate]
\typeout{bpfap4.tex [1997/09/30 AP application objects boilerplate]}

  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.

⟨/bpfap4⟩
⟨*ap⟩
```

`\apassert` Print boilerplate for AP application assertions subclause.

```
\newcommand{\apassert}{\input{bpfap5}}
```

Here is the text for file `bpfap5.tex`

```
⟨/ap⟩
⟨*bpfap5⟩
\ProvidesFile{bpfap5.tex}[1997/09/30 AP application assertions boilerplate]
\typeout{bpfap5.tex [1997/09/30 AP application assertions boilerplate]}

  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.

⟨/bpfap5⟩
⟨*ap⟩
```

\apmapping Print boilerplate for start of AP mapping table subclause.

```
\newcommand{\apmapping}{%
  \ifmapspec \input{apmpspec} \else \input{apmptbl} \fi}
```

Here is the contents of the `apmptbl.tex` file.

```
⟨/ap⟩ AP: boilerplate;
⟨*apmptbl⟩
\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 lower case.

 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:
\begin{enumerate}
\item \verb|[]| : enclosed section constrains multiple AIM elements
    or sections of the
    reference path are required to satisfy an information
    requirement;
\item \verb|()| : 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 \verb|{}| : enclosed section constrains the reference path
    to satisfy an information requirement;
\item \verb|<>| : enclosed section constrains at one or more
     required reference path;
\item \verb+||+ : enclosed section constrains the supertype entity;
\item \verb|->| : attribute references the entity or select type
    given in the following row;
\item \verb|<-| : entity or select type is referenced by the
     attribute in the following row;
\item \verb|[i]| : attribute is an aggregation of which a
     single member is given in the following row;
\item \verb|[n]| : attribute is an aggregation of which
     member \verb|n| is given in the following row;
\item \verb|=>| : entity is a supertype of the entity given in the
    following row;
\item \verb|<=| : entity is a subtype of the entity given in
    the following row;
\item \verb|=| : the string, select, or enumeration type is
    constrained to a choice or value;
\item \verb|\| : the reference path expression continues on
    the next line;
```

```
\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}

⟨/apmptbl⟩
⟨*ap⟩
```

\apmappingspec Print boilerplate for start of AP mapping specification subclause.

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

Here is the contents of the apmpspec.tex file.

```
⟨/ap⟩
⟨*apmpspec⟩
\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~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 seperate subclause below.
    Application object names are given in title case.
    Attribute names and assertions are listed after the application
    object to which they belong and are given in lower case.

\item[AIM element] The name of one or more AIM entity data types
    (see \aref{;saeel}), the term ``IDENTICAL MAPPING'',
    or the term ``PATH''.
    AIM entity data type names are given in lower case.
    Attributes of AIM entity data types are referred to as
    $<$entity name$>$.$<$attribute name$>$.
    The mapping of an application element may involve more than
    one AIM element.
    Each of these AIM elements is presented on a seperate line
    in the mapping specification.
    The term ``IDENTICAL MAPPING'' indicates that both application
    objects involved in an application assertion map to the same
    instance of an AIM entity data type.
    The term ``PATH'' indicates that the application assertion maps
    to a collection of related AIM entity instances specified
    by the entire reference 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.
```

```
    For those AIM elements that are created for the purpose of this part
    of ISO~10303, this is ``ISO~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~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[\texttt{[]}] enclosed section constrains multiple AIM elements
    or sections of the
    reference path are required to satisfy an information
    requirement;
\item[\texttt{()}] 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[\texttt{\{\}}]  enclosed section constrains the reference path
    to satisfy an information requirement;
\item[\texttt{<>}]  enclosed section constrains at one or more
     required reference path;
\item[\texttt{||}]  enclosed section constrains the supertype entity;
\item[\texttt{->}]  attribute references the entity or select type
    given in the following row;
\item[\texttt{<-}]  entity or select type is referenced by the
     attribute in the following row;
\item[\texttt{[i]}]  attribute is an aggregation of which a
     single member is given in the following row;
\item[\texttt{[n]}]  attribute is an aggregation of which
     member \texttt{n} is given in the following row;
\item[\texttt{=>}]  entity is a supertype of the entity given in the
```

```
    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;
\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}
```

⟨/apmpspec⟩
⟨*ap⟩

`\apmaptemplate` Print boilerplate for start of AP mapping template subsubclause.

