

L^AT_EX Package Files for ISO 10303: Source code*

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

2002/01/10

Contents

1	Introduction	1
2	A driver for this document	2
3	Identification	2
4	Initial Code	3
5	The STEP package	3
5.1	Preamble commands	3
5.2	Indexing style commands	4
5.3	Miscellaneous commands	5
5.3.1	Font changes	5
5.3.2	Logos	5
5.3.3	EXPRESS code symbols	5
5.3.4	marginal notes	6
5.4	EXPRESS code documentation	6
5.4.1	environments	6
5.4.2	Indexing	8
5.5	STEP part title	9
5.6	Headings and boilerplate	9
5.6.1	Foreword elements	9
5.6.2	The introduction	11
5.6.3	Miscellaneous headings	12
5.6.4	Miscellaneous boilerplate	14
5.7	Common references	16
5.8	Cover sheet	18

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

6	The Integrated Resources package	25
6.1	Boilerplate	26
7	The Application Protocol package	26
7.1	Preamble commands	26
7.2	Heading commands	27
7.2.1	Template headings	29
7.3	Boilerplate printing	29
8	The Application Interpreted Construct package	46
8.1	Heading commands	46
8.2	Boilerplate commands	46
9	The Abstract Test Suite package	48
9.1	Preamble commands	48
9.2	Keyword commands	48
9.3	Heading commands	48
9.4	Boilerplate printing	50

1 Introduction

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

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

Some of the STEP standard documents have been published by ISO from camera ready copy derived from electronic sources (this also means that ISO has not objected to the typographical conventions supported by these packages). Within ISO there are proposals to maintain and publish directly from SGML tagged electronic sources. The packages have been designed to simplify the conversion from \LaTeX to SGML tagging. Thus, there are more document structural elements defined than is usual with \LaTeX .

As already noted, the macros described later are based on the STEP Supplementary Directives. If in the future the Directives are modified or extended, then it may be necessary to modify or extend the macros. Essentially, this manual is provided as a service for maintainers of the L^AT_EX packages. It is assumed that any package maintainer is L^AT_EX literate and accustomed to supporting a L^AT_EX system [GMS94].

2 A driver for this document

The next bit of code contains the documentation driver file for L^AT_EX, i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

```
1 <*driver>
2 \documentclass{ltxdoc}
```

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

```
3 \EnableCrossrefs
4 \CodelineIndex
5 %% \RecordChanges
```

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

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

Define some commonly used abbreviations

```
7 \newcommand*\lopt[1]{\textsf {#1}}
8 \newcommand*\file[1]{\texttt {#1}}
9 \newcommand*\lcount[1]{\textsl {\small#1}}
10 \newcommand*\pstyle[1]{\textsl {#1}}
```

We also want the full details printed.

```
11 \begin{document}
12 \DocInput{stepe.dtx}
13 \PrintIndex
14 %% \PrintChanges
15 \end{document}
16 </driver>
```

3 Identification

These packages can only be used with L^AT_EX2e.

```
17 <*step | ir | ap | ats | aic | am>
```

Announce the Package name and its version:

```
18 <*step>
19 \ProvidesPackage{stepv13}[2002/01/10 v1.3.2 STEP general package]
20 </step>
21 <*ir>
```

```

22 \ProvidesPackage{irv12}[2002/01/10 v1.2.2 STEP IR package]
23 </ir>
24 <*ap>
25 \ProvidesPackage{apv12}[2002/01/10 v1.2.2 STEP AP package]
26 </ap>
27 <*ats>
28 \ProvidesPackage{atsv11}[2002/01/10 v1.1.2 STEP ATS package]
29 </ats>
30 <*aic>
31 \ProvidesPackage{aicv1}[2002/01/10 v1.0.2 STEP AIC package]
32 </aic>
33 <*am>
34 \ProvidesPackage{amv1}[2002/01/10 v1.0 STEP AM package]
35 </am>
36 </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.

```

37 <*ir | ap | ats | aic | am>
38   \RequirePackage{stepv13}[2002/01/10]
39
40 </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.
41 <step> \let\stepemptystring\isoemptystring

```

5 The STEP package

This section defines the facilities available in the STEP package.

```

42 <*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.
43 \gdef\thespartno{}
44 \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` 45 `\gdef\theseries{}`
46 `\gdef\Theseries{}`
47 `\newcommand{\series}[1]{\gdef\Theseries{#1}`
48 `\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` 49 `\gdef\thed@ctitle{}`
50 `\newcommand{\doctitle}[1]{\gdef\thed@ctitle{#1}}`
51 `\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.
52 `\newcounter{b@cyc}`
53 `\newcommand{\ballotcycle}[1]{\setcounter{b@cyc}{#1}}`

`\ifanir` TRUE if the document is an IR (generic or application).
54 `\newif\ifanir`
55 `\anirfalse`
56

`\ifhaspatents` TRUE if the document has identified patents.
57 `\newif\ifhaspatents`
58 `\haspatentsfalse`
59

`\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).
60 `\newif\ifmapspec`
61 `\mapspectfalse`

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` 62 `\renewcommand{\indexfill}{\dotfill}`
`\ssindexfill` 63 `\renewcommand{\sindexfill}{\dotfill}`
64 `\renewcommand{\ssindexfill}{\dotfill}`

`\alphaindexspace` No extra vertical spacing between blocks of index entries,
`\otherindexspace` 65 `\renewcommand{\alphaindexspace}[1]{}`
66 `\renewcommand{\otherindexspace}[1]{}`

`\indexsee` Formatting of *see* and *see also*.
`\indexseealso` 67 `\renewcommand{\indexsee}[1]{\par \hspace*{2em} {\em see} #1}`
68 `\renewcommand{\indexseealso}[1]{\par \hspace*{2em} {\em see also} #1}`
69
`\ix` Both print and index a word or phrase.
70 `\newcommand{\ix}[1]{#1\index{#1}}`
71

5.3 Miscellaneous commands

5.3.1 Font changes

`\B` `\B{<text>}` prints *<text>* in bold while `\E{<text>}` prints it *emphasized*. `\BG{<mathsymbol>}`
`\E` prints *<mathsymbol>* in bold.
`\BG` 72 `\newcommand{\B}[1]{\bf #1}`
73 `\newcommand{\E}[1]{\em #1}`
74 `\newcommand{\BG}[1]{\mbox{\boldmath $#1$}}`
75

5.3.2 Logos

`\Express` The commands print the logos for the EXPRESS family of information modeling
`\ExpressG` languages. (Note: In Part 11 the macros were specified as `{\small\s1 EX\~PRESS}`,
`\ExpressI` etc. but the STEP Editing Committee ignored the wishes of the authors of EX-
`\ExpressX` PRESS leading to the definitions below.)
76 `\newcommand{\Express}{\sc EX\~PRESS}`
77 `\newcommand{\ExpressG}{\sc EX\~PRESS-G}`
78 `\newcommand{\ExpressI}{\sc EX\~PRESS-I}`
79 `\newcommand{\ExpressX}{\sc EX\~PRESS-X}`
80

5.3.3 EXPRESS code symbols

`\nexp` Highlight an EXPRESS-defined name.
81 `\newcommand{\nexp}[1]{\textbf{#1}}`

`\HASH` Various symbols used within EXPRESS.
`\LT` 82 `\newcommand{\HASH}{\texttt{\small \#}}`
`\LE` 83 `\newcommand{\LT}{\texttt{\small <}}`
`\NE` 84 `\newcommand{\LE}{\texttt{\small <=}}`
`\INE` 85 `\newcommand{\NE}{\texttt{\small <>}}`
`\GE` 86 `\newcommand{\INE}{\texttt{\small <:>}}`
`\GT` 87 `\newcommand{\GE}{\texttt{\small >=}}`
88 `\newcommand{\GT}{\texttt{\small >}}`

`\CAT` More EXPRESS symbols.

`\HAT` 89 `\newcommand{\CAT}{\texttt{\small ||}}`

`\QUES` 90 `\newcommand{\HAT}{\texttt{\small ^}}`

`\BS` 91 `\newcommand{\QUES}{\texttt{\small ?}}`

`\IEQ` 92 `\newcommand{\BS}{\texttt{\small \}}`

`\INEQ` 93 `\newcommand{\IEQ}{\texttt{\small :=}}`

94 `\newcommand{\INEQ}{\texttt{\small :<}}`

95

`\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.

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

97

5.3.4 marginal notes

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

98 `\newcommand{\mnote}[1]{\ifdr@ftd@`

99 `\marginpar{\raggedright\tiny #1}`

100 `\fi}`

101

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.

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

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

`fspec`

`rspec` 103 `\newenvironment{espec}[1]{}{}`

`sspec` 104 `\newenvironment{fspec}[1]{}{}`

`tspec` 105 `\newenvironment{rspec}[1]{}{}`

106 `\newenvironment{sspec}[1]{}{}`

107 `\newenvironment{tspec}[1]{}{}`

`dttext` An environment to tag descriptive text.

108 `\newenvironment{dttext}{}{}`

109

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

```

110 \newcommand{\pbre@k}{\pagebreak[2]}
111 \newcommand{\nopbre@k}{\nopagebreak}
112

```

`\ehe@d` An internal command for (underlined) headings. `\ehe@dmarmk` is required otherwise

`\ehe@dmarmk` the title is printed twice!

```

113 \newcommand{\ehe@d}{\@startsection{ehe@d}{20}
114   {\z@}%           % indent
115   {-\baselineskip}% % beforekip
116   {0.5\baselineskip}% % afterskip
117   {}%             % normal body text style for heading
118 \newcounter{ehe@d}
119 \newcommand{\ehe@dmarmk}[1]{%
120

```

`ecode` Environment for writing EXPRESS code.

```

121 \newenvironment{ecode}{%
122   \ehe@d*{\underline{\protect\Express{}} specification}}{:}
123   \begin{Efont}}%
124   {\end{Efont}}
125

```

`eicode` Environment for writing EXPRESS-I code.

```

126 \newenvironment{eicode}{%
127   \ehe@d*{\underline{\protect\ExpressI{}} specification}}{:}
128   \begin{Efont}}%
129   {\end{Efont}}
130

```

`excode` Environment for writing EXPRESS-X code.

```

131 \newenvironment{excode}{%
132   \ehe@d*{\underline{\protect\ExpressX{}} specification}}{:}
133   \begin{Efont}}%
134   {\end{Efont}}
135

```

`expdesc` A non-indented description environment.

`\expdesclabel` The label for the description list. Note that it includes a colon.

```

136 \newcommand{\expdesclabel}[1]{\bf #1:}
137 \newenvironment{expdesc}{\list{}}%
138   {\setlength{\leftmargin}{\z@} \setlength{\labelsep}{0.5em}
139   \setlength{\itemindent}{\labelsep} \setlength{\labelwidth}{\z@}
140   \setlength{\itemsep}{\z@ \@plus 0.2ex \@minus 0.1ex}
141   \setlength{\parsep}{0.5\baselineskip}
142   \let\makelabel\expdesclabel}}%
143   {\endlist}
144

```


attrlist Listing of attribute descriptions.

```
145 \newenvironment{attrlist}{%
146     \ehead*{\underline{Attribute definitions}}:}
147     \begin{expdesc}}%
148     {\end{expdesc}}
149
```

fproplist Listing of formal propositions.

```
150 \newenvironment{fproplist}{%
151     \ehead*{\underline{Formal propositions}}:}
152     \begin{expdesc}}%
153     {\end{expdesc}}
154
```

iproplist Listing of informal propositions.

```
155 \newenvironment{iproplist}{%
156     \ehead*{\underline{Informal propositions}}:}
157     \begin{expdesc}}%
158     {\end{expdesc}}
159
```

enumlist Listing of enumerated items.

```
160 \newenvironment{enumlist}{%
161     \ehead*{\underline{Enumerated item definitions}}:}
162     \begin{expdesc}}%
163     {\end{expdesc}}
164
```

arglist Listing of argument definitions.

```
165 \newenvironment{arglist}{%
166     \ehead*{\underline{Argument definitions}}:}
167     \begin{expdesc}}%
168     {\end{expdesc}}
169
```

5.4.2 Indexing

\ixent Macros for indexing EXPRESS definitions.

```
\ixenum 170 \newcommand{\ixent}[1]{\index{#1 (entity)}}
\ixfun 171 \newcommand{\ixenum}[1]{\index{#1 (enumeration)}}
\ixproc 172 \newcommand{\ixfun}[1]{\index{#1 (function)}}
\ixrule 173 \newcommand{\ixproc}[1]{\index{#1 (procedure)}}
\ixsc 174 \newcommand{\ixrule}[1]{\index{#1 (rule)}}
\ixschema 175 \newcommand{\ixsc}[1]{\index{#1 (subtype\_constraint)}}
\ixselect 176 \newcommand{\ixschema}[1]{\index{#1 (schema)}}
\ixtype 177 \newcommand{\ixselect}[1]{\index{#1 (select)}}
178 \newcommand{\ixtype}[1]{\index{#1 (type)}}
179
```

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.

```
180 \gdef\thestepparttitle{}
181 \newcommand{\scivm@in}{Industrial automation systems and integration ---\newline}
182 \newcommand{\stepc@mp}{Product data representation and exchange ---\newline}
183 \newcommand{\this@rtno}[1]{Part #1 :\newline}
184 \newcommand{\sptitle}[1]{#1\par}
185 \newcommand{\stepparttitle}[1]{%
186     \cleardoublepage\pagenumbering{arabic}
187 %%%     \setcounter{section}{0}
188     \setcounter{clause}{0}
189     \ifotherdoc \else
190         \protect\thispagestyle{isotitlehead}
191     \fi
192     \gdef\thestepparttitle{{\Tfont\bf \scivm@in \stepc@mp
193                                     \this@rtno{\thespartno} \sptitle{#1}}}
194     \if@twocolumn
195         \twocolumn[\vspace*{2\baselineskip}\vbox to 35mm{\thestepparttitle}]
196     \else
197         \vspace*{2\baselineskip}\vbox to 35mm{\thestepparttitle}
198     \fi}
199
```

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.

```
200 \newcommand{\Foreword}{%
201     \begin{foreword}
202 %%%     \input{isofwdbp}
203     \fwdbp
204
205     \ifhaspatents\else\fdnopatents\fi
206
207     \iftechspec
208         ISO/TS~10303--\thespartno\
209     \else
210         \ifaspec
211             ISO/PAS~10303--\thespartno\
212         \else
213             ISO~10303--\thespartno\
214         \fi

```

```

215     \fi
216     was prepared by Technical Committee
217     ISO/TC~184, \textit{Industrial automation systems and integration},
218     Subcommittee SC4, \textit{Industrial data}.
219 }

