Contents Page

Figures

Tables

Foreword

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

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

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

Introduction

This document specifies the LaTeX class and package files for use in preparing ISO standard documents.

Overview

This document describes a LaTeX class file, called iso, for typesetting ISO standards. It also provides descriptions of some particular package files (e.g., isrot) that have been developed to support the writing of ISO standards. The electronic source of this document also provides an example of the use of these files.

The current set of files have been developed by Peter Wilson (`pwilson@cme.nist.gov`) (RPI, CUA and NIST) from files written by Kent Reed (NIST). In turn, these were revisions of files originally created by Phil Spiby (CADDETC, UK), based on early work by Phil Kennicott(GE).[1)]

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

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

With the issuance of these standards, the opportunity has been taken to to provide a new baseline release of the package files. This new baseline release has also been designed to cater for the fact that a major update of LaTeX to LaTeX2e took place during 1994. LaTeX2e is now the only officially supported version. However, users are still in the process of converting to LaTeX2e, and many have not yet done so. Consequently, although the facilities described in this document are common for both LaTeX v2.09 and LaTeX2e users, there are two different implementations.

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

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

LaTeX for standards: The LaTeX package files user manual — Peter Wilson (CUA and NIST) — May 1996

1 Scope

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

The following are within the scope of this manual:

- use of LaTeX for preparing ISO standard documents.

The following are outside the scope of this manual:

- use of LaTeX in general;

- use of other document preparation systems.

2 Normative references

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

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

ISO TC 184/SC4 N-432, *Supplementary directives for the drafting and presentation of ISO 10303.*

ISO/IEC 8824-1:—[2)], *Information technology — Open systems interconnection — Abstract syntax notation one (ASN.1) — Part 1: Specification of basic notation.*

3 Definitions

For the purposes of this manual, the following definitions apply.

[2)] To be published.

3.1 boilerplate: Text whose wording is fixed.

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

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

4 Symbols and abbreviations

For the purposes of this manual, the following abbreviations apply.

DIS — Draft International Standard

IS — International Standard

ISOD — ISO directives Part 3

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

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

5 Conformance requirements

The macro source files shall not be modified.

If there is a need to modify the macro definitions then the modifications shall be defined in a separate .sty file (or files), using the LaTeX `\renewcommand` and/or the `\renewenvironment` commands as appropriate. The resulting .sty file(s) shall then be called in within the preamble portion of the document to be typeset.

6 Fundamental concepts and assumptions

It is assumed that the reader of this document is familiar with the LaTeX document preparation system.

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

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

If there are any discrepancies between the layout and wording of this document and the require-

Table 1 – Current file versions

Facility	**LaTeX2e**	**LaTeX v2.09**
iso	isov1.cls	isonev1.sty
9pt option	iso9.clo	iso9.sty
10pt option	iso10.clo	iso10.sty
11pt option	iso11.clo	iso11.sty
isorot	isrot1.sty	isrotne1.sty
askinc	askincv1.sty	askincv1.sty
xtab	xtabv1.sty	xtabv1.sty

ments of the ISO Directives Part 3 or the Supplementary Directives, then the requirements in those documents shall be followed for ISO 10303 standard documents.

Because of many revisions over the years to the files described herein, a naming convention has been adopted for them. One part of the convention is that the primary name of the file is suffixed by v# or #, where # is the version number of the file in question. The other part of the convention is that files that are specific to LaTeX v2.09 that are not otherwise distinguishable from files specifically for LaTeX2e, include ne as part of their primary name. All file primary names have been limited to a maximum of eight characters.

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

EXAMPLES

1 – At the time of publication of this document, any references to iso.cls should be read as actually referring to isov1.cls, and similarly for references to other files.

2 – The files isrot1.sty and isrotne1.sty are respectively the LaTeX2e package file and the LaTeX v2.09 style file implementing the isrot facilities.

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

7 The iso facility

The LaTeX iso class/style file is a general file for use in preparing ISO standard documents using the LaTeX document preparation system.

As usual, any LaTeX document has the following structure:

```
\documentclass[<list of options>]{iso}
% preamble goes here
\begin{document}
% document body goes here
\end{document}
```

NOTE – Remember to replace iso above by the primary name of the current version.