```
\newcommand{\apmaptemplate}{\input{apmptempl}}
```

Here is the contents of the `apmptempl.tex` file.

⟨/ap⟩
⟨*apmptempl⟩

```
\ProvidesFile{apmptempl.tex}[2001/07/16 AP mapping template boilerplate]
\typeout{apmptempl.tex [2001/07/16 STEP AP mapping template boilerplate]}

  This mapping specification includes mapping templates.
A mapping template is a reusable portion of a reference path that defines
a commonly used part of the structure of the application interpreted model.
A mapping template is similar to a programming language macro.
The mapping templates used in this part of ISO~10303 are defined in this
subclause. Each mapping template definition has three components as follows:
\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}
```

```
1252
1253     Each mapping template is used at least once in the reference paths
1254 specified in~\ref{;uof1} to~\ref{;uoflast}.
1255 Each such template application is a reference to the template definition,
1256 based on the pattern established by the template signature, and supplies
1257 the value parameters that are to be substitued for the formal parameters
1258 specified in the template definition. The full reference path can be derived
1259 by replacing any formal parameters in the template body by the value
1260 parameters specified in the template application and then substituting
1261 the completed template body for the template application.
1262
1263 %%\begin{anexample}
1264 %%The following is an example of a template application that invokes and
1265 %%supplies parameters for the GROUPS mapping template.
1266 %%
1267 %%/GROUPS(shape\_aspect, 'boundary index 1')/
1268 %%
1269 %%\end{anexample}
1270
1271     The non-blank characters following the first '/' define the name of
1272 the mapping template. The name of the mapping template is given in
1273 upper case. The name of the template is followed by a list of parameter
1274 values, seperated by commas, enclosed in parentheses. Parameter values
1275 are given in lower case except in the case that the value parameter
1276 is a string literal that includes upper case characters.
1277
1278     The following notational conventions apply to the definitions and
1279 applications of templates:
1280
1281 \begin{itemize}
1282
1283 \item[\texttt{/}] marks the beginning and end of a template signature or a
1284          template application;
1285 \item[\texttt{\&}] prefixes the name of a formal parameter within the definition
1286           of a template body;
1287 \item[\texttt{()}] enclose the formal parameters in a template signature or the
1288           value parameters in a template application;
1289 \item[\texttt{,}] separates formal parameters in a template signature or
1290           value parameters in a template application;
1291 \item[\texttt{' '}] denotes a string literal that is used as a value parameter
1292           in a template application.
1293
1294 \end{itemize}
1295
1296     Value parameters that are not enclosed by quotes are \Express{} data type
1297 identifiers.
1298
1299     This part of ISO~10303 uses the templates that are specified in the
1300 following subclauses.
1301
```

```
⟨/apmptempl⟩
⟨*ap⟩
```

\sstemplates A macro for the boilerplate text for SUBTYPE and SUPERTYPE templates.

```
\newcommand{\sstemplates}{\input{apsstempl}}
```

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

```
⟨/ap⟩
⟨*apsstempl⟩
\ProvidesFile{apsstempl.tex}[2001/07/16 AP SUP/SUB templates boilerplate]
\typeout{apsstempl.tex [2001/07/16 AP SUP/SUB templates boilerplate]}

\sssclause{SUBTYPE}

    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

/SUBTYPE(application\_object)/

\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.


\sssclause{SUPERTYPE}

    The SUPERTYPE mapping template specifies a reference to the mapping of
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}

\signature

/SUPERTYPE(application\_object)/

```

```
\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.

⟨/apsstempl⟩
⟨*ap⟩
```

`\apshortexpress` Print boilerplate for AP AIM EXPRESS short listing.