```

`\endForeword` The command for ending the STEP Foreword. Use as:

```

\endForeword{\(normannexes)}{\(infannexes)}
220 \gdef\endForeword#1#2{%
221 \par
222     A complete list of parts of ISO~10303 is available from the Internet:\\
223 \centerline{\isourl{http://www.nist.gov/sc4/editing/step/titles/}}
224 \par
225
226 % Don't talk about annexes if relevent argument is empty.
227 \if\stepemptystring{#1} \else%
228 #1 a normative part of this part of ISO~10303. \fi%
229     % an integral part of this part of ISO~10303. \fi%
230 \if\stepemptystring{#2} \else%
231 #2 for information only. \fi
232 \end{foreword}
233 }
234

```

`\steprid` Boilerplate for the foreword describing the creators of a TR.

```

235 \newcommand{\steprid}{%
236
237     ISO/TR~10303--\thespartno, which is a Technical Report of type 2,
238     was prepared by Technical Committee
239     ISO/TC~184, \textit{Industrial automation systems and integration},
240     Subcommittee SC4, \textit{Industrial data.}
241
242 }
243

```

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

```

244 \newcommand{\fwdshortlist}{\input{stppdlst}}
245

```

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

```

246 </step>
247 < *fwd4 >
248 \ProvidesFile{stppdlst.tex}[2001/07/16 STEP parts and divisions URL]
249 \typeout{stppdlst.tex [2001/07/16 STEP parts and divisions URL]}
250
251     This International Standard is organized as a series of parts,
252     each published separately. The structure of this International

```

253 Standard is described in ISO~10303--1.
254
255 Each part of this International Standard is a member of one
256 of the following series:
257 description methods,
258 implementation methods,
259 conformance testing methodology and framework,
260 integrated generic resources,
261 integrated application resources,
262 application protocols,
263 abstract test suites,
264 application interpreted constructs,
265 and
266 application modules.
267 This part is a member of the \theseries{} series.
268 \ifanir The integrated generic resources and the integrated application
269 resources specify a single conceptual product data model.
270 \fi
271
272
273 </fwd4>
274 <*step>

5.6.2 The introduction

Introduction Starts a new ‘introduction’ clause, together with initial STEP boilerplate.

```
275 \newenvironment{Introduction}{%
276 \clearpage
277 \begin{introduction}
278 \input{bpfs1}
279
280 }%
281 {\end{introduction}}
```

Here is the text maintained in file bpfs1.tex.

```
283 </step>
284 <*bpfs1>
285 \ProvidesFile{bpfs1.tex}[2001/07/16 STEP Intro boilerplate]
286 \typeout{bpfs1.tex [2001/07/16 STEP Intro boilerplate]}
287
288 ISO 10303 is an International Standard for the computer-interpretable
289 representation of product information and for the exchange of product data.
290 The objective is to
291 provide a neutral mechanism capable of describing products
292 throughout their life cycle.
293 This mechanism is suitable
294 not only for neutral file exchange, but also as a basis for
295 implementing and sharing product databases, and as a basis for archiving.
```

```

296
297 </bpfsl>
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 defined 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 Identification of a clause describing an EXPRESS schema, and the introductory boilerplate.

\schemaintro

```

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 Starts a ‘type definition’ or ‘type definitions’ subclause.

```

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

```

`\atypehead` Starts a ‘type definition’ subsubclause.
318 `\newcommand{\atypehead}[1]{\ssclause{#1}}`

`\singleentityhead` Starts an ‘entity definition’ subclause or an ‘entity definitions’ subclause. Use the latter as:
`\entityhead` `\entityhead{<schema>}{<group>}` where `<schema>` is the name of the schema and `<group>` is a possibly blank grouping identifier.
319 `\newcommand{\singleentityhead}[2]{\sclause{#1 entity definition: #2}}`
320 `\newcommand{\entityhead}[2]{%`
321 `\ifstepemptystring{#2}`
322 `\sclause{#1 entity definitions}`
323 `\else`
324 `\sclause{#1 entity definitions: #2}`
325 `\fi`
326 `}`

`\anentityhead` Starts an ‘entity definition’ subsubclause.
327 `\newcommand{\anentityhead}[1]{\ssclause{#1}}`

`\singerulehead` Starts a ‘rule definition’ or ‘rule definitions’ subclause.
`\rulehead` 328 `\newcommand{\singerulehead}[2]{\sclause{#1 rule definition: #2}}`
329 `\newcommand{\rulehead}[1]{\sclause{#1 rule definitions}}`

`\arulehead` Starts a ‘rule definition’ subsubclause.
330 `\newcommand{\arulehead}[1]{\ssclause{#1}}`

`\singlefunctionhead` Starts a ‘function definition’ or a ‘function definitions’ subclause.
`\functionhead` 331 `\newcommand{\singlefunctionhead}[2]{\sclause{#1 function definition: #2}}`
332 `\newcommand{\functionhead}[1]{\sclause{#1 function definitions}}`

`\afunctionhead` Starts a ‘function definition’ subsubclause.
333 `\newcommand{\afunctionhead}[1]{\ssclause{#1}}`

`\shortnamehead` Starts a ‘Short names of entities’ normative annex
334 `\newcommand{\shortnamehead}{\normannex{Short names of entities}\label{;ssne}}`

`\objreghead` Starts a ‘Information object registration’ normative annex.
335 `\newcommand{\objreghead}{\normannex{Information object registration}\label{;sior}}`

`\docidhead` Starts a ‘Document identification’ subclause.
336 `\newcommand{\docidhead}{\sclause{Document identification}}`

`\schemidhead` Starts a ‘Schema identification’ subclause
337 `\newcommand{\schemaidhead}{\sclause{Schema identification}}`

`\aschemidhead` Starts a ‘Schema identification’ subsubclause
338 `\newcommand{\aschemaidhead}[1]{\ssclause{#1 identification}}`

`\expresshead` Starts an ‘EXPRESS listing’ informative annex
339 `\newcommand{\expresshead}{\infannex{EXPRESS listing}}`

`\listingshead` Starts a ‘Computer interpretable listings’ informative annex.
340 `\newcommand{\listingshead}{\infannex{Computer interpretable listings}\label{;scil}}`

`\expressghead` Starts a ‘EXPRESS-G diagrams’ informative annex
341 `\newcommand{\expressghead}{\infannex{EXPRESS-G diagrams}\label{;seg}}`

`\picshead` Starts a ‘Protocol Implementation Conformance Statement (PICS) proforma’ normative annex
342 `\newcommand{\picshead}{\normannex{Protocol Implementation`
343 `Conformance Statement (PICS) proforma}\label{;spics}}`

`\techdishead` Starts a ‘Technical discussions’ informative annex.
344 `\newcommand{\techdishead}{\infannex{Technical discussions}\label{;std}}`

`\exampleshead` Starts an ‘Examples’ informative annex
345 `\newcommand{\exampleshead}{\infannex{Examples}\label{;sex}}`
346

5.6.4 Miscellaneous boilerplate

`\expressgdef` Where EXPRESS-G is defined.
347 `\newcommand{\expressgdef}{\ExpressG{} is defined in annex~D of ISO 10303-11}`
348

`\mactableorspec` Depending on `\ifmapspec`, prints either ‘table’ or ‘specification’.
349 `\DeclareRobustCommand{\mactableorspec}{%`
350 `\ifmapspec specification\else table\fi}`
351

`\shortnames` Boilerplate for Short Name annex.
352 `\newcommand{\shortnames}{\input{bpfir1}}`
Here is the text of file `bpfir1.tex`.
353 `\step`
354 `*bpfir1`
355 `\ProvidesFile{bpfir1.tex}[1997/09/30 short names annex boilerplate]`
356 `\typeout{bpfir1.tex [1997/09/30 short names annex boilerplate]}`
357
358 Table A.1 provides the short names of entities specified in this
359 part of ISO~10303. Requirements on the use of short names are
360 found in the implementation methods included in ISO~10303.
361
362 `\bpfir1`
363 `*step`

```

\docreg Boilerplate for document registration annex. Use as:
  \docreg{version no}
364 \newcommand{\docreg}[1]{%
365   To provide for unambiguous identification of an information
366   object in an open system, the object identifier
367   \begin{center}
368   \{\~iso standard 10303 part(\thespartno) version(#1)\}
369   \end{center}
370   is assigned to this part of ISO~10303. The meaning of this value is defined
371   in ISO/IEC~8824-1, and is described in ISO~10303-1.
372 }
373

\schemareg Boilerplate for EXPRESS schema registration. Use as:
  \schemareg{version no}{underscored schema}{schema no}{hyphenated
  schema}{schema-name no}{clause/annex no}
374 \newcommand{\schemareg}[6]{%
375   To provide for unambiguous identification of the schema-name % #2
376   in an open information system, the object identifier
377   \begin{center}
378   \{\~iso standard 10303 part(\thespartno) version(#1) schema(#3) #4(#5)\}
379   \end{center}
380   is assigned to the \exp{#2} schema (see #6). The meaning of this
381   value is defined in ISO/IEC~8824-1, and is described in ISO~10303-1.
382 }
383

\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.
384 \newcommand{\expurls}[2]{\input{bpfir2}
385   Short names: \isourl{#1} \
386   \Express: \isourl{#2}
387   \input{bpfir3}}
388

      Here is the text of file bpfir2.tex
389 \step
390 \*bpfir2
391 \ProvidesFile{bpfir2.tex}[2002/01/22 IR short names and EXPRESS annex initial boilerplate]
392 \typeout{bpfir2.tex [2002/01/22 IR short names and EXPRESS annex initial boilerplate]}
393
394 This annex references a listing of the \Express{} entity data type
395 names and corresponding short names as specified in this part of ISO~10303.
396 It also references a listing of each \Express{} schema specified in
397 this part of ISO~10303, without comments or other explanatory text. These
398 listings are available in computer-interpretable form
399 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 These macros specify some standard normative references.
\nrefpartii 420 \newcommand{\nrefasni}{\isref{ISO/IEC 8824-1:1998}{%
\nrefpartxi 421     Information technology ---
\nrefpartxii 422     Abstract Syntax Notation One (ASN.1):
\nrefpartxxi 423     Specification of basic notation.}}
\nrefpartxxxii 424 \newcommand{\nrefpartii}{\isref{ISO 10303-1:1994}{%
\nrefpartxxxxi 425     Industrial automation systems and integration ---
\nrefpartxxxii 426     Product data representation and exchange ---
\nrefpartxli 427     Part 1: Overview and fundamental principles.}}
\nrefpartxlii 428 \newcommand{\nrefpartxi}{\isref{ISO 10303-11:1994}{%
\nrefpartxlili 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{\nrefpartxxxii}{\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{\nrefpartxxxii}{\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{\nrefpartxxxiii}{\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{\nrefpartxliia}{\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{\nrefpartxliiaa}{\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{\nrefpartxliiii}{\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 These macros specify some bibliographic references and the associated commands
\brefidfo to cite them in the text.
\bibidefix 489 \newcommand{\bibidefo}{\reference{}}{%
\bibieeidefix 490         IDEFO (ICAM Definition Language 0),}{%
\brefidefix

```

```

491         Federal Information Processing Standards Publication 183,
492         Integration Definition for Information Modeling (IDEFO),
493         FIPS PUB 183, National Institute for Standards and
494         Technology, December 1993.}\label{bibidefo}}
495 \newcommand{\brefidefo}{\bref{bibidefo}}
496 \newcommand{\bibidefix}{\reference{}{%
497     IDEF1X (ICAM Definition Language 1 Extended),}%
498     Federal Information Processing Standards Publication 184,
499     Integration Definition for Information Modeling (IDEF1X),
500     FIPS PUB 184, National Institute for Standards and
501     Technology, December 1993.}\label{bibidefix}}
502 \newcommand{\bibieeidefix}{\reference{IEEE Std 1320.2--1998,}%
503     Standard for Conceptual Modeling Language ---
504     Syntax and Semantics for IDEF1X.}\label{bibidefix}}
505 \newcommand{\brefidefix}{\bref{bibidefix}}
506

```

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.

```

507 \gdef\thest@tus{}
508 \ifisstandard
509   \gdef\thest@tus{}
510 \fi
511 \iffdisstandard
512   \gdef\thest@tus{/FDIS}
513 \fi
514 \ifdisstandard
515   \gdef\thest@tus{/DIS}
516 \fi
517 \ifcdstandard
518   \gdef\thest@tus{/CD}
519 \fi
520 \ifwdstandard
521   \gdef\thest@tus{/WD}
522 \fi
523 \iftechrep
524   \gdef\thest@tus{/TR}
525 \fi
526 \iftechspec
527   \gdef\thest@tus{/TS}
528 \fi
529 \ifpaspec
530   \gdef\thest@tus{/PAS}

```

```

531 \fi
532 \ifotherdoc
533   \gdef\thet@tus{}
534 \fi
535

```

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 The Working Group number. Use as \wg{WG the_number}.
\wg 536 \newcommand{\@wg}{-}
537 \newcommand{\wg}[1]{\def\@wg{\put(47,235){\Large\textbf{#1}}}}

\@docnumber Document number. Use as \docnumber{1234}.
\docnumber 538 \newcommand{\@docnumber}{-}
539 \newcommand{\docnumber}[1]{\def\@docnumber{\put(72,235){\Large\textbf{#1}}}}

\@docdate Document date. Use as \docdate{yyyy/mm/dd}.
\docdate 540 \newcommand{\@docdate}{-}
541 \newcommand{\docdate}[1]{\def\@docdate{\put(148,235){#1}}}

\@oldwg Developers of the immediately prior version of the document.
\oldwg 542 \newcommand{\@oldwg}{-}
543 \newcommand{\oldwg}[1]{\def\@oldwg{\put(58,227){\textbf{#1}}}}

\@olddocnumber The number of the immediately prior version of the document.
\olddocnumber 544 \newcommand{\@olddocnumber}{-}
545 \newcommand{\olddocnumber}[1]{\def\@olddocnumber{\put(78,227){\textbf{#1}}}}

\@abstract Document abstract. Use as \abstract{<text>}.
\abstract 546 \newcommand{\@abstract}{-}
547 \newcommand{\abstract}[1]{%
548   \def\@abstract{\put(2,110){\parbox[t]{161mm}{#1}}}
549 %% \def\@abstract{\put(2,120){\parbox[t]{161mm}{#1}}}

\@keywords Document keywords. Use as \keywords{<text>}.
\keywords 550 \newcommand{\@keywords}{-}
551 %% \newcommand{\keywords}[1]{\def\@keywords{\put(35,77){#1}}}
552 \newcommand{\keywords}[1]{\def\@keywords{\put(35,87){#1}}}

\@comread Document comments to the reader. Use as \comread{<text>}.
\comread 553 \newcommand{\@comread}{-}
554 \newcommand{\comread}[1]{%
555 %% \def\@comread{\put(2,65){\parbox[t]{161mm}{#1}}}
556 \def\@comread{\put(2,75){\parbox[t]{161mm}{#1}}}
557

```