7.1 Options

The iso class file supports the following options:

- `draft` For a draft document where overfull hboxes are marked, marginal notes are allowed, and ISO copyright text is not placed in the document;
- `final` The opposite of `draft` (this is the default);
- `letterpaper` For printing on US sized paper;
- `a4paper` For printing on A4 size paper (this is the default);
- `twocolumn` For two column formatting;
- `onecolumn` For single column printing (this is the default);
- One of `11pt`, `10pt`, `9pt` for 11pt, 10pt or 9pt printing respectively (the default is `11pt`);
- `notcopyright` For disabling the printing of copyright notices (this is the default);
- `copyright` Enables printing of copyright notices;
- `dis` For documents that are Draft International Standards;
- `is` For International Standard documents;
- `otherdoc` For documents that are not intended to become a Standard;
- `sect` To enable use of a `\section` command, which is unusual in ISO documents;
- Any other facilities that are available via packages.

When no options are specified, then the result is 11pt, single column printing on A4 size paper, without copyright notice and a running header for an International Standard document.

NOTES

1 – ISO calls for 9pt double column printing but the STEP Supplementary Directives call for 11pt single column printing.

2 – Using 9pt double column printing is awkward if any computer code has to be placed in the document. The IS-REVIEW accepted the STEP 11pt single column layout.

3 – The user of iso is encouraged to process this document using different combinations of the options to gain experience with their effects.

7.2 Sectioning commands

Table 2 gives the sectioning commands defined for iso class documents.

Table 2 – Sectioning commands.

Level	Command
0	`\section`
1	`\clause`, `\normannex`, `\infannex`, `\repannex`
2	`\sclause`
3	`\ssclause`
4	`\sssclause`
5	`\ssssclause`
6	`\sssssclause`

The `\section` command is similar to the normal LaTeX `\part` command. It is only available when the `sect` option is specified.

The `\clause` commands are similar to the normal LaTeX `\section`ing commands. There are also starred versions of these commands (e.g., `\clause*`).

Three varieties of annex commands are available:

– `\normannex{`⟨*title*⟩`}` for a normative annex;

– `\infannex{`⟨*title*⟩`}` for an informative annex;

– `\repannex{`⟨*title*⟩`}` for an annex that is neither normative or informative (e.g. an annex in a technical report).

Only `\sclause` and lower level sectioning commands can be used after an annex command.

EXAMPLES

3 – The command `\infannex{Bibliography}`, assuming that this is the first annex in the document, produces:

Annex A
(informative)

Bibliography

4 – The command `\repannex{Title of annex}`, assuming that this is the second annex in the document, produces:

Annex B

Title of annex

7.3 LaTeX environments and commands

Many of the standard LaTeX environments and commands are available. In particular, all the normal mathematical typesetting capabilities are present. However, there are some additional environments and commands defined.

7.3.1 Lists

The standard LaTeX `itemize`, `enumerate` and `description` environments are provided. The labels in these lists, though, differ from those normally provided by LaTeX.

EXAMPLE 5 – The list environments provided are shown below as:

```
\begin{itemize}
\item First level itemized element;
  \begin{itemize}
  \item Second level itemized element;
    \begin{itemize}
    \item Third level itemized element.
    \end{itemize}
  \end{itemize}
\end{itemize}

\begin{enumerate}
\item First level enumerated element;
  n\begin{enumerate}
  \item Second level enumerated element;
    \begin{enumerate}
    \item Third level enumerated element;
    \end{enumerate}
  \end{enumerate}
\end{enumerate}


\begin{description}
\item[Description] A description element. Note that a colon is
                   automatically added to the item label.
\end{description}
```

and they produce

– First level itemized element;

  • Second level itemized element;

    ∗ Third level itemized element.

a) First level enumerated element;

  1) Second level enumerated element;

    i) Third level enumerated element;

Description: A description element. Note that a colon is automatically added to the item label.