```
\newcommand{\apshortexpress}{\input{bpfap6}}
```

Here is the text of file `bpfap6.tex`

```
⟨/ap⟩
⟨*bpfap6⟩
\ProvidesFile{bpfap6.tex}[2002/01/22 AP AIM EXPRESS short listing boilerplate]
\typeout{bpfap6.tex [2002/01/22 AP AIM EXPRESS short listing boilerplate]}

  This clause specifies the \Express{} schema that uses
elements from the integrated resources
\ifaicinap and the AICs \fi
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
\ifaicinap resources and the AICs. \else resources. \fi
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.

⟨/bpfap6⟩
⟨*ap⟩
```

`\apconformance` Print boilerplate for AP conformance.
`\apconformance{`⟨*implentation methods*⟩`}`

```
\newcommand{\apconformance}[1]{%

\input{bpfap7}

  An implementation shall support at least one of the following
implementation methods: #1.

\input{bpfap8}

```

```
}
```

Here is the text of file bpfap7.tex

```
⟨/ap⟩
⟨*bpfap7⟩
\ProvidesFile{bpfap7.tex}[1997/09/30 AP conformance boilerplate (1)]
\typeout{bpfap7.tex [1997/09/30 AP conformance boilerplate (1)]}

  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.

⟨/bpfap7⟩
⟨*ap⟩
```

Here is the text of file bpfap8.tex

```
⟨/ap⟩
⟨*bpfap8⟩
\ProvidesFile{bpfap8.tex}[1997/09/30 AP conformance boilerplate (2)]
\typeout{bpfap8.tex [1997/09/30 AP conformance boilerplate (2)]}

Requirements with respect to implementation methods-specific
requirements are specified in \aref{;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 \aref{;spics}.

⟨/bpfap8⟩
⟨*ap⟩
```

apconformclasses Print boilerplate for AP conformance classes. `\begin{apconformclasses}`⟨*class list*⟩`\end{apconformclasses}` where ⟨*class list*⟩ is a list of conformance classes in `\item` format.

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

 }
```

\apshortnames Print boilerplate for AP short names.

```
\newcommand{\apshortnames}{\input{bpfap9}}
```

Here is the text of file bpfap9.tex

```
⟨/ap⟩
⟨*bpfap9⟩
\ProvidesFile{bpfap9.tex}[1997/09/30 AP short names boilerplate]
\typeout{bpfap9.tex [1997/09/30 AP short boilerplate]}

  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.

⟨/bpfap9⟩
⟨*ap⟩
```

`\picsannex` Print boilerplate for PICS annex.

```
\newcommand{\picsannex}{\input{bpfap10}}
```

Here is the text of file bpfap10.tex

```
⟨/ap⟩
⟨*bpfap10⟩
\ProvidesFile{bpfap10.tex}[1997/09/30 AP PICS annex boilerplate]
\typeout{bpfap10.tex [1997/09/30 AP PICS annex boilerplate]}

  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.

⟨/bpfap10⟩
⟨*ap⟩
```

`\aamfigrs` `\aamfigrange` 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 `\aamfigrs` which is given an initial value just in case the user forgets to call `\aamfigrange`. The value of `\aamfigrs` is used in later boilerplate.

```
\gdef\aamfigrs{figure F.1}
\newcommand{\aamfigrange}[1]{\gdef\aamfigrs{#1}}
```

`\apaamintro` Print boilerplate for AAM annnex intro.

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

Here is the contents of `bpfap11.tex`. Note the use of the `\aamfigrs` command.

```
⟨/ap⟩
⟨*bpfap11⟩
\ProvidesFile{bpfap11.tex}[2001/07/16 AP AAM annex intro boilerplate]
\typeout{bpfap11.tex [2001/07/16 AP AAM annex intro boilerplate}

  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 \aamfigrs.
Activities and data flows that are out of scope are marked with
an asterisk.

⟨/bpfap11⟩
⟨*ap⟩
```

`\apaamdefs` Print boilerplate for AAM definitions.