```

\@owner Contact information for the document's project leader. Each of these commands
\owner takes a single text argument (e.g., \address{text}).
\@address 558 \newcommand{\@owner}{}
\address 559 \newcommand{\owner}[1]{\def\@owner{\put(35,45){#1}}}
\@telephone 560 \newcommand{\@address}{}
\telephone 561 \newcommand{\address}[1]{\def\@address{\put(22,40){\parbox[t]{59mm}{#1}}}
\@fax 562 \newcommand{\@telephone}{}
\fax 563 \newcommand{\telephone}[1]{\def\@telephone{\put(25,11){#1}}}
\@email 564 \newcommand{\@fax}{}
\email 565 %%\newcommand{\fax}[1]{\def\@fax{\put(30,6){#1}}}
566 \newcommand{\fax}[1]{\def\@fax{\put(25,6){#1}}}
567 \newcommand{\@email}{}
568 %%\newcommand{\email}[1]{\def\@email{\put(35,1){#1}}}
569 \newcommand{\email}[1]{\def\@email{\put(22,1){#1}}}
570

\@altowner Contact information for the document's editor.
\altowner 571 \newcommand{\@altowner}{}
\@altaddress 572 \newcommand{\altowner}[1]{\def\@altowner{\put(117.5,45){#1}}}
\altaddress 573 \newcommand{\@altaddress}{}
\@alttelephone 574 \newcommand{\altaddress}[1]{\def\@altaddress{\put(104.5,40){\parbox[t]{59mm}{#1}}}
\alttelephone 575 \newcommand{\@alttelephone}{}
\@altfax 576 \newcommand{\alttelephone}[1]{\def\@alttelephone{\put(107.5,11){#1}}}
\altfax 577 \newcommand{\@altfax}{}
\@altemail 578 %%\newcommand{\altfax}[1]{\def\@altfax{\put(112.5,6){#1}}}
\altemail 579 \newcommand{\altfax}[1]{\def\@altfax{\put(107.5,6){#1}}}
\@altemail 580 \newcommand{\@altemail}{}
581 %%\newcommand{\altemail}[1]{\def\@altemail{\put(117.5,1){#1}}}
582 \newcommand{\altemail}[1]{\def\@altemail{\put(104.5,1){#1}}}
583

\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}.
584 \newcommand{\STEPcover}[1]{%
Make sure that the internal commands are picked up.
585 #1
and call the routine to draw the picture.
586 \drawcoversheet
Put a copyright notice at the bottom of the next page.
587 \clearpage
588 \thispagestyle{startpage}
589 \mbox{}
590 \ifcopyright\@copyrighttext\fi
591 \newpage
592 }
593

```

```

\drawcoversheet This draws the STEP cover sheet.
594 \newcommand{\drawcoversheet}{%
    Make sure we have an empty page style.
595 \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.
596 \setlength{\unitlength}{1mm}
597 \begin{picture}(165,200)(0,40) %% actual size is (165,240)
598 \thicklines
    Revision notice for the cover sheet layout.
599 \put(165,-1){\makebox(0,0)[tr]{\tiny revision 8, 1/02 (PRW)}}
    Project leader information. (Box at y=0, height 50)
600 \put(0,0){\framebox(82.5,50){}}
601 \put(2,1){\bf E-mail:}
602 \put(2,6){\bf Facsimile:}
603 \put(2,11){\bf Telephone:}
604 \put(2,40){\bf Address:}
605 \put(2,45){\bf Project Leader:}
    Document editor information.
606 \put(82.5,0){\framebox(82.5,50){}}
607 \put(84.5,1){\bf E-mail:}
608 \put(84.5,6){\bf Facsimile:}
609 \put(84.5,11){\bf Telephone:}
610 \put(84.5,40){\bf Address:}
611 \put(84.5,45){\bf Project Editor:}
    Comments to reader box. (Box at y=50, height 35, total height 85)
612 %% \put(0,50){\framebox(165,25){}}
613 %% \put(2,70){\large\bf COMMENTS TO READER:}
614 \put(0,50){\framebox(165,35){}}
615 \put(2,80){\large\bf COMMENTS TO READER:}
    Draw abstract and keyword headings. (Box at y=85, height 35, total 120)
616 \put(0,85){\framebox(165,35){}}
617 \put(2,87){\large\bf KEYWORDS:}
618 \put(2,115){\large\bf ABSTRACT:}
619 %\put(0,85){\framebox(165,45){}}
620 %\put(2,87){\large\bf KEYWORDS:}
621 %\put(2,125){\large\bf ABSTRACT:}
    Do the copyright element. (Box at y=120, height 80, total 200)
622 \put(0,120){\framebox(165,80)[t]{
623 %%\put(0,130){\framebox(165,70)[t]{
624 \ifc@pyrightopt
625 \begin{minipage}{161mm}
626 \ifisstandard

```

```

627     \input{bpfs2} %% unknown at present
628     \fi
629     \iffdisstandard
630     \input{bpfs2}
631     \fi
632     \ifdisstandard
633     \input{bpfs2}
634     \fi
635     \ifcdstandard
636     \input{bpfs3}
637     \fi
638     \ifwdstandard
639     \input{bpfs3}
640     \fi
641     \iftechrep
642     \input{bpfs3} %% unknown at present
643     \fi
644     \end{minipage}
645 \else
646 %%% \put(2,195){\large\bf COPYRIGHT NOTICE:}
647     {\vspace*\baselineskip}
648     \textbf{\large\space COPYRIGHT NOTICE}\hfill\vspace*{\fill}}
649 \fi}}
650
        Draw the STEP title. (y=215 and 210)
651 \put(0,215){%
652   \ifnum\value{b@cyc} < 2
653     {\bf ISO\thest@tus\ 10303-\thespartno}
654   \else
655     {\bf ISO\thest@tus\ 10303-\thespartno.\theb@cyc}
656   \fi}
657 \put(0,210){\begin{minipage}[t]{165mm}
658   {\bf \st@pn@me: \Theseries: \thed@ctitle}
659 \end{minipage}}
        Identify the slots for the superseded document information.
660 \put(0,227){\bf Supersedes ISO TC 184/SC4/} % (y=227)
661 \put(67,226){\line(1,0){5}}
662 \put(73,227){\bf N}
663 \put(78,226){\line(1,0){8}}
        Draw the heading block
664 \put(0,235){\Large\bf ISO TC 184/SC4/} % (y=235)
665 \put(58,234){\line(1,0){7}}
666 \put(67,235){\Large\bf N}
667 \put(72,234){\line(1,0){11}}
        Identify the date slot.
668 \put(135,235){\bf Date:}

```

Finish off the picture. Note that this is where all the specific drawing commands are called.

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

Force printing of cover sheet, and remove the STEPcover internal commands as they are no longer needed.

```
675 \clearpage
676 \undef@covercmds
```

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

```
677 }
678
```

`\undef@covercmds` Make the `\STEPcover` internal commands undefined to make space for later macros, if necessary.

```
679 \newcommand{\undef@covercmds}{%
680 \let\@wg\relax \let\wg\relax
681 \let\@docnumber\relax \let\docnumber\relax
682 \let\@docdate\relax \let\docdate\relax
683 \let\@oldwg\relax \let\oldwg\relax
684 \let\@olddocnumber\relax \let\olddocnumber\relax
685 \let\@abstract\relax \let\abstract\relax
686 \let\@keywords\relax \let\keywords\relax
687 \let\@comread\relax \let\comread\relax
688 \let\@owner\relax \let\owner\relax
689 \let\@address\relax \let\address\relax
690 \let\@telephone\relax \let\telephone\relax
691 \let\@fax\relax \let\fax\relax
692 \let\@email\relax \let\email\relax
693 \let\@altowner\relax \let\altowner\relax
694 \let\@altaddress\relax \let\altaddress\relax
695 \let\@alttelephone\relax \let\alttelephone\relax
696 \let\@altfax\relax \let\altfax\relax
697 \let\@altemail\relax \let\altemail\relax
698 }
699
```

Here is the text of the file `bpfs2.tex`.

```
700 </step>
701 <*bpfs2>
702 \ProvidesFile{bpfs2.tex}[2002/01/10 STEP cover DIS+ copyright boilerplate]
703 \typeout{bpfs2.tex [2002/01/10 STEP cover DIS+ copyright boilerplate]}
704
705 \vspace*{\baselineskip}
706 \textbf{\large COPYRIGHT NOTICE}
```


707
708 \begin{small}
709 This ISO document is
710 \iffdisstandard
711 a Final Draft
712 \else
713 \ifdisstandard
714 a Draft
715 \else
716 an
717 \fi
718 \fi
719 International
720 Standard and is copyright protected by ISO. Except
721 as permitted under the applicable laws of the user's
722 country, neither this ISO draft nor any extract from
723 it may be reproduced, stored in a retrieval system or
724 transmitted in any form or by any means, electronic,
725 photocopying, recording, or otherwise, without prior
726 written permission being secured.
727
728 Requests for permission to reproduce should be addressed
729 to ISO at the address below or ISO's member body in the
730 country of the requester:
731 \begin{center}
732 ISO copyright office \\
733 Case postale 56. CH-1211 Geneva 20 \\
734 Tel. +41 22 749 01 11 \\
735 Fax +41 22 734 01 79 \\
736 E-mail \texttt{copyright@iso.ch}
737 \end{center}
738 Reproduction for sales purposes for any of the above-mentioned
739 documents may be subject to royalty payments or a licensing
740 agreement.
741
742 Violators may be prosecuted.
743
744 \end{small}
745
746 </bpf2>
747 <*step>

748 </step>
749 <*bpf3>
750 \ProvidesFile{bpfs3.tex}[2002/01/10 STEP cover WD/CD copyright boilerplate]
751 \typeout{bpfs3.tex [2002/01/10 STEP cover WD/CD copyright boilerplate]}
752
753 \vspace*{\baselineskip}
754 \textbf{\large COPYRIGHT NOTICE}
755

```

756 \begin{small}
757 This ISO document is a working draft or Committee Draft
758 and is copyright protected by ISO.
759 While the reproduction of working drafts or Committee Drafts
760 in any form for use by Participants in the ISO standards
761 development process is permitted without prior permission
762 from ISO, neither this document nor any extract from
763 it may be reproduced, stored or
764 transmitted in any form for any other purpose without prior
765 written permission from ISO.
766
767 Requests for permission to reproduce this document for the
768 purposes of selling it should be addressed as shown below
769 (via the ISO TC 184/SC4 Secretariat's member body)
770 or to ISO's member body in the
771 country of the requester:
772 \begin{center}
773 Copyright Manager \\
774 ANSI \\
775 11 West 42nd Street \\
776 New York, New York 10036 \\
777 USA \\
778 phone: +1--212--642--4900 \\
779 fax: +1--212--398--0023
780 \end{center}
781 Reproduction for sales purposes may be subject to royalty payments
782 or a licensing agreement.
783
784 Violators may be prosecuted.
785
786 \end{small}
787
788 </bpf3>
789 <*step>

```

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

```

790 \newcommand{\draftctr}{Recipients of this draft are invited to submit,
791 with their comments, notification of any relevant patent rights of
792 which they are aware and to provide supporting documentation. }
793

```

The end of this package.

```

794 </step>

```

6 The Integrated Resources package

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

```

795 <*ir>

```

`\anirtrue` We are meant to be processing an IR.

```
796 \anirtrue
797
```

6.1 Boilerplate

This section defines the commands used to print boilerplate text.

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

```
\irexpressg
798 \newcommand{\irexpressg}{%
799   The diagrams in this annex correspond to the \Express{} schemas
800   specified in this part of ISO~10303. The
801   diagrams use the \ExpressG{} graphical notation for the
802   \Express{} language. \expressgdef.
803 }
804
```

The end of this package.

```
805 </ir>
```

7 The Application Protocol package

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

```
806 <*ap>
```

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

```
807 \anirfalse
```

In general, the ToC should contain subclauses.

```
808 \settocdepth{sclause}
809
```

7.1 Preamble commands

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

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

```
\theap 810 \gdef\theap{}
811 \newcommand{\aptitle}[1]{\gdef\theap{#1}}
```

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

```
812 \newif\ifaicinap
813 \aicinapfalse
```

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

```

814 \newif\ifmaptemplate
815 \maptemplatefalse

```

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

```

816 \newif\ifidefix
817 \idefixfalse
818

```

7.2 Heading commands

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

`\inforeqhead` Starts a ‘Information requirements’ clause. N200 says that subsubclauses of this should be in the ToC.

```

819 \newcommand{\inforeqhead}{%
820 \settocdepth{ssclause}
821 \clause{Information requirements}\label{;sireq}}

```

`\uofhead` Starts a ‘Units of functionality’ subclause

```

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

```

`\auofhead` Starts a subsubclause for a UoF

```

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

```

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

```

825 \newcommand{\applobjhead}{%
826 \settocdepth{sclause}
827 \sclause{Application objects}\label{;sao}}

```

`\applasserthead` Starts a ‘Application assertions’ subclause

```

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

```

`\aimhead` Starts a ‘Application interpreted model’ clause

```

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

```

`\mappinghead` Starts a ‘Mapping table’ or ‘Mapping specification’ subclause

```

832 \newcommand{\mappinghead}{%
833 \sclause{Mapping \mactableorspec}\label{;smap}}

```

`\templateshead` Starts a ‘Mapping templates’ subsubclause.