EXAMPLE 6 – Here is a more complex set of lists:

– First level itemization

  a) First level enumeration

    • Second level itemization

      1) Second level enumeration

        * Third level itemization

        * 3rd level itemization

      2) 2nd level enumeration

    • 2nd level itemization

  b) 1st level enumeration

– 1st level itemization

NOTES

1 – On occasion, LaTeX objects to a well formed list. Typically, this happens when a list has only one entry. The LaTeX error message is of the form:

```
! Something's wrong--perhaps a missing \item.
```

Hitting the `<return>` key usually gets LaTeX to run on happily.

2 – Similarly, you may get the message

```
! Too deeply nested
```

Again, hit `<return>` and processing should continue. However, the indentation of later lists may not be correct.

7.3.2 Notes

Three different kind of note environments are provided.

7.3.2.1 Isolated notes

The environment `\begin{note}` ⟨*text*⟩ `\end{note}` produces a numbered note whose body consists of ⟨*text*⟩.

The environment `\begin{anote}` ⟨*text*⟩ `\end{anote}` produces an un-numbered note whose body consists of ⟨*text*⟩.

EXAMPLE 7 – The commands

```
\begin{note}Numbered note.\end{note}
\begin{anote}Plain note.\end{anote}
```

produce

NOTE 1 – Numbered note.

NOTE – Plain note.

7.3.2.2 Grouped notes

The environment

```
\begin{notes}
\begin{note}First note\end{note}
\begin{note}Second note\end{note}
...
\end{notes}
```

produces a list of numbered notes, as shown in the following example.

EXAMPLE 8 – A list of notes.

NOTES

1 – First note

2 – Second note

7.3.3 Examples

Two example environments are defined, both producing numbered examples.

EXAMPLES

9 – The environments `example` and `examples` produces examples in the same style as the `note` and `notes` environments do for notes.

10 – This grouping of examples was produced by the commands:

```
\begin{examples}
  \begin{example}The environments ... \end{example}
  \begin{example}This grouping ... \end{example}
\end{examples}
```

NOTE 1 – Numbered examples and notes may be identified using the `\label{}` construct (as this note 1 is) and then referred to by the `\ref{}` construct.

7.3.4 Bibliographic references

Two different kinds of environments are provided for bibliographic references. It should be noted that neither of these has anything to do with BibTeX.

7.3.4.1 Normative references

The normative references are begun by the command `\begin{nreferences}` and ended by the command `\end{nreferences}`.

Between these two commands, bibliographic entries are denoted by either `\isref{`⟨*ref*⟩`}{`⟨*title*⟩`}` or by `\disref{`⟨*ref*⟩`}{`⟨*title*⟩`}`. The ⟨*ref*⟩ parameter is the number of the standard document and the ⟨*title*⟩ parameter is the title of the standard.

Use the `\isref` command for published standards and the `\disref` command for documents that have not yet been finally approved as a standard. This latter command automatically adds a footnote to the effect that the document is to be published.

EXAMPLE 11 – The LaTeX source for the references in clause 2 of this document is:

```
\begin{nreferences}
\isref{ISO Directives Part 3}{Drafting and presentation of International
           Standards.}
\isref{ISO TC 184/SC4 Editing N-48}{Supplementary directives for the
           drafting and presentation of ISO 10303 Version 2.3.}
\disref{ISO/IEC 8824-1:---}{Information technology ---
       Open systems interconnection ---
       Abstract syntax notation one (ASN.1) ---
       Part 1: Specification of basic notation.}
\end{nreferences}
```

7.3.4.2 Informative references

Informative bibliography elements are started by `\begin{references}` and ended by `\end{references}`. Each element in the list is specified as `\reference{`⟨*author*⟩`}{`⟨*title*⟩`}{`⟨*publisher*⟩`}`.