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

Here is the text of file `bpfap12.tex`

```
⟨/ap⟩
⟨*bpfap12⟩
\ProvidesFile{bpfap12.tex}[1997/09/30 AP AAM definitions boilerplate]
\typeout{bpfap12.tex [1997/09/30 AP AAM definitions boilerplate]}

    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.

⟨/bpfap12⟩
⟨*ap⟩
```

`\aamfigures` Print boilerplate for AAM figures. `\aamfigures`

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

Here is the contents of `bpfap15.tex`.

```
⟨/ap⟩
```

```
⟨*bpfap15⟩
\ProvidesFile{bpfap15.tex}[2001/07/16 AP AAM annex figures subclause boilerplate]
\typeout{bpfap15.tex [2001/07/16 AP AAM annex figures subclause boilerplate]}

  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.

⟨/bpfap15⟩
⟨*ap⟩
```

`\armintro` Print boilerplate for ARM introduction.

```
\newcommand{\armintro}{%
  This annex provides the application reference model for this part of ISO
  10303. The application reference model is a graphical
  representation of the structure and constraints of the application objects
  specified in \cref{;sireq}. The graphical form of the application reference
  model is presented in \ifidefix IDEF1X. \else \ExpressG. \fi
  The application reference model is
  independent from any implementation method.
  \ifidefix The diagrams use the IDEF1X graphical notation~\brefidefix.
  \else \expressgdef. \fi

}
```

`\aimexpressg` Print boilerplate for AIM EXPRESS-G.

```
\newcommand{\aimexpressg}{%
  The diagrams in this annex correspond to the AIM \Express{} expanded
listing given in \aref{;saeel}.
The diagrams use the \ExpressG{} graphical notation for the
\Express{} language. \expressgdef.

}
```

`\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.

```
\newcommand{\apexpurls}[2]{\input{bpfap13}

  \begin{itemize}
  \item Short names: \isourl{#1}
  \item \Express: \isourl{#2}
  \end{itemize}

  \input{bpfap16}}
```

Here is the text of file bpfap13.tex

```
⟨/ap⟩
⟨*bpfap13⟩
\ProvidesFile{bpfap13.tex}[2001/07/16 AP short names/EXPRESS listing boilerplate (1)]
\typeout{bpfap13.tex [2001/07/16 AP ahort 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:

⟨/bpfap13⟩
%
% Here is the text of \file{bpfap16.tex}.
% \changes{v1.3}{1999/02/15}{Added file bpfap16.tex}
⟨*bpfap16⟩
\ProvidesFile{bpfap16.tex}[1999/02/15 AP short names and EXPRESS annex ending boilerplate]
\typeout{bpfap16.tex [1997/09/30 AP short names and EXPRESS annex ending boilerplate]}

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

\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}

⟨/bpfap16⟩
⟨*ap⟩
```

`\aimlongexp` Print boilerplate for AIM EXPRESS expanded listing.

```
\newcommand{\aimlongexp}{\input{bpfap14}}
```

Here is the text of file bpfap14.tex

```
⟨/ap⟩
⟨*bpfap14⟩
\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~\ref{;saesl}. In the event of any discrepancy
between the short form and this expanded listing, the expanded
listing shall be used.

⟨/bpfap14⟩
⟨*ap⟩
```

`\apimpreq` Print boilerplate for AP requirements on exchange structure.
`\apimpreq{`⟨*schema name*⟩`}`.

```
\newcommand{\apimpreq}[1]{%
  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 \aref{;saeel} 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 ‘#1’.

}
```

The end of this package.

⟨/ap⟩

8 The Application Interpreted Construct package

This section defines the contents of the package designed for use in documenting STEP AICs.

⟨∗aic⟩

If we are in an AIC we are not in an IR.

```

\anirfalse

```

8.1 Heading commands

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

`\aicshortexphead` Starts an ‘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~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.

```
\newcommand{\aicdef}{%
\definition{application interpreted construct (AIC)}%
           {a logical grouping of interpreted constructs
            that supports a specific function for
            the usage of product data across multiple
            application contexts.}
}
```

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

```
\newcommand{\aicshortexpintro}{%
    This clause specifies the \Express{} schema that uses
elements from the integrated resources and contains the
types, entity data types specializations, and functions that are
specific to this part of ISO~10303.
\begin{anote}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.
\end{anote} % end note
}
```