```

834 \newcommand{\templateshead}{%
835 \ssclause{Mapping templates}\label{;stemp}}

```

`\mapuofhead` Starts a UoF mapping subsubclause.
836 `\newcommand{\mapuofhead}[1]{\ssclause{#1}}`

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

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

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

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

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

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

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

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

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

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

`\armhead` Starts a ‘Application reference model’ informative annex
857 `\newcommand{\armhead}{%`
858 `\settocdepth{sclause}`
859 `\infannex{Application reference model}\label{;sarm}}`

`\aimexpressghead` Starts a ‘AIM EXPRESS-G’ informative annex
860 `\newcommand{\aimexpressghead}{%`
861 `\infannex{AIM EXPRESS-G}\label{;saeg}}`

`\aimexpresshead` Starts a ‘AIM EXPRESS listing’ informative annex
862 `\newcommand{\aimexpresshead}{%`
863 `\infannex{AIM EXPRESS listing}}`

`\apusagehead` Starts a ‘Application protocol usage guide’ informative annex
864 `\newcommand{\apusagehead}{%`
865 `\infannex{Application protocol usage guide}\label{;sapug}}`
866

7.2.1 Template headings

`\signature` The ‘mapping signature’ heading.
867 `\newcommand{\signature}{\ehe@d*{\underline{Mapping signature}}}`

`\parameters` The ‘parameter definitions’ heading.
868 `\newcommand{\parameters}{\ehe@d*{\underline{Parameter definitions}}}`

`\body` The ‘template body’ heading.
869 `\newcommand{\body}{\ehe@d*{\underline{Template body}}}`
870

7.3 Boilerplate printing

`\apextraintro` Print boilerplate for end of AP introduction clause.
871 `\newcommand{\apextraintro}{\input{apendint}}`
Here is the text of `apendint.tex`.
872 `</ap>`
873 `<*apf1>`
874 `\ProvidesFile{apendint.tex}[1996/05/31 AP end intro boilerplate]`
875 `\typeout{apendint.tex [1996/05/31 AP end intro boilerplate]}`
876
877 Application protocols provide the basis for developing
878 implementations of ISO~10303 and abstract test suites for
879 the conformance testing of AP implementations.
880
881 Clause~\ref{;i1} defines the scope of the application protocol
882 and summarizes the functionality and data covered by the AP.
883 Clause~\ref{;i3} lists the words defined in this part of ISO~10303 and
884 gives pointers to words defined elsewhere.
885 An application activity model that is the basis for the definition
886 of the scope is provided in \aref{;saam}. The information requirements
887 of the application are specified in \cref{;sireq} using terminology
888 appropriate to the application. A graphical representation of the

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

891

892 Resource constructs are interpreted to meet the information
893 requirements. This interpretation produces the application
894 interpreted model (AIM). This interpretation, given in \ref{smap}, shows
895 the correspondence between the information requirements and the
896 AIM. The short listing of the AIM specifies the interface to the
897 integrated resources and is given in \ref{saesl}. Note that the definitions
898 and \Express{} provided in the integrated resources for constructs
899 used in the AIM may include select list items and subtypes which are
900 not imported into the AIM. The expanded listing given in \aref{saeel}
901 contains the complete \Express{} for the AIM without annotation. A
902 graphical representation of the AIM is given in \aref{saeg}. Additional
903 requirements for specific implementation methods are given in
904 \aref{simreq}.

905

906 </apf1>

907 <*ap>

\apscope Print boilerplate for start of AP scope clause.

\apscope{*application purpose and context*}

908 \newcommand{\apscope}[1]{%

909 This part of ISO 10303 specifies the use of the integrated
910 resources necessary for the scope and information requirements
911 for #1

912

913 \input{bpfap1}

914

915 }

Here is the text for file bpfap1.tex

916 </ap>

917 <*bpfap1>

918 \ProvidesFile{bpfap1.tex}[2001/07/16 AP start scope clause boilerplate]

919 \typeout{bpfap1.tex [2001/07/16 AP start scope clause boilerplate]}

920

921 \begin{anote}The application activity model in \aref{saam} provides a
922 graphical representation of the processes and
923 information flows that are the basis for the definition
924 of the scope of this part of ISO~10303.\end{anote}

925

926 </bpfap1>

927 <*ap>

\apinforeq Print boilerplate for start of AP clause on information requirements.

\apinforeq{*AP purpose*}

928 \newcommand{\apinforeq}[1]{%

929 This clause specifies the information required for #1

930

```

931 \input{bpfap2}
932
933 }
    Here is the text for file bpfap2.tex.
934 </ap>
935 <*bpfap2>
936 \ProvidesFile{bpfap2.tex}[2001/07/16 AP info boilerplate]
937 \typeout{bpfap2.tex [2001/07/16 AP info boilerplate]}
938
939 The information requirements are specified as a set of
940 units of functionality, application objects, and
941 application assertions. These assertions pertain to
942 individual application objects and to relationships
943 between application objects. The information requirements
944 are defined using the terminology of the subject area of
945 this application protocol.
946
947 \begin{note}A graphical representation of the information
948 requirements is given in \aref{;sarm}.\end{note}
949 \begin{note}The information requirements correspond to those of
950 the activities identified as being within the scope of this
951 application protocol in \aref{;saam}.\end{note}
952 \begin{note}The mapping \mactableorspec{}
953 specified in \ref{;smap} shows how the
954 integrated resources
955 \ifaicinap and application interpreted constructs \fi
956 are used to meet the information requirements of this
957 application protocol. \end{note}
958
959 </bpfap2>
960 <*ap>

```

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

```

961 \newenvironment{apuof}{%
962 This subclause specifies the units of functionality for the
963 \thead\space application protocol. This part of ISO~10303
964 specifies the following units of functionality:
965 \begin{itemize}}{%
966 \end{itemize}
967
968 \input{bpfap3}
969
970 }

```

Here is the text for file bpfap3.tex.

```

971 </ap>
972 <*bpfap3>

```



```

973 \ProvidesFile{bpfap3.tex}[1997/09/30 AP uof boilerplate]
974 \typeout{bpfap3.tex [1997/09/30 AP uof boilerplate]}
975
976 The units of functionality and a description of the functions
977 that each UoF supports are given below. The application objects
978 included in the UoFs are defined in~\ref{sao}.
979
980 </bpfap3>
981 <*ap>

```

`\apapplobj` Print boilerplate for Application objects.

```

982 \newcommand{\apapplobj}{\input{bpfap4}}
    Here is the text for file bpfap4.tex
983 </ap>
984 <*bpfap4>
985 \ProvidesFile{bpfap4.tex}[1997/09/30 AP application objects boilerplate]
986 \typeout{bpfap4.tex [1997/09/30 AP application objects boilerplate]}
987
988 This subclause specifies the application objects for
989 the \thead\space application protocol. Each application
990 object is an atomic element that embodies a unique
991 application concept and contains attributes specifying
992 the data elements of the object. The application objects
993 and their definitions are given below.
994
995 </bpfap4>
996 <*ap>

```

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

```

997 \newcommand{\apassert}{\input{bpfap5}}
    Here is the text for file bpfap5.tex
998 </ap>
999 <*bpfap5>
1000 \ProvidesFile{bpfap5.tex}[1997/09/30 AP application assertions boilerplate]
1001 \typeout{bpfap5.tex [1997/09/30 AP application assertions boilerplate]}
1002
1003 This subclause specifies the application assertions for the
1004 \thead\space application protocol. Application assertions
1005 specify the relationships between application objects,
1006 the cardinality of the relationships, and the rules required
1007 for the integrity and validity of the application objects and
1008 UoFs. The application assertions and their definitions are
1009 given below.
1010
1011 </bpfap5>
1012 <*ap>

```

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

```
1013 \newcommand{\apmapping}{%
1014   \ifmapspec \input{apmpspec} \else \input{apmptbl} \fi
    Here is the contents of the apmptbl.tex file.
1015 \ap AP: boilerplate;
1016 \*apmptbl
1017 \ProvidesFile{apmptbl.tex}[2002/01/22 AP mapping table boilerplate]
1018 \typeout{apmptbl.tex [2002/01/22 STEP AP mapping table boilerplate]}
1019
1020 This clause contains the mapping table that shows how each
1021 UoF and application object of this part of ISO~10303
1022 (see \cref{sireq}) maps to one or more AIM constructs
1023 (see \aref{saeel}).
1024 The mapping table is organized in five columns.
1025
1026
1027 Column 1) Application element: Name of an application
1028     element as it appears in the application object definition
1029     in~\ref{sao}. Application object names are written in uppercase.
1030     Attribute names and assertions are listed after the application
1031     object to which they belong and are written in lower case.
1032
1033 Column 2) AIM element: Name of an AIM element as it
1034     appears in the AIM (see \aref{saeel}), the term ‘IDENTICAL MAPPING’,
1035     or the term ‘PATH’. AIM entities are written in lower case.
1036     Attribute names of AIM entities are referred to as
1037     <entity name>$.<attribute name>$. The mapping of an
1038     application element may result in several related AIM
1039     elements. Each of these AIM elements requires a line of its
1040     own in the table. The term ‘IDENTICAL MAPPING’ indicates
1041     that both application objects of an application assertion
1042     map to the same AIM element. The term ‘PATH’ indicates
1043     that the application assertion maps to the entire reference
1044     path.
1045
1046 Column 3) Source: For those AIM elements that are
1047     interpreted from the integrated resources or the application
1048     interpreted constructs, this is the
1049     number of the corresponding part of ISO~10303. For those
1050     AIM elements that are created for the purpose of this part
1051     of ISO~10303, this is the number of this part.
1052     Entities or types that are defined within the integrated
1053     resources have an AIC as the source reference if the use
1054     of the entity or type for the mapping is within the scope
1055     of the AIC.
1056
1057 Column 4) Rules: One or more numbers may be given that
1058     refer to rules that apply to the current AIM element or
1059     reference path. For rules that are derived from
```

1060 relationships between application objects, the same rule
1061 is referred to by the mapping entries of all the involved AIM
1062 elements. The expanded names of the rules are listed after
1063 the table.

1064

1065 Column 5) Reference path: To describe fully the mapping
1066 of an application object, it may be necessary to specify a
1067 reference path through several related AIM elements. The
1068 reference path column documents the role of an AIM element
1069 relative to the AIM element in the row succeeding it.
1070 Two or more such related AIM elements define the
1071 interpretation of the integrated resources that satisfies
1072 the requirement specified by the application object.
1073 For each AIM element that has been created for use within this
1074 part of ISO~10303, a reference path up to its supertype from
1075 an integrated resource is specified.

1076

1077

1078 For the expression of reference paths the following notational
1079 conventions apply:

1080 \begin{enumerate}

1081 \item \verb|[]| : enclosed section constrains multiple AIM elements
1082 or sections of the
1083 reference path are required to satisfy an information
1084 requirement;

1085 \item \verb|()| : enclosed section constrains multiple AIM elements
1086 or sections of the
1087 reference path are identified as alternatives within the
1088 mapping to satisfy an information requirement;

1089 \item \verb|{}| : enclosed section constrains the reference path
1090 to satisfy an information requirement;

1091 \item \verb|<>| : enclosed section constrains at one or more
1092 required reference path;

1093 \item \verb|++| : enclosed section constrains the supertype entity;

1094 \item \verb|->| : attribute references the entity or select type
1095 given in the following row;

1096 \item \verb|<-| : entity or select type is referenced by the
1097 attribute in the following row;

1098 \item \verb|[i]| : attribute is an aggregation of which a
1099 single member is given in the following row;

1100 \item \verb|[n]| : attribute is an aggregation of which
1101 member \verb|n| is given in the following row;

1102 \item \verb|=>| : entity is a supertype of the entity given in the
1103 following row;

1104 \item \verb|<=| : entity is a subtype of the entity given in
1105 the following row;

1106 \item \verb|=| : the string, select, or enumeration type is
1107 constrained to a choice or value;

1108 \item \verb|\| : the reference path expression continues on
1109 the next line;

```

1110 \item \verb|*| : used in conjunction with braces to indicate that any
1111     number of relationship entity data types may be assembled in a
1112     relationship tree structure.
1113 \end{enumerate}
1114
1115 </apmptbl>
1116 <*ap>

```

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

```

1117 \newcommand{\apmappingspec}{\input{apmspec}}
      Here is the contents of the apmspec.tex file.
1118 </ap>
1119 <*apmspec>
1120 \ProvidesFile{apmspec.tex}[2001/07/16 AP mapping spec boilerplate]
1121 \typeout{apmspec.tex [2001/07/16 STEP AP mapping spec boilerplate]}
1122
1123 This clause contains the mapping specification that shows how each
1124 UoF and application object of this part of ISO~10303
1125 (see \cref{;sireq}) maps to one or more AIM constructs
1126 (see \aref{;saeel}).
1127 Each mapping specifies up to five elements.
1128
1129 \begin{description}
1130 \item[Application element] The mapping for each application element
1131     is specified in a seperate subclause below.
1132     Application object names are given in title case.
1133     Attribute names and assertions are listed after the application
1134     object to which they belong and are given in lower case.
1135
1136 \item[AIM element] The name of one or more AIM entity data types
1137     (see \aref{;saeel}), the term ‘‘IDENTICAL MAPPING’’,
1138     or the term ‘‘PATH’’.
1139     AIM entity data type names are given in lower case.
1140     Attributes of AIM entity data types are referred to as
1141     <$entity name$>$.<$attribute name$>$.
1142     The mapping of an application element may involve more than
1143     one AIM element.
1144     Each of these AIM elements is presented on a seperate line
1145     in the mapping specification.
1146     The term ‘‘IDENTICAL MAPPING’’ indicates that both application
1147     objects involved in an application assertion map to the same
1148     instance of an AIM entity data type.
1149     The term ‘‘PATH’’ indicates that the application assertion maps
1150     to a collection of related AIM entity instances specified
1151     by the entire reference path.
1152
1153 \item[Source] For those AIM elements that are
1154     interpreted from any common resource, this is the ISO standard
1155     number and part number in which the resource is defined.

```

1156 For those AIM elements that are created for the purpose of this part
1157 of ISO~10303, this is ‘‘ISO~10303--’’ followed by the number of
1158 this part.
1159

1160 \item[Rules] One or more global rules may be specified that
1161 apply to the population of the AIM entity data types specified
1162 as the AIM element or in the reference path.
1163 For rules that are derived from
1164 relationships between application objects, the same rule
1165 is referred to by the mapping entries of all the involved AIM
1166 elements.
1167 A reference to a global rule may be accompanied by a reference to
1168 the subclause in which the rule is defined.
1169

1170 \item[Reference path] To describe fully the mapping
1171 of an application object, it may be necessary to specify a
1172 reference path involving several related AIM elements.
1173 Each line in the reference path documents the role of an AIM
1174 element relative to the AIM element in the line following it.
1175 Two or more such related AIM elements define the
1176 interpretation of the integrated resources that satisfies
1177 the requirement specified by the application object.
1178 For each AIM element that has been created for use within this
1179 part of ISO~10303, a reference path to its supertype from
1180 an integrated resource is specified.
1181 For the expression of reference paths and the relationships
1182 between AIM elements the following notational conventions apply:
1183 \begin{itemize}

1184 \item[\texttt{[]}] enclosed section constrains multiple AIM elements
1185 or sections of the
1186 reference path are required to satisfy an information
1187 requirement;

1188 \item[\texttt{()}] enclosed section constrains multiple AIM elements
1189 or sections of the
1190 reference path are identified as alternatives within the
1191 mapping to satisfy an information requirement;

1192 \item[\texttt{\{\}}] enclosed section constrains the reference path
1193 to satisfy an information requirement;

1194 \item[\texttt{<>}] enclosed section constrains at one or more
1195 required reference path;

1196 \item[\texttt{||}] enclosed section constrains the supertype entity;

1197 \item[\texttt{->}] attribute references the entity or select type
1198 given in the following row;

1199 \item[\texttt{<-}] entity or select type is referenced by the
1200 attribute in the following row;

1201 \item[\texttt{[i]}] attribute is an aggregation of which a
1202 single member is given in the following row;

1203 \item[\texttt{[n]}] attribute is an aggregation of which
1204 member \texttt{n} is given in the following row;

1205 \item[\texttt{=>}] entity is a supertype of the entity given in the