EXAMPLE 12 – The LaTeX source for annex F in one version of this document was:

```
\infannex{Bibliography}\label{biblio}
\begin{references}
\reference{LAMPORT, L.,}{\LaTeX\/ A Document Preparation System,}%
           {Addison-Wesley Publishing Co., 1986} \label{lamport}
\reference{GOOSENS, M., MITTELBACH, F. and SAMARIN, A.,}{%
           The \LaTeX\/ Companion,}
           Addison-Wesley Publishing Co., 1994} \label{goosens}
\reference{CHEN, P. and HARRISON, M.A.,}{Index preparation and
           processing,}{Software--Practice and Experience, 19(9):897--915,
           September 1988.} \label{chen}
....
\end{references}
```

Informative references may be cited in the text via the LaTeX `\label` and `\ref` mechanism. Note that `\cite` does not work for references to bibliographic items. For the purposes of ISO documents, the command `\bref{`⟨*ref*⟩`}` is supplied which properly formats a bibliographic reference.

7.3.5 Listing of scope items

The `inscope` and `outofscope` environments are provided for itemized listing of elements that are within and outside the scope of the standard. Each list element is introduced via the `\item` command. Some boilerplate text is also printed that introduces the scope list.

The environments take one parameter, ⟨*text*⟩, that must read naturally in a sentence of the form: 'The following are within/outside the scope of this ⟨*text*⟩:'. The non-parameterized part of this sentence is specified by the commands —`\inscopename` and `\outofscopename` respectively for 'within' and 'outside'.

EXAMPLE 13 – The following text was printed by the commands shown at the end of the example.

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

- use of LaTeX for preparing ISO standard documents;
- use of LaTeX for preparing ISO 10303 documents.

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

- use of LaTeX in general;
- use of other document preparation systems.

```
\begin{inscope}{part of ISO~10303}
\item use of \LaTeX\ for preparing ISO standard documents;
\item use of \LaTeX\ for preparing ISO~10303 documents.
\end{inscope}

\begin{outofscope}{part of ISO~10303}
\item use of \LaTeX\ in general;
\item use of other document preparation systems.
\end{outofscope}
```

7.3.6 Listing of definitions

The `\begin{olddefinitions}` ...`\end{olddefinitions}` environment is provided for listing terms that have been defined within the normatively referenced documents. Each term in the list is specified as
`\olddefinition{`⟨*phrase*⟩`}{`⟨*optional text*⟩`}`.

The `\begin{definitions}` ...`\end{definitions}` environment is provided for listing the definitions of terms specific to the document being written. Each term in the list is specified as
`\definition{`⟨*phrase*⟩`}{`⟨*definition text*⟩`}`.

EXAMPLES

14 – A listing of terms defined elsewhere could be specified as:

```
\begin{olddefinitions}
\olddefinition{application protocol (AP)}{}
\olddefinition{integrated resource}{}
\end{olddefinitions}
```

15 – The definition listing earlier in this document was produced by:

```
\begin{definitions}
\definition{boilerplate}{Text whose wording is fixed.}
\definition{style file}{A set of \LaTeX\ macros assembled
                        into a single file.}
...
\end{definitions}
```

7.3.7 Listing of abbreviations

The `\begin{symbols}` ...`\end{symbols}` environment is provided for listing symbols and abbreviations. Each term in the list is specified as
`\symbol{`⟨*symbol*⟩`}{`⟨*definition text*⟩`}`.

EXAMPLE 16 – The list of abbreviations earlier in this document was specified as:

```
\begin{symbols}
\symbol{DIS}{Draft International Standard}
\symbol{IS}{International Standard}
...
\end{symbols}
```

7.4 Floating bodies

LaTeX provides the `figure` and `table` environments. Captions (produced by the `\caption` command) increment the figure or table number and add the caption to the relevant contents listing file.

The command `\contcaption{`⟨*text*⟩`}` may be used instead. This command neither increments the number nor adds anything to the listing files.

EXAMPLE 17 – The following code:

```
\begin{table}[tbp]
\centering
\caption{Example table in three parts} \label{tab:cont}
\begin{tabular}{|c|c|} \hline
{\bf Col 1} & {\bf Col 2} \\ \hline\hline
A & 1 \\
B & 2 \\ \hline
```

Table 3 – Example table in three parts

Col 1	**Col 2**
A	1
B	2

Table 3 – (continued)

Col 1	**Col 2**
C	3
D	4

```
\end{tabular}
\end{table}

\begin{table}[tbp]
\centering
\contcaption{(continued)}
\begin{tabular}{|c|c|} \hline
{\bf Col 1} & {\bf Col 2} \\ \hline\hline
C & 3 \\
D & 4 \\ \hline
\end{tabular}
\end{table}

\begin{table}[tbp]
\centering
\contcaption{(concluded)}
\begin{tabular}{|c|c|} \hline
{\bf Col 1} & {\bf Col 2} \\ \hline\hline
E & 5 \\
F & 6 \\ \hline
\end{tabular}
\end{table}
```

produces the three-part table 3.