`\aicexpressg` Print boilerplate for AIC EXPRESS-G.

```
\newcommand{\aicexpressg}{%
  The diagrams in this annex are generated from the short
listing given in \cref{;sesl} and correspond to the \Express{} schemas
specified in this part of ISO 10303.
The diagrams use the \ExpressG{} graphical notation for the
\Express{} language. \expressgdef. \par
}
```

The end of this package.

```
⟨/aic⟩
```

9 The Abstract Test Suite package

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

1644 ⟨∗ats⟩

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

1645
1646 `\anirfalse`
1647

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
`\APnumber` ATS. Internally it is referred to as `\theAPpartno`.
`\theAPtitle`
`\APtitle`

```
1648 \gdef\theAPpartno{}
1649 \newcommand{\APnumber}[1]{\gdef\theAPpartno{#1}}
1650 \gdef\theAPtitle{}
1651 \newcommand{\APtitle}[1]{\gdef\theAPtitle{#1}}
1652
```

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`

```
1653 \newcommand{\atssummary}{\underline{\texttt{Test case summary:}}}
1654 \newcommand{\atscovered}{\underline{\texttt{Test purposes covered:}}}
1655 \newcommand{\atsinput}{\underline{\texttt{Input specification:}}}
1656 \newcommand{\atsconstraints}{\underline{\texttt{Constraints on values:}}}
1657 \newcommand{\atsverdict}{\underline{\texttt{Verdict criteria:}}}
1658 \newcommand{\atsexecution}{\underline{\texttt{Execution sequence:}}}
1659 \newcommand{\atsextra}{\underline{\texttt{Extra details:}}}
1660
```

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.

```
1661 \newcommand{\purposeshead}{\clause{Test purposes}}
```

`\domainpurposehead` Command to start a 'Domain test purposes' clause.

```
1662 \newcommand{\domainpurposehead}{\sclause{Domain test purposes}}
```

\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}{\sclause{Other test purposes}}
```

\gtpvchead Command to start a 'General test purposes and verdict criteria' clause.

```
\newcommand{\gtpvchead}{\clause{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]{\sclause{#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. Us as \confclasshead{⟨*number*⟩}.

```
\newcommand{\confclasshead}[1]{\sclause{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.

```
⟨/ats⟩
⟨*bpfats1⟩
\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~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 implentations 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.

⟨/bpfats1⟩
⟨*ats⟩
```

`\atsscopebp` The boilerplate for the ATS scope clause.

```
\newcommand{\atsscopebp}{%
  \input{bpfats2}
}
```

Here is the text of `bpfats2.tex`.

```
⟨/ats⟩
⟨*bpfats2⟩
\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}

⟨/bpfats2⟩
⟨*ats⟩
```

`\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. \par
}

```

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

```
\newcommand{\aetpbp}{%
  \input{bpfats3}
}
```

And here is the text of file bpfats3.tex.

```
⟨/ats⟩
⟨*bpfats3⟩
\ProvidesFile{bpfats3.tex}[2002/01/23 ATS AE test purpose intro boilerplate]
\typeout{bpfats3.tex [2002/01/23 ATS AE test purpose intro boilerplate]}

    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
preprocessr and postprocessor test cases. AE test purposes are implicitly
derived from the AP information requirements as follows:
\begin{itemize}
\item 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;

\item 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;

\item 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.

\end{itemize}


They shall be interpreted as given in the
following statement:
%\begin{quotation}
    the IUT shall preserve the semantic associated with the unique
application element from which the test purpose was implicitly derived.
%\end{quotation}
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


⟨/bpfats3⟩
⟨*ats⟩
```

`\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 `bpfats4.tex`.

```
⟨/ats⟩
⟨*bpfats4⟩
\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 \Express,
and are not explicitly documented here. AIM test purposes are implicitly
derived from the expanded \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}

    Aim test purposes shall be interpreted as given in the
following statement:
%\begin{quotation}
the postprocessor shall accept the input in accordance with the
AIM \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
```

```
1891
1892 ⟨/bpfats4⟩
1893 ⟨*ats⟩
```

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

```
1894 \newcommand{\atsimtpbp}{\input{bpfats14}}
1895
```