```

1206     following row;
1207 \item[\texttt{<=}] entity is a subtype of the entity given in
1208     the following row;
1209 \item[\texttt{=}] the string, select, or enumeration type is
1210     constrained to a choice or value;
1211 \item[\texttt{\textbackslash}] the reference path expression continues on
1212     the next line;
1213 \item[\texttt{*}] used in conjunction with braces to indicate that
1214     any number of relationship entity data types may be assembled
1215     in a relationship tree structure;
1216 \ifmaptemplate
1217 \item[\texttt{//}] enclosed section is an application of one of the
1218     mapping templates defined in \ref{;stemp} below;
1219 \fi
1220 \item[\texttt{--}] the text following is a comment
1221     (normally a clause reference).
1222 \end{itemize}
1223
1224 \end{description}
1225
1226 </apmspec>
1227 <*ap>

```

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

```

1228 \newcommand{\apmaptemplate}{\input{apmptempl}}
1229     Here is the contents of the apmptempl.tex file.
1230 </ap>
1231 <*apmptempl>
1232 \ProvidesFile{apmptempl.tex}[2001/07/16 AP mapping template boilerplate]
1233 \typeout{apmptempl.tex [2001/07/16 STEP AP mapping template boilerplate]}
1234 This mapping specification includes mapping templates.
1235 A mapping template is a reusable portion of a reference path that defines
1236 a commonly used part of the structure of the application interpreted model.
1237 A mapping template is similar to a programming language macro.
1238 The mapping templates used in this part of ISO~10303 are defined in this
1239 subclause. Each mapping template definition has three components as follows:
1240 \begin{itemize}
1241 \item the template signature that specifies the name of the template
1242     and may also specify the names and the order of the formal parameters
1243     of the template;
1244
1245 \item descriptions of the formal parameters of the template, if any;
1246
1247 \item the template body that defines the reusable portion of a reference
1248     path and may indicate, through the use of the formal parameter
1249     names included in the template signature, the points at which
1250     the value parameters are supplied in each template application.
1251 \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 substituted 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, separated 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

```

1302 </apmptempl>
1303 <*ap>

\sstemplates A macro for the boilerplate text for SUBTYPE and SUPERTYPE templates.
1304 \newcommand{\sstemplates}{\input{apsstempl.tex}}

    Here is the text for the file apsstempl.tex.
1305 </ap>
1306 <*apsstempl>
1307 \ProvidesFile{apsstempl.tex}[2001/07/16 AP SUP/SUB templates boilerplate]
1308 \typeout{apsstempl.tex [2001/07/16 AP SUP/SUB templates boilerplate]}
1309
1310 \sssclause{SUBTYPE}
1311
1312     The SUBTYPE mapping template specifies a reference to the mapping of
1313 a subtype of the current application object. Several such references may
1314 be included for one supertype application object.
1315
1316 \begin{anote} This template definition only consists of a template signature,
1317     there is no matching template body. The template is included to ease the
1318     automatic processing of the mapping specification.
1319 \end{anote}
1320
1321 \signature
1322
1323 /SUBTYPE(application\_object)/
1324
1325 \parameters
1326
1327 application\_object: the application object that is a subtype of the current
1328     supertype application object and that has the entire
1329     or a part of the mapping specification of this
1330     supertype.
1331
1332
1333 \sssclause{SUPERTYPE}
1334
1335     The SUPERTYPE mapping template specifies a reference to the mapping of
1336 a supertype of the current application object. Several such references may
1337 be included for the subtype application object.
1338
1339 \begin{anote} This template only consists of a signature,
1340     there is no matching body. The template is included to ease the
1341     automatic processing of the mapping specification.
1342 \end{anote}
1343
1344 \signature
1345
1346 /SUPERTYPE(application\_object)/
1347

```



```

1348 \parameters
1349
1350 application\_object: the application object that is a supertype of the current
1351 subtype application object and that has the entire
1352 or a part of the mapping specification of this
1353 subtype.
1354
1355 </apsstempl>
1356 <*ap>

```

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

```

1357 \newcommand{\apshortexpress}{\input{bpfap6}}
      Here is the text of file bpfap6.tex
1358 </ap>
1359 <*bpfap6>
1360 \ProvidesFile{bpfap6.tex}[2002/01/22 AP AIM EXPRESS short listing boilerplate]
1361 \typeout{bpfap6.tex [2002/01/22 AP AIM EXPRESS short listing boilerplate]}
1362
1363 This clause specifies the \Express{} schema that uses
1364 elements from the integrated resources
1365 \ifaicinap and the AICs \fi
1366 and contains the types, entity specializations, rules,
1367 and functions that are specific to this part of ISO~10303.
1368 This clause also specifies modifications to the text
1369 for constructs that are imported from the
1370 integrated
1371 \ifaicinap resources and the AICs. \else resources. \fi
1372 The definitions and
1373 \Express{} provided in the integrated resources for constructs
1374 used in the AIM may include select list items and subtypes
1375 that are not imported into the AIM. Requirements stated
1376 in the integrated resources that refer to select list items and
1377 subtypes apply exclusively to those items that are imported
1378 into the AIM.
1379
1380 </bpfap6>
1381 <*ap>

```

`\apconformance` Print boilerplate for AP conformance.

```

      \apconformance{(implentation methods)}
1382 \newcommand{\apconformance}[1]{%
1383
1384 \input{bpfap7}
1385
1386 An implementation shall support at least one of the following
1387 implementation methods: #1.
1388
1389 \input{bpfap8}
1390

```

```

1391 }
      Here is the text of file bpfap7.tex
1392 </ap>
1393 <*bpfap7>
1394 \ProvidesFile{bpfap7.tex}[1997/09/30 AP conformance boilerplate (1)]
1395 \typeout{bpfap7.tex [1997/09/30 AP conformance boilerplate (1)]}
1396
1397 Conformance to this part of ISO 10303 includes satisfying
1398 the requirements stated in this part, the requirements of
1399 the implementation method(s) supported, and the relevant
1400 requirements of the normative references.
1401
1402 </bpfap7>
1403 <*ap>

```

```

      Here is the text of file bpfap8.tex
1404 </ap>
1405 <*bpfap8>
1406 \ProvidesFile{bpfap8.tex}[1997/09/30 AP conformance boilerplate (2)]
1407 \typeout{bpfap8.tex [1997/09/30 AP conformance boilerplate (2)]}
1408
1409 Requirements with respect to implementation methods-specific
1410 requirements are specified in \aref{;simreq}.
1411
1412 The Protocol Information Conformance Statement (PICS)
1413 proforma lists the options or the combination of options
1414 that may be included in the implementation. The PICS
1415 proforma is provided in \aref{;spics}.
1416
1417 </bpfap8>
1418 <*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.

```

1419 \newenvironment{apconformclasses}{%
1420 This part of ISO~10303 provides for a number of options that
1421 may be supported by an implementation. These options have been
1422 grouped into the following conformance classes:
1423 \begin{itemize}}{%
1424 \end{itemize}
1425 Support for a particular conformance class requires support of
1426 all the options specified in this class.
1427
1428 }

```

\apshortnames Print boilerplate for AP short names.
`\newcommand{\apshortnames}{\input{bpfap9}}`

Here is the text of file `bpfap9.tex`

```
1430 </ap>
1431 <*bpfap9>
1432 \ProvidesFile{bpfap9.tex}[1997/09/30 AP short names boilerplate]
1433 \typeout{bpfap9.tex [1997/09/30 AP short boilerplate]}
1434
1435 Table B.1 provides the short names of entities specified
1436 in the AIM of this part of ISO~10303. Requirements on the
1437 use of the short names are found in the implementation methods
1438 included in ISO~10303.
1439
1440 </bpfap9>
1441 <*ap>
```

`\picsannex` Print boilerplate for PICS annex.

```
1442 \newcommand{\picsannex}{\input{bpfap10}}
    Here is the text of file bpfap10.tex
1443 </ap>
1444 <*bpfap10>
1445 \ProvidesFile{bpfap10.tex}[1997/09/30 AP PICS annex boilerplate]
1446 \typeout{bpfap10.tex [1997/09/30 AP PICS annex boilerplate]}
1447
1448 This clause lists the optional elements of this part
1449 of ISO~10303. An implementation may choose to support
1450 any combination of these optional elements. However,
1451 certain combinations of options are likely to be
1452 implemented together. These combinations are called
1453 conformance classes and are described in the subclauses
1454 of this annex.
1455
1456 This annex is in the form of a questionnaire. This
1457 questionnaire is intended to be filled out by the
1458 implementor and may be used in preparation for conformance
1459 testing by a testing laboratory. The completed PICS proforma
1460 is referred to as a PICS.
1461
1462 </bpfap10>
1463 <*ap>
```

`\aamfigrs` 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.

```

1464 \gdef\aamfigrs{figure F.1}
1465 \newcommand{\aamfigrange}[1]{\gdef\aamfigrs{#1}}

\apaamintro Print boilerplate for AAM annex intro.
1466 \newcommand{\apaamintro}{\input{bpfap11}}
    Here is the contents of bpfap11.tex. Note the use of the \aamfigrs command.
1467 </ap>
1468 <*bpfap11>
1469 \ProvidesFile{bpfap11.tex}[2001/07/16 AP AAM annex intro boilerplate]
1470 \typeout{bpfap11.tex [2001/07/16 AP AAM annex intro boilerplate]}
1471
1472 The application activity model (AAM) is provided as an aid
1473 in understanding the scope and information requirements
1474 defined in this application protocol. The model is presented
1475 as a set of figures that contain the activity
1476 diagrams and a set of definitions of the activities
1477 and their data.
1478 %%% The application activity model is given in \aamfigrs.
1479 Activities and data flows that are out of scope are marked with
1480 an asterisk.
1481
1482 </bpfap11>
1483 <*ap>

\apaamdefs Print boilerplate for AAM definitions.
1484 \newcommand{\apaamdefs}{\input{bpfap12}}
    Here is the text of file bpfap12.tex
1485 </ap>
1486 <*bpfap12>
1487 \ProvidesFile{bpfap12.tex}[1997/09/30 AP AAM definitions boilerplate]
1488 \typeout{bpfap12.tex [1997/09/30 AP AAM definitions boilerplate]}
1489
1490 The following terms are used in the application
1491 activity model. Terms marked with an asterisk are outside
1492 the scope of this application protocol.
1493
1494 The definitions given in this annex do not supersede
1495 the definitions given in the main body of the text.
1496
1497 </bpfap12>
1498 <*ap>

\aamfigures Print boilerplate for AAM figures. \aamfigures
1499 \newcommand{\aamfigures}{\input{bpfap15}}
    Here is the contents of bpfap15.tex.
1500 </ap>

```

```

1501 <*bpfap15>
1502 \ProvidesFile{bpfap15.tex}[2001/07/16 AP AAM annex figures subclause boilerplate]
1503 \typeout{bpfap15.tex [2001/07/16 AP AAM annex figures subclause boilerplate]}
1504
1505 The application activity model diagrams are given in \aamfigrs. The
1506 graphical form of the application activity model is
1507 presented in the IDEF0 activity modelling format \brefidefo.
1508 Activities and data flows that are out of scope are
1509 marked with asterisks.
1510
1511 </bpfap15>
1512 <*ap>

```

`\armintro` Print boilerplate for ARM introduction.

```

1513 \newcommand{\armintro}{%
1514 This annex provides the application reference model for this part of ISO
1515 10303. The application reference model is a graphical
1516 representation of the structure and constraints of the application objects
1517 specified in \cref{sireq}. The graphical form of the application reference
1518 model is presented in \ifidefix IDEF1X. \else \ExpressG. \fi
1519 The application reference model is
1520 independent from any implementation method.
1521 \ifidefix The diagrams use the IDEF1X graphical notation~\brefidefix.
1522 \else \expressgdef. \fi
1523
1524 }

```

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

```

1525 \newcommand{\aimexpressg}{%
1526 The diagrams in this annex correspond to the AIM \Express{} expanded
1527 listing given in \aref{saeel}.
1528 The diagrams use the \ExpressG{} graphical notation for the
1529 \Express{} language. \expressgdef.
1530
1531 }

```

`\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.

```

1532 \newcommand{\apexpurls}[2]{\input{bpfap13}
1533
1534 \begin{itemize}
1535 \item Short names: \isourl{#1}
1536 \item \Express: \isourl{#2}
1537 \end{itemize}
1538
1539 \input{bpfap16}}

```

Here is the text of file bpfap13.tex

```
1540 </ap>
1541 <*bpfap13>
1542 \ProvidesFile{bpfap13.tex}[2001/07/16 AP short names/EXPRESS listing boilerplate (1)]
1543 \typeout{bpfap13.tex [2001/07/16 AP short names/EXPRESS listing boilerplate (1)]}
1544
1545 This annex provides a listing of the complete \Express{} schema
1546 specified in \aref{;saeel} of this part of ISO~10303 without comments
1547 or explanatory text. It also provides a listing of the \Express{} entity
1548 names and corresponding short names as specified in \aref{;sasn}
1549 of this part of ISO~10303. The content of this annex is available
1550 in computer-interpretable form and can be found at the following URLs:
1551
1552 </bpfap13>
1553 %
1554 % Here is the text of \file{bpfap16.tex}.
1555 % \changes{v1.3}{1999/02/15}{Added file bpfap16.tex}
1556 <*bpfap16>
1557 \ProvidesFile{bpfap16.tex}[1999/02/15 AP short names and EXPRESS annex ending boilerplate]
1558 \typeout{bpfap16.tex [1997/09/30 AP short names and EXPRESS annex ending boilerplate]}
1559
1560 If there is difficulty accessing these sites contact ISO Central Secretariat or
1561 contact the ISO TC~184/SC4 Secretariat directly at: \url{sc4sec@cme.nist.gov}.
1562
1563 \begin{anote}The information provided in computer-interpretable form at the above
1564 URLs is informative. The information that is contained in the body of this
1565 part of ISO~10303 is normative.
1566 \end{anote}
1567
1568 </bpfap16>
1569 <*ap>
```

\aimlongexp Print boilerplate for AIM EXPRESS expanded listing.

```
1570 \newcommand{\aimlongexp}{\input{bpfap14}}
1571
1572 Here is the text of file bpfap14.tex
1573 </ap>
1574 <*bpfap14>
1575 \ProvidesFile{bpfap14.tex}[1997/09/30 AP AIM EXPRESS expanded listing boilerplate]
1576 \typeout{bpfap14.tex [1997/09/30 AP AIM EXPRESS expanded listing boilerplate]}
1577
1578 The following \Express{} is the expanded form of the short
1579 form schema given in~\ref{;saesl}. In the event of any discrepancy
1580 between the short form and this expanded listing, the expanded
1581 listing shall be used.
1582 </bpfap14>
1583 <*ap>
```

```

\apimpreq Print boilerplate for AP requirements on exchange structure.
  \apimpreq{schema name}.