Table 3 – (concluded)

Col 1	**Col 2**
E	5
F	6

7.5 Title command

The command to produce the title of the standard is `\title{`⟨*intro*⟩`}{`⟨*main*⟩`}{`⟨*compl*⟩`}`. The command takes three parameters according to the three elements of the title as specified in the ISO directives.

a) ⟨*intro*⟩ The introductory element of the title.

b) ⟨*main*⟩ The main element of the title.

c) ⟨*compl*⟩ The complementary element of the title.

EXAMPLE 18 – If this were actually meant to be an ISO standard, then the LaTeX source for the title of this document might be:

```
\title{Industrial automation systems and integration}%
{Product data representation and exchange}%
{Part 3456 : Documentation methods: The LaTeX style
file reference manual}
```

NOTE – There must be a space on either side of the colon separating the part number and the final part of the title.

Three other commands are used for setting the running head throughout the document. These must be placed in the preamble.

The command `\standard{`⟨*number of standard*⟩`}` is used to identify the standard.

The command `\yearofedition{`⟨*year*⟩`}` is used to identify the year of the edition.

The command `\languageofedition{`⟨*code*⟩`}` is used to identify the language of the edition.

EXAMPLE 19 – This document is identified by

```
\standard{LaTeX for standards}
\yearofedition{1995}
\languageofedition{(E)}
```

The `\title` command sets the page numbering style to be arabic, starts a new page, numbered 1, and puts the title at the start of the page. If the options `draft`, `notcopyright`, `dis` or `otherdoc` have not been called for, then the standard ISO header is put at the top of the page.

7.6 Cross referencing commands

The usual LaTeX `\label` and `\ref` commands are supported. The class also provides some formatted referencing commands in addition to `\ref`.

The following commands are useful for referring to figures, clauses etc. Each takes a parameter that is the parameter of a LaTeX `\label{}` command.

– The command `\aref{biblio}` prints annex F while `\ref{biblio}` prints F

– The command `\bref{lamport}` prints [1] while `\ref{lamport}` prints 1

- The command `\cref{sec:nrefs}` prints clause 2 while `\ref{sec:nrefs}` prints 2
- The command `\eref{sec:nrefs}` prints example 2 while `\ref{sec:nrefs}` prints 2
- The command `\fref{sec:nrefs}` prints figure 2 while `\ref{sec:nrefs}` prints 2
- The command `\tref{sec:nrefs}` prints table 2 while `\ref{sec:nrefs}` prints 2

7.7 Heading commands

7.7.1 Foreword

The `foreword` environment specifies that a table of contents, list of figures and list of tables be produced, and starts a new unnumbered clause called Foreword. Formatting is one column style only and roman page numbering is set. The table of contents starts on page ii.

More precisely, the title of the clause is given by the value of `\forewordname` (see annex A).

> NOTE – The default style only lists level 1 clauses in the table of contents. If you need a more detailed listing, then put the command `\setcounter{tocdepth}{<level #>}` in the preamble. For example `\setcounter{tocdepth}{3}` will produce a contents listing down to the level of `\ssclause` (see table 2 for the definitions of clause levels).

The command `\fwdbp` prints the ISO specified boilerplate for the initial paragraphs of a foreword.

7.7.2 Introduction

The `introduction` environment starts a new unnumbered clause called 'Introduction' with one column formatting.

More precisely, the title of the clause is given by the value of `\introductionname` (see annex A).

7.7.3 Scope

The `\scopeclause` command starts a new numbered clause called 'Scope'.

More precisely, the title of the clause is given by the value of `\scopename` (see annex A).

7.7.4 Normative references clause

The `\normrefsclause` command starts a new numbered clause called 'Normative references'.