And here is the text of file bpfats14.tex.

```
1896 ⟨/ats⟩
1897 ⟨*bpfats14⟩
1898 \ProvidesFile{bpfats14.tex}[2001/07/16 ATS implementation method test purpose intro boilerplate
1899 \typeout{bpfats14.tex [2001/07/16 ATS implementation method test purpose intro boilerplate]}
1900
1901     The following test purpose is derived from requirements in
1902 ISO 10303-21 and applies to preprocessors only.
1903
1904 other1 The IUT correctly encodes the AIM schema name in the exchange
1905        structure.
1906
1907     The following test purposes are derived from requirements in
1908 ISO 10303-21 and apply to postprocessors only.
1909
1910 other2 The IUT interprets the ISO 10303-21 header section
1911        present in the exchange structure.
1912
1913 other3 The IUT interprets the ISO 10303-21 SCOPE and EXPORT constructs
1914        present in the exchange structure.
1915
1916 other4 The IUT interprets the ISO 10303-21 user-defined entity constructs
1917        present in the exchange structure.
1918
1919 other5 The IUT interprets various representations of numbers
1920        present in the exchange structure
1921        in accordance with ISO 10303-21.
1922
1923 other6 The IUT interprets various sequences of symbols
1924        present in the exchange structure
1925        in accordance with ISO 10303-21.
1926
1927 \par
1928
1929 ⟨/bpfats14⟩
1930 ⟨*ats⟩
```

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

```
1931 \newcommand{\atsgtpvcbp}{%
```

```
  \input{bpfats5}
}
```

Here is the text of file `bpfats5.tex`.

```
⟨/ats⟩
⟨*bpfats5⟩
\ProvidesFile{bpfats5.tex}[1997/09/30 ATS general verdict boilerplate]
\typeout{bpfats5.tex [1997/09/30 ATS general verdict boilerplate]}

    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.

⟨/bpfats5⟩
⟨*ats⟩
```

`\gtpbp` Command to print the boilerplate introduction to General test purposes clause.

```
\newcommand{\gtpbp}{%
  \input{bpfats6} }
```

And here is the text of file `bpfats6.tex`

```
⟨/ats⟩
⟨*bpfats6⟩
\ProvidesFile{bpfats6.tex}[2001/07/16 ATS general test purpose boilerplate]
\typeout{bpfats6.tex [2001/07/16 ATS general test purpose boilerplate]}

    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

⟨/bpfats6⟩
⟨*ats⟩
```

\gvatcbp Command to print the boilerplate introduction to *General verdict criteria* clause.

```
\newcommand{\gvatcbp}{%
  \input{bpfats7} }
```

And here is the text of file `bpfats7.tex`

```
⟨/ats⟩
⟨*bpfats7⟩
\ProvidesFile{bpfats7.tex}[2001/07/16 ATS general verdict criteria boilerplate]
\typeout{bpfats7.tex [2001/07/16 ATS general verdict criteria boilerplate]}

    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
     \maptableorspec{} defined in ISO 10303-\theAPpartno, clause 5 (g1).

\par

⟨/bpfats7⟩
⟨*ats⟩
```

\gvcprebp Command to print the boilerplate introduction to *General verdict criteria for preprocessor* clause.

```
\newcommand{\gvcprebp}{%
  \input{bpfats8} }
```

And here is the text of file `bpfats8.tex`

```
⟨/ats⟩
⟨*bpfats8⟩
\ProvidesFile{bpfats8.tex}[2001/07/16 ATS general verdict pre boilerplate]
\typeout{bpfats8.tex [2001/07/16 ATS general verdict pre boilerplate]}

    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 \maptableorspec{}
     defined in ISO 10303-\theAPpartno, annex A and 5.1 (g3).
\par

⟨/bpfats8⟩
⟨*ats⟩
```

\gvcpostbp Command to print the boilerplate introduction to *General verdict criteria for postprocessor* clause.

```
\newcommand{\gvcpostbp}{%
  \input{bpfats9} }
```

And here is the text of file `bpfats9.tex`

```
⟨/ats⟩
⟨*bpfats9⟩
\ProvidesFile{bpfats9.tex}[2001/07/16 ATS general verdict post boilerplate]
\typeout{bpfats9.tex [2001/07/16 ATS general verdict post boilerplate]}

    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).
\par


⟨/bpfats9⟩
⟨*ats⟩
```