1583 \newcommand{\apimpreq}[1]{%
1584   The implementation method defines what types of exchange
1585   behaviour are required with respect to this part of ISO~10303.
1586   Conformance to this part of ISO~10303 shall be realized in an
1587   exchange structure. The file format shall be encoded according
1588   to the syntax and \Express{} language mapping defined in
1589   ISO~10303-21 and in the AIM defined in \aref{;saeel} of this part
1590   of ISO~10303. The header of the exchange structure shall identify
1591   use of this part of ISO~10303 by the schema name '#1'.
1592
1593 }

      The end of this package.
1594 \ap)

```

8 The Application Interpreted Construct package

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

```

1595 (*aic)
      If we are in an AIC we are not in an IR.
1596
1597 \anirfalse
1598

```

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
1599 \newcommand{\aicshortexphead}{\clause{EXPRESS short listing}\label{;sesl}}

```

8.2 Boilerplate commands

```

\aicextraintro Print boilerplate for an extra AIC paragraph in the Introduction.

1600 \newcommand{\aicextraintro}{%
1601   This part of ISO~10303 is a member of the application
1602   interpreted construct series.
1603   An application interpreted construct (AIC) provides a
1604   logical grouping of interpreted constructs that supports
1605   a specific functionality for the usage of product data across
1606   multiple application contexts. An interpreted construct is a
1607   common interpretation of the integrated resources that

```

1608 supports shared information requirements among application
1609 protocols.
1610 }

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

```
1611 \newcommand{\aicdef}{%  
1612 \definition{application interpreted construct (AIC)}%  
1613     {a logical grouping of interpreted constructs  
1614     that supports a specific function for  
1615     the usage of product data across multiple  
1616     application contexts.}  
1617 }
```

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

```
1618 \newcommand{\aicshortexpintro}{%  
1619     This clause specifies the \Express{} schema that uses  
1620 elements from the integrated resources and contains the  
1621 types, entity data types specializations, and functions that are  
1622 specific to this part of ISO 10303.  
1623 \begin{anote}There may be subtypes and items of select lists that  
1624 appear in the integrated resources that are not  
1625 imported into the AIC. Constructs are eliminated  
1626 from the subtype tree or select list through the  
1627 use of the implicit interface rules of ISO 10303-11.  
1628 References to eliminated constructs are outside the  
1629 scope of the AIC. In some cases, all items of the select  
1630 list are eliminated. Because AICs are intended to be  
1631 implemented in the context of an application protocol,  
1632 the items of the select list will be defined by the  
1633 scope of the application protocol.  
1634 \end{anote} % end note  
1635 }
```

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

```
1636 \newcommand{\aicexpressg}{%  
1637     The diagrams in this annex are generated from the short  
1638 listing given in \cref{;ses1} and correspond to the \Express{} schemas  
1639 specified in this part of ISO 10303.  
1640 The diagrams use the \ExpressG{} graphical notation for the  
1641 \Express{} language. \expressgdef. \par  
1642 }
```

The end of this package.

```
1643 \end{aic}
```


9 The Abstract Test Suite package

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

```
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.
\theAPTtitle 1648 \gdef\theAPpartno{}
\APTtitle   1649 \newcommand{\APnumber}[1]{\gdef\theAPpartno{#1}}
            1650 \gdef\theAPTtitle{}
            1651 \newcommand{\APTtitle}[1]{\gdef\theAPTtitle{#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  1653 \newcommand{\atssummary}{\underline{\texttt{Test case summary:}}}
\atsinput    1654 \newcommand{\atscovered}{\underline{\texttt{Test purposes covered:}}}
\atsconstraints 1655 \newcommand{\atsinput}{\underline{\texttt{Input specification:}}}
\atsverdict  1656 \newcommand{\atsconstraints}{\underline{\texttt{Constraints on values:}}}
\atsexecution 1657 \newcommand{\atsverdict}{\underline{\texttt{Verdict criteria:}}}
\atsextra    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.
1663 `\newcommand{\aepurposehead}{\sclause{Application element test purposes}}`

`\apobjhead` Command to start an application object clause. Use as `\apobjhead{(Application object n)}`.
1664 `\newcommand{\apobjhead}[1]{\ssclause{#1}}`

`\apasserthead` Command to start an ‘Application assertions’ clause.
1665 `\newcommand{\apasserthead}{\ssclause{Application assertions}}`

`\aimpurposehead` Command to start a ‘AIM test purposes’ clause.
1666 `\newcommand{\aimpurposehead}{\sclause{AIM test purposes}}`

`\aimenthead` Command to start an AIM entity clause. Use as `\aimenthead{(aim entity n)}`.
1667 `\newcommand{\aimenthead}[1]{\ssclause{#1}}`

`\extrefpurposehead` Command to start a ‘External reference test purposes’ clause.
1668 `\newcommand{\extrefpurposehead}{\sclause{External reference test purposes}}`

`\implementpurposehead` Command to start a ‘Implementation method test purposes’ clause.
1669 `\newcommand{\implementpurposehead}{\sclause{Implementation method test purposes}}`

`\otherpurposehead` Command to start an ‘Other test purposes’ clause.
1670 `\newcommand{\otherpurposehead}{\sclause{Other test purposes}}`

`\gtpvthead` Command to start a ‘General test purposes and verdict criteria’ clause.
1671 `\newcommand{\gtpvthead}{\clause{General test purposes and verdict criteria}}`

`\generalpurposehead` Commands to start a ‘General test purposes’ clause.
1672 `\newcommand{\generalpurposehead}{\sclause{General test purposes}}`

`\gvcatchhead` Commands to start a ‘General verdict criteria for all test cases’ clause.
1673 `\newcommand{\gvcatchhead}{\sclause{General verdict criteria for all abstract test cases}}`

`\gvcprehead` Commands to start a ‘General verdict criteria for preprocessor abstract test cases’ clause.
1674 `\newcommand{\gvcprehead}{\sclause{General verdict criteria for preprocessor`
1675 `abstract test cases}}`

`\gvcposthead` Commands to start a ‘General verdict criteria for postprocessor abstract test cases’ clause.
1676 `\newcommand{\gvcposthead}{\sclause{General verdict criteria for postprocessor`
1677 `abstract test cases}}`

`\atthead` Commands to start a ‘Abstract test cases’ clause.
1678 `\newcommand{\atthead}{\clause{Abstract test cases}}`

`\atctitlehead` Command `\atctitlehead{<title>}` to start a particular test case clause.
1679 `\newcommand{\atctitlehead}[1]{\sclause{#1}}`

`\prehead` Commands to start a ‘Preprocessor’ clause.
1680 `\newcommand{\prehead}{\ssclause{Preprocessor}}`

`\posthead` Command `\posthead{<title>}` to start a ‘Postprocessor’ clause.
1681 `\newcommand{\posthead}[1]{\ssclause{Postprocessor}}`

`\confclassannexhead` Commands to start a ‘Conformance classes’ annex.
1682 `\newcommand{\confclassannexhead}{\normannex{Conformance classes}}`

`\confclasshead` Commands to start a ‘Conformance class N’ clause. Us as `\confclasshead{<number>}`.
1683 `\newcommand{\confclasshead}[1]{\sclause{Conformance class #1}}`

`\postipfilehead` Command to start a ‘Postprocessor input specification file names’ annex.
1684 `\newcommand{\postipfilehead}{\normannex{Postprocessor input specification file names}}`

`\atsusagehead` Command to start an ‘ATS Usage scenarios’ annex.
1685 `\newcommand{\atsusagehead}{\infannex{Usage scenarios}}`

9.4 Boilerplate printing

`\atsintroendbp` Print boilerplate for the end of ATS introduction clause.
1686 `\newcommand{\atsintroendbp}{%`
1687 `\input{bpfats1}`
1688 `}`

Here is the text of `bpfats1.tex`.

1689 `</ats>`
1690 `<*bpfats1>`
1691 `\ProvidesFile{bpfats1.tex}[2001/07/16 ATS end intro boilerplate]`
1692 `\typeout{bpfats1.tex [2001/07/16 ATS end intro boilerplate]}`
1693
1694 `The purpose of an abstract test suite is to provide a basis for`
1695 `evaluating whether a particular implementation of an application`
1696 `protocol actually conforms to the requirements of that application`
1697 `protocol. A standard abstract test suite helps ensure that`
1698 `evaluations of conformance are conducted in a consistent manner`
1699 `by different test laboratories.`
1700
1701 `This part of ISO~10303 specifies the abstract test suite for`
1702 `ISO 10303-\theAPpartno, application protocol \theAPtitle.`
1703 `The abstract test cases presented here are the basis for`
1704 `conformance testing of implementations of ISO 10303-\theAPpartno.`
1705
1706 `This abstract test suite is made up of two major parts:`

1707 \begin{itemize}

1708 \item the test purposes, the specific items to be covered by

1709 conformance testing;

1710 \item the set of abstract test cases that meet those test purposes.

1711 \end{itemize}

1712

1713 The test purposes are statements of the application protocol

1714 requirements that are to be addressed by the abstract test cases.

1715 Test purposes are derived primarily from the application protocol's

1716 information requirements and AIM,

1717 as well as from other sources such as standards

1718 referenced by the application protocol and other requirements

1719 stated in the application protocol conformance requirements clause.

1720

1721 The abstract test cases address the test purpose by:

1722 \begin{itemize}

1723 \item specifying the requirements for input data to be used when

1724 testing an implementation of the application protocol;

1725 \item specifying the verdict criteria to be used when evaluating

1726 whether the implementation successfully converted the input

1727 data to a different form.

1728 \end{itemize}

1729

1730 The abstract test cases set the requirements for the

1731 executable test cases that are required to actually conduct

1732 a conformance test. Executable test cases contain the scripts,

1733 detailed values, and other explicit information required to

1734 conduct a conformance test on a specific implementation of

1735 the application protocol.

1736

1737 At the time of publication of this document, conformance

1738 testing requirements had been established for implementations

1739 of application protocols in combination with ISO 10303-21 and

1740 ISO 10303-22. This part of ISO 10303 only specifies

1741 test purposes and abstract test cases for a subset of such

1742 implementations.

1743

1744 For ISO 10303-21, two kinds of implementations, preprocessors and

1745 postprocessors, must be tested. Both of these are addressed in this

1746 abstract test suite.

1747

1748 For ISO 10303-22, a class of applications will possess the capability

1749 to upload and download AP-compliant SDAI-models or schema instances

1750 to and from applications that implement the SDAI. By providing test case

1751 data that correspond with SDAI-models, this abstract test suite addresses

1752 such applications in a single-schema scenario.

1753

1754 </bpfats1>

1755 <*ats>

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

```
1756 \newcommand{\atsscopebp}{%
1757   \input{bpfats2}
1758 }

      Here is the text of bpfats2.tex.
1759 </ats>
1760 <*bpfats2>
1761 \ProvidesFile{bpfats2.tex}[1997/09/30 ATS scope boilerplate]
1762 \typeout{bpfats2.tex [1997/09/30 ATS scope boilerplate]}
1763
1764     This part of ISO 10303 specifies the abstract test suite to be
1765 used in the conformance testing of implementations of
1766 ISO 10303-\theAPpartno.
1767 The following are within the scope of this part of ISO 10303:
1768 \begin{itemize}
1769 \item the specification of the test purposes associated with
1770     ISO 10303-\theAPpartno;
1771 \item the verdict criteria to be applied during conformance
1772     testing of an implementation of ISO 10303-\theAPpartno\
1773     using ISO 10303-21 or ISO 10303-22;
1774 \begin{anote}
1775     The verdict criteria are used to ascertain whether a test purpose
1776     has been satisfactorily met by an implementation under test (IUT)
1777     within the context of a given test case.
1778 \end{anote}
1779 \item the abstract test cases to be used as the basis for the
1780     executable test cases for conformance testing.
1781 \end{itemize}
1782
1783 The following are outside the scope of this part of ISO 10303:
1784 \begin{itemize}
1785 \item the creation of executable test cases;
1786 \item test specifications for tests other than conformance testing
1787     such as interoperability or acceptance testing;
1788 \item other implementation methods.
1789 \end{itemize}
1790
1791 </bpfats2>
1792 <*ats>
```

`\atspurposebp` The boilerplate for the introduction to the Test purposes clause.

```
1793 \newcommand{\atspurposebp}{%
1794
1795     This clause specifies the test purposes for this part of ISO 10303.
1796 Clauses 4.1 and 4.2 are describe the source and meaning of test
1797 purposes that are derived from the information
1798 requirements defined in ISO 10303-\theAPpartno, clause 4, and the
1799 AIM \Express{} schema defined in ISO 10303-\theAPpartno, annex A.
1800 These test purposes are not repeated in this part of ISO~10303.
```

1801 However, through reference in a test case each specific element
 1802 from the application elements of the AIM implicitly requires
 1803 that the identified element, as specified in the test purpose statement,
 1804 will be correctly instantiated by the implementation under test. \par
 1805 }
 1806

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

```
1807 \newcommand{\aetpbp}{%
1808   \input{bpfats3}
1809 }
```

And here is the text of file bpfats3.tex.

```
1810 \</ats>
1811 \<*bpfats3>
1812 \ProvidesFile{bpfats3.tex}[2002/01/23 ATS AE test purpose intro boilerplate]
1813 \typeout{bpfats3.tex [2002/01/23 ATS AE test purpose intro boilerplate]}
1814
1815   Application element (AE) test purposes are implicitly derived
1816 from the AP information requirements and are not explicitly documented
1817 here. AE test purposes apply to the input specifications of both
1818 preprocessr and postprocessor test cases. AE test purposes are implicitly
1819 derived from the AP information requirements as follows:
1820 \begin{itemize}
1821 \item Application objects (see ISO 10303-\theAPpartno, 4.2):
1822   a test purpose derived from an application object is a simple
1823   statement of the object's name;
1824
1825 \item Application object attributes (see ISO 10303-\theAPpartno, 4.2):
1826   test purposes derived from application object attributes are
1827   statements of the application object name with a specific attribute name;
1828
1829 \item Application assertions (see ISO 10303-\theAPpartno, 4.3):
1830   test purposes derived from application assertions are
1831   statements describing the relationships between two application objects.
1832   Application assertion test purposes address the directions of
1833   relationships as well as the number (cardinality) of relationships.
1834
1835 \end{itemize}
1836
1837
1838 They shall be interpreted as given in the
1839 following statement:
1840 %\begin{quotation}
1841   the IUT shall preserve the semantic associated with the unique
1842 application element from which the test purpose was implicitly derived.
1843 %\end{quotation}
1844 This implies that the semantics of the application element are
1845 preserved by the IUT between the input and output of a test,
```

```

1846 according to the reference path specified by the mapping
1847 \mactableorspec{}
1848 defined in ISO 10303-\theAPpartno, 5.1.
1849 \par
1850
1851
1852 </bpfats3>
1853 <*ats>

```

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

```

1854 \newcommand{\aimtpbp}{%
1855   \input{bpfats4}
1856 }

```

And here is the text of file `bpfats4.tex`.