More precisely, the title of the clause is given by the value of `\normrefsname` (see annex A).

The command `\normrefbp{`⟨*document identifier*⟩`}` prints the ISO required text for the introduction to the listing of normative references. The ⟨*document identifier*⟩ parameter must be such that it reads sensibly in a sentence of the form '...constitute provisions of this ⟨*document identifier*⟩.'.

Table 4 – Definition, symbol and abbreviation clause commands

Command	Title	Default text
`\defclause`	`\defname`	Definitions
`\defsymabbclause`	`\defsymabbname`	Definitions, symbols and abbreviations
`\defsymclause`	`\defsymname`	Definitions and symbols
`\defabbclause`	`\defabbname`	Definitions and abbreviations
`\symabbclause`	`\symabbname`	Symbols and abbreviations

EXAMPLE 20 – Clause 2 in this document was started by the commands:

```
\normrefsclause \label{sec:nrefs}

\normrefbp{manual}
\begin{nreferences}
...
```

7.7.5 Definitions, symbols and abbreviations

A variety of commands are provided that initiate new numbered clauses for definitions, symbols and abbreviations. Depending on the amount of material in these respective categories, one or more clauses may be used. The commands and clause titles are listed in table 4.

7.7.6 Bibliography

The command `\bibannex` starts an informative annex entitled 'Bibliography'. Or, more precisely, by the value of the `\bibname` command.

7.8 Version control commands

A set of commands are provided to assist when revising a document. For these commands to flag the changes in the printed document the command `\changemarkstrue` must be put in the preamble.

In the commands described below, the ⟨*number*⟩ parameter can be used to correlate changes in a document to some external (numbered) requirement for the change.

7.8.1 Editorial

The command `\editorial{`⟨*number*⟩`}` flags an editorial change to the document text with ED^{number}.

EXAMPLE 21 – Just to start things off, this is an original sentence, which should take up about a line. This example sentence contains an editorial change. The command `\editorial{37}` was ED^{37} included in the previous sentence. This sentence, though, has no changes marked in it and may be assumed to be unaltered from a prior version.

7.8.2 Added

The command `\added{`⟨*text*⟩`}{`⟨*number*⟩`}` flags the position of the additional ⟨*text*⟩ and highlights it.

> EXAMPLE 22 – Just to start things off, this is an original sentence, which should take up about
> 27 ⇒ a line. This example sentence contains *some added text* in the middle of it. The command `\added{some added text}{27}` was included in the previous sentence. This sentence, though, has no changes marked in it and may be assumed to be unaltered from a prior version.

7.8.3 Deleted

The command `\deleted{`⟨*number*⟩`}` flags the position of deleted text.

> EXAMPLE 23 – Just to start things off, this is an original sentence, which should take up about
> 34 ⇐ a line. Some text was deleted from the middle of this sentence. The command `\deleted{34}` was included in the previous sentence. This sentence, though, has no changes marked in it and may be assumed to be unaltered from a prior version.

7.8.4 Moved

The command `\moved{`⟨*text*⟩`}{`⟨*number*⟩`}` flags the position of moved ⟨*text*⟩ and highlights it.

> EXAMPLE 24 – Just to start things off, this is an original sentence, which should take up about a
> 81 ⇔ line. This sentence contains *some moved* text in the middle of it. The command `\moved{some moved}{81}` was included in the previous sentence. This sentence, though, has no changes marked in it and may be assumed to be unaltered from a prior version.

8 The isorot facility

The isorot facility enables the rotation of document elements on a page. It uses the LaTeX `\special` command to perform its effects, and thus can only be used with a limited number of dvi to postscript programs. The facilities available are summarized in table 5.

isorot is a modification of the rotation.sty file created by Rahtz and Barroca [2]. Further examples of the usage of their style are given in Goosens *et al* [3].

> NOTE 1 – Several examples of the effects of the commands described herein are shown. In many cases the results are not pretty. This should act as a warning that dealing with rotational elements requires more care than for the usual document elements.