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

```
\newcommand{\atcbp}{%
    This clause specifies the abstract test cases for this part of
ISO 10303. Each abstract test case addresses one or more test purposes
explicitly or implicitly specified in clause~4.
\par
}

```

`\atcbpii` is for printing the major portion of the boilerplate (paragraphs 3 onwards).

```
\newcommand{\atcbpii}{%
  \input{bpfats10}
}
```

And here is the text of files `bpfats10.tex` and `bpfats11.tex`.

```
⟨/ats⟩
⟨*bpfats10⟩
\ProvidesFile{bpfats10.tex}[2001/07/16 ATS ats clause boilerplate]
\typeout{bpfats10.tex [2001/07/16 ATS ats clause boilerplate]}

    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:
\begin{itemize}
\item The 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.

\item The 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.

\item The Application Elements column identifies the particular
      application element instance that is being
      defined by the table. For assertions the role is specified
      in parenthesis.

\item 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.
      A '$<$not\_present$>$' indicates that the
      application element is not present in the
      input model.

```

```
\item The Req column specifies whether the value in the Value column
      is mandatory (M), suggested (S) or constrained (C$<$n$>$), 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$<$number$>$ at the end of the preprocessor
      input model table and may be modified according to specific
      constraints specified in it.
\end{itemize}

    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.

⟨/bpfats10⟩
⟨*ats⟩
```

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

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

```

Here is the text of `bpfats11.tex`.

```
⟨/ats⟩
⟨*bpfats11⟩
\ProvidesFile{bpfats11.tex}[2001/07/16 ATS preprocessor purposes covered boilerplate]
\typeout{bpfats11.tex [2001/07/16 ATS preprocessor purposes covered boilerplate]}

    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.
\par

⟨/bpfats11⟩
⟨*ats⟩
```

`\atcposttpc` `\atcposttpc` prints boilerplate for the Postprocessor Test Purposes Covered subclause.

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

Here is the text of `bpfats12.tex`.

```
⟨/ats⟩
⟨*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-\theAPpartno, 4.2, where the
application object is defined.
\par

⟨/bpfats12⟩
⟨*ats⟩
```

`\confclassbp` `\atsnoclassesbp` `\confclassbp{`⟨*number*⟩`}` prints the boilerplate for the start of a *Conformance class* clause.

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

```
\newcommand{\confclassbp}[1]{%

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

`\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:
\begin{center}
\isourl{#2}
```

```
\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~#3 lists
the file name of the postprocessor input specification that is
associated with the postprocessor subclause(s) of a test case.
\par
%%%  \input{bpfats13}
}
```

Here is the text of `bpfats13.tex`.

```
⟨/ats⟩
⟨*bpfats13⟩
\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~10303 without comments or other
explanatory text. These specifications are documented using
ISO 10303-\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~\atstempc{} lists
the file name of the postprocessor input specification that is
associated with the postprocessor subclause(s) of a test case.


⟨/bpfats13⟩
⟨*ats⟩
```

The end of this package.

```
⟨/ats⟩
```

References

[GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.

[ISO97] *ISO/IEC Directives Part 3 — Drafting and presentation of International Standards*, third edition, 1997.

[Sec97a] SC4 Secretariat. *Guidelines for the development of abstract test suites*. ISO TC184/SC4/WG6 Document N536, March 1997. (Available from NIST, Gaithersburg, MD 20899.).

[Sec97b] SC4 Secretariat. *Supplementary directives for the drafting and presentation of ISO 10303*. ISO TC184/SC4 Document N537, March 1997. (Available from NIST, Gaithersburg, MD 20899.).

[Wil96a] Peter R. Wilson. *LaTeX files for Typesetting ISO Standards: Source code*. NIST Report NISTIR, June 1996.

[Wil96b] Peter R. Wilson. *LaTeX for standards: The LaTeX package files user manual*. NIST Report NISTIR, June 1996.

[Wil96c] Peter R. Wilson. *LaTeX package files for ISO 10303: User manual*. NIST Report NISTIR, June 1996.

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.

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z