```

1857 </ats>
1858 <*bpfats4>
1859 \ProvidesFile{bpfats4.tex}[2002/01/23 ATS AIM test purpose intro boilerplate]
1860 \typeout{bpfats4.tex [2002/01/23 ATS AIM test purpose intro boilerplate]}
1861
1862     Test purposes are implicitly derived from the AP AIM \Express,
1863 and are not explicitly documented here. AIM test purposes are implicitly
1864 derived from the expanded \Express{} listing contained in
1865 annex~A of ISO 10303-\theAPpartno{} as follows:
1866 \begin{itemize}
1867 \item AIM entity data types: a test purpose derived from an AIM
1868     entity data type is a simple statement of the entity data type name;
1869
1870 \item AIM entity attributes: a test purpose derived from an AIM
1871     entity attribute is a statement of the AIM entity data type with
1872     a given attribute.
1873 \end{itemize}
1874
1875     Aim test purposes shall be interpreted as given in the
1876 following statement:
1877 %\begin{quotation}
1878 the postprocessor shall accept the input in accordance with the
1879 AIM \Express{} structure corresponding to this test purpose.
1880 %\end{quotation}
1881 This implies that the semantics of the application element
1882 represented by the AIM element are preserved by the IUT between
1883 the input and output of a test according to the reference path
1884 specified in the mapping
1885 \mactableorspec{}
1886 of the AP. This also implies
1887 no violations of any constraints (local rules or global
1888 rules) that apply to the AIM element. AIM test purposes apply
1889 to the input specifications of postprocessor test cases only.
1890 \par

```

```
1891
1892 </bpfats4>
1893 <*ats>
```

`\atsimtpbp` `\atsimtpbp` — 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}{%
```



```
1932 \input{bpfats5}
1933 }
```

Here is the text of file bpfats5.tex.

```
1934 </ats>
1935 <*bpfats5>
1936 \ProvidesFile{bpfats5.tex}[1997/09/30 ATS general verdict boilerplate]
1937 \typeout{bpfats5.tex [1997/09/30 ATS general verdict boilerplate]}
1938
1939     General test purposes are statements of requirements that apply
1940 to all abstract test cases, all preprocessor abstract test cases,
1941 or all postprocessor abstract test cases. General verdict criteria
1942 are the means for evaluating whether the general test purposes are
1943 met. General verdict criteria shall be evaluated as a part of every
1944 executable test case to which they apply. Each general verdict criterion
1945 includes a reference to its associated test purpose.
1946
1947 </bpfats5>
1948 <*ats>
```

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

```
1949 \newcommand{\gtppb}{%
1950 \input{bpfats6} }
```

And here is the text of file bpfats6.tex

```
1951 </ats>
1952 <*bpfats6>
1953 \ProvidesFile{bpfats6.tex}[2001/07/16 ATS general test purpose boilerplate]
1954 \typeout{bpfats6.tex [2001/07/16 ATS general test purpose boilerplate]}
1955
1956     The following are the general test purposes for this part of
1957 ISO 10303:
1958
1959 g1 The output of an IUT shall preserve all the semantics defined by
1960 the input model according to the reference paths specified in the
1961 mapping \mappableorspec{} defined in clause~5 of ISO 10303-\theApartno.
1962
1963 g2 The output of a preprocessor shall conform to the implementation
1964 method to which the IUT claims conformance.
1965
1966 g3 The instances in the output of a preprocessor shall be encoded
1967 according to the mapping \mappableorspec{} and the AIM \Express{} long form
1968 defined in 5.1 and annex~A of ISO 10303-\theApartno.
1969
1970 g4 A postprocessor shall accept input data which is encoded according
1971 to the implementation method to which the IUT claims conformance.
1972
1973 g5 A postprocessor shall accept input data structured
1974 according to the mapping \mappableorspec{}
```

```

1975 and the AIM \Express{} long form
1976 defined in 5.1 and annex^A of ISO 10303-\theAPpartno.
1977
1978 \par
1979
1980 </bpfats6>
1981 <*ats>

```

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

```

1982 \newcommand{\gvatcbp}{%
1983 \input{bpfats7} }

And here is the text of file bpfats7.tex
1984 </ats>
1985 <*bpfats7>
1986 \ProvidesFile{bpfats7.tex}[2001/07/16 ATS general verdict criteria boilerplate]
1987 \typeout{bpfats7.tex [2001/07/16 ATS general verdict criteria boilerplate]}
1988
1989 The following verdict criteria apply to all abstract test cases
1990 contained in this part of ISO 10303:
1991
1992 gvc1 The semantics of the input model are preserved in the output of
1993 the IUT according to the reference paths specified in the mapping
1994 \mactableorspec{} defined in ISO 10303-\theAPpartno, clause 5 (g1).
1995
1996 \par
1997
1998 </bpfats7>
1999 <*ats>

```

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

```

2000 \newcommand{\gvcprebp}{%
2001 \input{bpfats8} }

And here is the text of file bpfats8.tex
2002 </ats>
2003 <*bpfats8>
2004 \ProvidesFile{bpfats8.tex}[2001/07/16 ATS general verdict pre boilerplate]
2005 \typeout{bpfats8.tex [2001/07/16 ATS general verdict pre boilerplate]}
2006
2007 The following verdict criteria apply to all preprocessor
2008 abstract test cases contained in this part of ISO 10303:
2009
2010 gvc2 The output of a preprocessor conforms
2011 to the implementation method to which the IUT claims conformance (g2).
2012
2013 gvc3 The instances in the output of a preprocessor are encoded according

```

2014 to the AIM \Express{} long form and mapping \mactableorspec{}
 2015 defined in ISO 10303-\theAPpartno, annex A and 5.1 (g3).
 2016 \par
 2017
 2018 </bpfats8>
 2019 <*ats>

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

2020 \newcommand{\gvcpostbp}{%
 2021 \input{bpfats9} }

And here is the text of file bpfats9.tex

2022 </ats>
 2023 <*bpfats9>
 2024 \ProvidesFile{bpfats9.tex}[2001/07/16 ATS general verdict post boilerplate]
 2025 \typeout{bpfats9.tex [2001/07/16 ATS general verdict post boilerplate]}
 2026
 2027 The following verdict criteria apply to all postprocessor
 2028 abstract test cases contained in this part of ISO 10303:
 2029
 2030 gvc4 The postprocessor accepts input data which is encoded according
 2031 to the implementation method to which the IUT claims conformance (g4).
 2032
 2033 gvc5 The postprocessor accepts input data which is structured according
 2034 to the AIM \Express{} long form and mapping \mactableorspec{}
 2035 defined in ISO 10303-\theAPpartno, annex A and 5.1 (g5).
 2036 \par
 2037
 2038
 2039 </bpfats9>
 2040 <*ats>

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

2041 \newcommand{\atcbp}{%
 2042 This clause specifies the abstract test cases for this part of
 2043 ISO 10303. Each abstract test case addresses one or more test purposes
 2044 explicitly or implicitly specified in clause~4.
 2045 \par
 2046 }
 2047

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

2048 \newcommand{\atcbpii}{%
 2049 \input{bpfats10}
 2050 }

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

```
2051 </ats>
2052 <*bpfats10>
2053 \ProvidesFile{bpfats10.tex}[2001/07/16 ATS ats clause boilerplate]
2054 \typeout{bpfats10.tex [2001/07/16 ATS ats clause boilerplate]}
2055
2056     Each abstract test case has a subclause for the preprocessor
2057 test information and a subclause for each postprocessor
2058 input specification and related test information.
2059 The preprocessor and postprocessor input specifications
2060 are mirror images of each other: they represent the same
2061 semantic information. The preprocessor input model is presented
2062 in the form of a table with five columns:
2063 \begin{itemize}
2064 \item The Id column contains an identifier for the application object
2065     instantiated in a particular row. The identifier may be
2066     referenced as the value of an application assertion.
2067     The identifier is the lowest-level subclause number from
2068     ISO 10303-\theAPpartno, 4.2 where the application
2069     element that appears in that row of the table is specified.
2070
2071 \item The V column specifies whether or not the element in that
2072     row of the table is assigned a verdict in this test case.
2073     A blank indicates that it is not assigned a verdict in this test case.
2074     A ‘*’ indicates that it is assigned a verdict
2075     using a derived verdict criteria. The derived verdict criteria
2076     determine whether the semantics associated with the application
2077     element are preserved in the output of the IUT according to
2078     the reference paths specified in the mapping table defined
2079     in ISO 10303-\theAPpartno, 5.1. A number in the V column references
2080     a specific verdict criterion defined in the verdict criteria
2081     section that follows the preprocessor input specification table.
2082
2083 \item The Application Elements column identifies the particular
2084     application element instance that is being
2085     defined by the table. For assertions the role is specified
2086     in parenthesis.
2087
2088 \item The Value column specifies a specific value for the application
2089     element. For application objects and attributes the value column
2090     defines the semantic value for that element’s instance in the
2091     input model. A ‘\#<number>’ in the column is a reference
2092     to an entity instance name in the postprocessor input specification
2093     where the corresponding value is specified. For assertions, this
2094     column holds a link to the related application object.
2095     A ‘<not\_present>’ indicates that the
2096     application element is not present in the
2097     input model.
2098
```

2099 \item The Req column specifies whether the value in the Value column
2100 is mandatory (M), suggested (S) or constrained (C\$<n\$>), where 'n'
2101 is an integer referencing a note that follows the table.
2102 A suggested value may be changed by the test realizer.
2103 A mandatory value may not be changed due to rules in \Express,
2104 rules in the mapping \mactableorspec, or the requirements of the test
2105 purpose being assigned a verdict. Each constrained value references
2106 a note labelled C\$<number\$> at the end of the preprocessor
2107 input model table and may be modified according to specific
2108 constraints specified in it.
2109 \end{itemize}
2110
2111 The postprocessor input specifications are defined using
2112 ISO 10303-\theAPpartno. The values in the postprocessor specifications
2113 are suggested unless declared mandatory or constrained by the
2114 preprocessor input table.
2115
2116 The abstract test case specifies all the verdict criteria that are
2117 used to assign a verdict during testing. Special verdict criteria for
2118 preprocessor and postprocessor testing are defined explicitly in each
2119 abstract test case subclause. The relevant derived verdict criteria
2120 for preprocessor and postprocessor testing are identified in the V
2121 column of the preprocessor input table.
2122
2123 </bpfats10>
2124 <*ats>

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

2125 \newcommand{\atcpretpc}{%
2126 \input{bpfats11}
2127 }
2128

Here is the text of bpfats11.tex.

2129 </ats>
2130 <*bpfats11>
2131 \ProvidesFile{bpfats11.tex}[2001/07/16 ATS preprocessor purposes covered boilerplate]
2132 \typeout{bpfats11.tex [2001/07/16 ATS preprocessor purposes covered boilerplate]}
2133
2134 In the preprocessor input specification table of this test case, the
2135 numbers in the Id column (ignoring the part beyond the decimal point, if any)
2136 whose rows are not empty in the V column identify the application objects
2137 that are covered by this test case. These Id numbers refer directly to
2138 the subclause numbers within ISO 10303-\theAPpartno, 4.2, where the
2139 application object is defined.
2140 \par
2141
2142 </bpfats11>
2143 <*ats>

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

```
2144 \newcommand{\atcposttpc}{%
2145   \input{bpfats12}
2146 }
2147
```

Here is the text of `bpfats12.tex`.

```
2148 </ats>
2149 <*bpfats12>
2150 \ProvidesFile{bpfats12.tex}[2001/07/16 ATS postprocessor purposes covered boilerplate]
2151 \typeout{bpfats12.tex [2001/07/16 ATS postprocessor purposes covered boilerplate]}
2152
2153     In the postprocessor input specification table of this test case, the
2154 numbers in the Id column (ignoring the part beyond the decimal point, if any)
2155 whose rows are not empty in the V column identify the application objects
2156 that are covered by this test case. These Id numbers refer directly to
2157 the subclause numbers within ISO 10303-\theAPpartno, 4.2, where the
2158 application object is defined.
2159 \par
2160
2161 </bpfats12>
2162 <*ats>
```

`\confclassbp` `\confclassbp{<number>}` prints the boilerplate for the start of a *Conformance class* clause.

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

```
2163 \newcommand{\confclassbp}[1]{%
2164
2165     To conform to conformance class #1 of ISO 10303-\theAPpartno,
2166 an implementation shall pass executable versions of the following
2167 abstract test cases: }
2168 \newcommand{\atsnoclassesbp}{%
2169     Conformance to ISO 10303-\theAPpartno\ is defined only in terms
2170 of the entire AP. Therefore, conformance requires that an
2171 implementation pass executable versions of all abstract
2172 test cases in clause 6. }
```

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

```
2173 \newcommand{\pisfbp}[3]{\par
2174     This annex references a listing of the postprocessor input
2175 specifications for this part of ISO 10303 without comments or other
2176 explanatory text. These specifications are documented using
2177 ISO 10303-#1. These specifications are available in
2178 computer-interpretable form and can be found at the following URL:
2179 \begin{center}
2180 \isourl{#2}
```

2181 \end{center}

2182

2183 If there is difficulty accessing this site contact the ISO Central
2184 Secretariat or contact the ISO TC184/SC4 Secretariat directly at:
2185 \url{sc4sec@cme.nist.gov}.

2186

2187 The postprocessor input specifications for each test case is supplied
2188 electronically via the Internet. Table~#3 lists
2189 the file name of the postprocessor input specification that is
2190 associated with the postprocessor subclause(s) of a test case.

2191 \par

2192 %% \input{bpfats13}

2193 }

2194

 Here is the text of bpfats13.tex.

2195 </ats>

2196 <*bpfats13>

2197 \ProvidesFile{bpfats13.tex}[2001/07/16 ATS postprocessor annex (B) boilerplate]

2198 \typeout{bpfats13.tex [2001/07/16 ATS postprocessor annex (B) boilerplate]}

2199

2200 This annex references a listing of the postprocessor input
2201 specifications for this part of ISO~10303 without comments or other
2202 explanatory text. These specifications are documented using
2203 ISO 10303-\atstempa. These specifications are available in
2204 computer-interpretable form and can be found at the following URL:

2205 \begin{center}

2206 \isourl{\atstempb}

2207 \end{center}

2208

2209 If there is difficulty accessing this site contact the ISO Central
2210 Secretariat or contact the ISO TC184/SC4 Secretariat directly at:
2211 \url{sc4sec@cme.nist.gov}.

2212

2213 The postprocessor input specifications for each test case is supplied
2214 electronically via the Internet. Table~\atstempc{} lists
2215 the file name of the postprocessor input specification that is
2216 associated with the postprocessor subclause(s) of a test case.

2217

2218

2219 </bpfats13>

2220 <*ats>

 The end of this package.

2221 </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 *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	<code>\@fax</code> <u>558</u> , 671, 691	<code>\l</code> 1108
<code>\#</code> 82, 2091	<code>\@keywords</code> <u>550</u> , 670, 686	
<code>\&</code> 1285	<code>\@minus</code> 140	
<code>\-</code> 76–79	<code>\@olddocnumber</code>	<code>\lrcorner</code> 208, 211, 213, 653, 655, 1772, 2169
<code>\@abstract</code> <u>546</u> , 670, 685 <u>544</u> , 669, 684	
<code>\@address</code> . <u>558</u> , 671, 689	<code>\@oldwg</code> . . . <u>542</u> , 669, 683	
<code>\@altaddress</code>	<code>\@owner</code> . . . <u>558</u> , 671, 688	A
. <u>571</u> , 672, 694	<code>\@plus</code> 140	<code>\aamdefhead</code> <u>851</u>
<code>\@altemail</code> <u>571</u> , 672, 697	<code>\@startsection</code> 113	<code>\aamfighead</code> <u>854</u>
<code>\@altfax</code> . . <u>571</u> , 672, 696	<code>\@telephone</code> <u>558</u> , 671, 690	<code>\aamfigrange</code> <u>1464</u>
<code>\@altowner</code> <u>571</u> , 672, 693	<code>\@wg</code> <u>536</u> , 669, 680	<code>\aamfigrs</code>
<code>\@alttelephone</code>	<code>\</code> 92, 222, 385, 732–735, 773–778	. . . <u>1464</u> , 1478, 1505
. <u>571</u> , 672, 695	<code>\{</code> 368, 378, 1192	<code>\aamfigures</code> <u>1499</u>
<code>\@comread</code> . <u>553</u> , 670, 687	<code>\}</code> 368, 378, 1192	<code>\aamhead</code> <u>849</u>
<code>\@copyrighttext</code> . . . 590	<code>_</code> 175, 1267, 1323, 1327, 1346, 1350, 2095	<code>\abstract</code> <u>546</u> , 685
<code>\@docdate</code> . <u>540</u> , 669, 682		<code>\address</code> <u>558</u> , 689
<code>\@docnumber</code> <u>538</u> , 669, 681		<code>\aepurposehead</code> . . . <u>1663</u>
<code>\@email</code> . . . <u>558</u> , 671, 692		<code>\aetpbb</code> <u>1807</u>

<code>\cleardoublepage</code>	.. 186	350, 645, 654,	<code>specific@tion</code>	.. 102		
<code>\clearpage</code>	276, 587, 675	712, 715, 1014,	<code>sspec</code> 103		
<code>\CodelineIndex</code> 4	1371, 1518, 1522	<code>tspec</code> 103		
<code>\comread</code> 553 , 687	<code>\em</code> 67, 68, 73	<code>espec (environment)</code>	.. 103	
<code>\confclassannexhead</code> 1682	<code>\email</code> 558 , 692	<code>\exampleshead</code> 345	
<code>\confclassbp</code> 2163	<code>\EnableCrossrefs</code>	... 3	<code>excode (environment)</code>	131	
<code>\confclasshead</code>	... 1683	<code>\end</code>	15, 124, 129, 134,	<code>expdesc (environment)</code>	136	
<code>\confreqhead</code> 841		148, 153, 158,	<code>\expdesclabel</code>	.. 136 , 142	
<code>\cref</code> 887, 1022,		163, 168, 232,	<code>\Express</code>	... 76 , 122 ,	
	1125, 1517, 1638		281, 301, 369,		311, 386, 394,	
			379, 416, 644,		396, 799, 802,	
			659, 673, 737,		898, 901, 1296,	
			744, 780, 786,		1363, 1373,	
			924, 948, 951,		1526, 1529,	
			957, 966, 1113,		1536, 1545,	
			1222, 1224,		1547, 1576,	
			1251, 1269,		1588, 1619,	
			1294, 1319,		1638, 1641,	
			1342, 1424,		1799, 1862,	
			1537, 1566,		1864, 1879,	
			1634, 1711,		1967, 1975,	
			1728, 1778,		2014, 2034, 2103	
			1781, 1789,		<code>\ExpressG</code>	76 , 347 , 801 ,
			1835, 1843,			1518, 1528, 1640
			1873, 1880,		<code>\expressgdef</code>	347 , 802 ,
			2109, 2181, 2207			1522, 1529, 1641
			<code>\endForeword</code> 220	<code>\expressghead</code> 341
			<code>\endlist</code> 143	<code>\expresshead</code> 339
			<code>\entityhead</code> 319	<code>\ExpressI</code> 76 , 127
			<code>enumlist</code>	(environ-	<code>\ExpressX</code> 76 , 132
				ment)	<code>\expurls</code> 384
				160	<code>\extrefpurposehead</code>	1668
				environments:		
				Introduction		
				... 275		
				<code>apconformclasses</code>		
			 1419		
				<code>apuof</code> 961	
				<code>arglist</code> 165	
				<code>attrlist</code> 145	
				<code>dtext</code> 108	
				<code>ecode</code> 121	
				<code>eicode</code> 126	
				<code>enumlist</code> 160	
				<code>espec</code> 103	
				<code>excode</code> 131	
				<code>expdesc</code> 136	
				<code>fproplist</code> 150	
				<code>fspec</code> 103	
				<code>iproplist</code> 155	
				<code>majorsublist</code>	... 299	
				<code>rspec</code> 103	
						1371, 1518, 1522
D						
<code>\DeclareRobustCommand</code> 349					
<code>\def</code>	.. 537, 539, 541,					
	543, 545, 548,					
	549, 551, 552,					
	555, 556, 559,					
	561, 563, 565,					
	566, 568, 569,					
	572, 574, 576,					
	578, 579, 581, 582					
<code>\definition</code> 1612					
<code>\disref</code> 443, 453					
<code>\docdate</code> 540 , 682					
<code>\docidhead</code> 336					
<code>\DocInput</code> 12					
<code>\docnumber</code>	... 538 , 681					
<code>\docreg</code> 364					
<code>\doctitle</code> 49					
<code>\documentclass</code> 2					
<code>\domainpurposehead</code>	1662					
<code>\dotfill</code> 62–64					
<code>\draftctr</code> 790					
<code>\drawcoversheet</code>	586, 594					
<code>dtext (environment)</code>	.. 108					
E						
<code>\E</code> 72					
<code>ecode (environment)</code>	.. 121					
<code>\ehe@d</code> 113 ,					
	122, 127, 132,					
	146, 151, 156,					
	161, 166, 867–869					
<code>\ehe@dmarg</code> 113					
<code>eicode (environment)</code>	126					
<code>\else</code> 189, 196,					
	205, 209, 212,					
	227, 230, 323,					

<code>\nrefpartxlili</code>	420	1017,	1120,	<code>\schemahead</code>	309
<code>\nrefpartxliia</code>	483	1231,	1307,	<code>\schemaidhead</code>	337
<code>\nrefpartxxi</code>	420	1360,	1394,	<code>\schemaintro</code>	309
<code>\nrefpartxxii</code>	420	1406,	1432,	<code>\schemareg</code>	374
<code>\nrefpartxxxi</code>	420	1445,	1469,	<code>\schemidhead</code>	337
<code>\nrefpartxxxii</code>	420	1487,	1502,	<code>\scivm@in</code>	181, 192
		1542,	1557,	<code>\sclause</code>	
O		1573,	1691,	. 306–308, 314–	
<code>\objreghead</code>	335	1761,	1812,	317, 319, 322,	
<code>\olddocnumber</code>	544 , 684	1859,	1898,	324, 328, 329,	
<code>\oldwg</code>	542 , 683	1936,	1953,	331, 332, 336,	
<code>\otherdefhead</code>	308	1986,	2004,	337, 823, 827,	
<code>\otherindexspace</code>	65	2024,	2053,	829, 833, 840,	
<code>\otherpurposehead</code> 1670		2131,	2150,	853, 856, 1662,	
<code>\owner</code>	558 , 688	<code>\ProvidesPackage</code> 19,		1663, 1666,	
		22, 25, 28, 31, 34		1668–1670,	
P		<code>\pstyle</code>	10	1672–1674,	
<code>\pagebreak</code>	110	<code>\purposeshead</code>	1661	1676, 1679, 1683	
<code>\pagenumbering</code>	186	<code>\purposesname</code>	1661	<code>\series</code>	45
<code>\par</code> . 67, 68, 184, 221,		<code>\put</code>	537,	<code>\setcounter</code>	
224, 312, 1641,		539, 541, 543,		. . . 6, 53, 187, 188	
1804, 1849,		545, 548, 549,		<code>\setlength</code>	
1890, 1927,		551, 552, 555,		. 138–141, 596, 674	
1978, 1996,		556, 559, 561,		<code>\settocdepth</code> 808, 820,	
2016, 2036,		563, 565, 566,		826, 852, 855, 858	
2045, 2140,		568, 569, 572,		<code>\shortnamehead</code>	334
2159, 2173, 2191		574, 576, 578,		<code>\shortnames</code>	352
<code>\parameters</code>		579, 581, 582,		<code>\signature</code>	
.	868 , 1325 , 1348	599–623, 646,		867 , 1321 , 1344
<code>\parbox</code>	548, 549,	651, 657, 660–668		<code>\indexfill</code>	62
555, 556, 561, 574				<code>\singleentityhead</code>	319
<code>\parsep</code>	141	Q		<code>\singlefunctionhead</code> 331	
<code>\partidefhead</code>	306	<code>\QUES</code>	89	<code>\singerulehead</code>	328
<code>\partno</code>	43	R		<code>\singletypehead</code>	316
<code>\pbre@k</code>	110	<code>\raggedright</code>	99	<code>\small</code>	9, 82–94
<code>\picsannex</code>	1442	<code>\RecordChanges</code>	5	<code>\space</code> 648, 963, 989, 1004	
<code>\picshead</code>	342	<code>\ref</code> 881, 883, 894, 897,		<code>specific@tion</code> (envi-	
<code>\pisfbp</code>	2173	953, 978, 1029,		ronment)	102
<code>\posthead</code>	1681	1218, 1254, 1577		<code>\sptitle</code>	184, 193
<code>\postipfilehead</code>	1684	<code>\refdefhead</code>	307	<code>\ssclause</code>	318,
<code>\prehead</code>	1680	<code>\reference</code> 489, 496, 502		327, 330, 333,	
<code>\PrintChanges</code>	14	<code>\relax</code>	680–697	338, 824, 835,	
<code>\PrintIndex</code>	13	<code>\renewcommand</code>	62–68	836, 1664, 1665,	
<code>\protect</code>	122,	<code>\RequirePackage</code>	38	1667, 1680, 1681	
127, 132, 190, 595		<code>rspec</code> (environment)	103	<code>\ssindexfill</code>	62
<code>\ProvidesFile</code>	248,	<code>\rulehead</code>	328	<code>sspec</code> (environment)	103
285, 355, 391,				<code>\sssclause</code>	
406, 702, 750,		S		837, 1310, 1333
874, 918, 936,		<code>\sc</code>	76–79	<code>\sssssclause</code>	838
973, 985, 1000,				<code>\sstemplates</code>	1304

<code>\st@pn@me</code>	49 , 658	1825,	1829,	1543,	1558,
<code>\stepc@mp</code>	182 , 192	1848,	1865,	1574,	1692,
<code>\STEPcover</code>	584	1961,	1968,	1762,	1813,
<code>\stepemptystring</code> . .		1976,	1994,	1860,	1899,
. 41 , 227 , 230 , 321		2015,	2035,	1937,	1954,
<code>\stepparttitle</code>	180	2068,	2079,	1987,	2005,
<code>\steptrid</code>	235	2112,	2138,	2025,	2054,
		2157,	2165,	2132,	2151,
		2169		2198	
T					
<code>\techdishead</code>	344	<code>\theAPtitle</code> .	1648 , 1702	U	
<code>\telephone</code>	558 , 690	<code>\theb@cyc</code>	655	<code>\undef@covercmds</code> . .	
<code>\templateshead</code>	834	<code>\thed@ctitle</code>	49 , 658	676 , 679
<code>\textbackslash</code>	1211	<code>\Theseries</code>	45 , 658	<code>\underline</code>	122 , 127 ,
<code>\textbf</code>	81 ,	<code>\theseries</code>	45 , 267	132 , 146 , 151 ,	
537 , 539 , 543 ,		<code>\thespartno</code>		156 , 161 , 166 ,	
545 , 648 , 706 , 754		43 , 193 , 208 ,	867 – 869 , 1653 – 1659	
<code>\textit</code>	217 , 218 , 239 , 240	211 , 213 , 237 ,	<code>\unitlength</code>	596 , 674
<code>\textsc</code>	96	368 , 378 , 653 , 655	<code>\uofhead</code>	822
<code>\textsf</code>	7	<code>\thest@tus</code>	507 , 653 , 655	<code>\url</code>	411 , 1561 , 2185 , 2211
<code>\textsl</code>	9 , 10	<code>\thesteparttitle</code> .		V	
<code>\texttt</code>	8 , 82 – 94 ,	180 , 192 , 195 , 197	<code>\value</code>	652
736 , 1184 , 1188 ,		<code>\thicklines</code>	598	<code>\vbox</code>	195 , 197
1192 , 1194 ,		<code>\this@rtno</code>	183 , 193	<code>\verb</code>	1081 , 1085 , 1089 ,
1196 , 1197 ,		<code>\thispagestyle</code>		1091 , 1093 ,	
1199 , 1201 ,		190 , 588 , 595	1094 , 1096 ,	
1203 – 1205 ,		<code>\tiny</code>	99 , 599	1098 , 1100 –	
1207 , 1209 ,		<code>tspec</code> (environment) .	103	1102 , 1104 ,	
1211 , 1213 ,		<code>\twocolumn</code>	195	1106 , 1108 , 1110	
1217 , 1220 ,		<code>\typehead</code>	316	<code>\vspace</code>	195 , 197 ,
1283 , 1285 ,		<code>\typeout</code>	249 ,	647 , 648 , 705 , 753	
1287 , 1289 ,		286 , 356 , 392 ,	407 , 703 , 751 ,	W	
1291 , 1653 – 1659		875 , 919 , 937 ,	974 , 986 , 1001 ,	<code>\wg</code>	536 , 680
<code>\Tfont</code>	192	1018 , 1121 ,	1232 , 1308 ,	X	
<code>\theap</code>	810 , 963 , 989 , 1004	1361 , 1395 ,	1407 , 1433 ,	<code>\xword</code>	96
<code>\theAPpartno</code>	1648 ,	1446 , 1470 ,	1488 , 1503 ,	Z	
1702 , 1704 ,				<code>\z@</code>	114 , 138 – 140
1766 , 1770 ,					
1772 , 1798 ,					
1799 , 1821 ,					