8.1 DVI drivers

The isorot facility supports only a limited number of dvi to postscript translators. The default translator is *dvips*. The following command must be put in the preamble of the document if *dvips* is not being used: `\rotdriver{<drivername>}`, where `<drivername>` is one of the following:[3)]

[3)] I have been able to try the `dvips` driver but not the others. If anyone is able to try the other drivers, or to extend the range of drivers, I would like to be given the results.

Table 5 – The rotation facilities

Facility	**Effect**
	Commands
\rotdriver{<driver>}	declare the name of the dvi to Postscript translator (default dvips)
\clockwise	sets rotation direction clockwise for positive angles (the default)
\counterclockwise	sets rotation direction counterclockwise for positive angles
\figuresright	sets rotation direction for sideways floats counterclockwise (the default)
\figuresleft	sets rotation direction for sideways floats clockwise
\rotcaption	like the caption command, but rotates the caption through 90 degrees
\controtcaption	like the contcaption command, but rotates the caption through 90 degrees
	Environments
sideways	rotates the contents through 90 degrees counterclockwise
turn	rotates the contents through the given angle
rotate	rotates the contents through the given angle, but no space allowed for the result
sidewaystable	like the table environment, but rotated 90 degrees
sidewaysfigure	like the figure environment, but rotated 90 degrees
landscape	prints all enclosed pages in landscape mode

a) `dvips` for the *dvips* translator;

b) `dvipsone` for the *dvipsone* translator;

c) `dvitops` for the *dvitops* translator;

d) `pctexps` for the *pctexps* translator;

e) `pubps` for the Ann Arbor *pubps* translator.

f) `textures` for the *Textures* translator;

8.2 Rotational directions

isorot enables the textual and other elements of a document to be rotated from their normal horizontal layout. In some cases elements can be rotated through arbitrary angles, whereas in others only 90 degree rotation is possible.

By default, a rotation through a positive number of degrees corresponds to a clockwise rotation. The command `\counterclockwise` sets the following rotations to be counterclockwise for positive angles. The command `\clockwise` sets the following rotations to be clockwise for positive angles. These commands can be used to toggle the rotational behavior.

Rotated floating environments are normally rotated so that they are printed with a counterclockwise rotation (i.e. the original bottom of the float is placed at the right hand side of the paper), which is what is normally required. This behavior can be altered by the command `\figuresleft`, which will give the reverse effect. The command `\figuresright` will set the behavior to the default. These commands can be used to toggle the rotational behavior of floats.

8.3 Rotation of text

The `sideways` environment rotates the contents of the environment by 90 degrees counterclockwise, and leaves space for the result.

The `turn` environment rotates the contents by the given number of degrees in the direction specified by the most recent of the `\clockwise` or `\counterclockwise` commands, leaving space for the result.

The `rotate` environment rotates the contents by the given number of degrees in the direction specified by the most recent of the `\clockwise` or `\counterclockwise` commands, but no arrangements are made for leaving space for the result.

EXAMPLES

25 – Some simple rotations:

This code

```
Default rotation direction: \\
```

```
A
\begin{sideways}%
B C
\end{sideways}
D E F G H I J K L M
\begin{turn}{-90}%
Minus 90 turn
\end{turn}
N O P
\begin{rotate}{90}%
Plus 90 rotate
\end{rotate}
Q \\
and continue on with another line after rotations.
```

produces the following (note how space is allowed for the `turn`ed text, whereas the `rotate`d text runs into the text below).

Default rotation direction:

A B C D E F G H I J K L M Minus 90 turn N O P Plus 90 rotate Q
and continue on with another line after rotations.

26 – This example shows the effect of using the `\counterclockwise` command.

This code

```
Flip rotation direction: \\
\counterclockwise
A
\begin{sideways}%
B C
\end{sideways}
D E F G H I J K L M
\begin{turn}{-90}%
Minus 90 turn
\end{turn}
N O P
\begin{rotate}{90}%
Plus 90 rotate
\end{rotate}
Q \\
Set rotation direction back to default value.
\clockwise
```

produces the following, which should be compared with example 25.

Flip rotation direction:

A B C D E F G H I J K L M Minus 90 turn N O P Plus 90 rotate Q

Set rotation direction back to default value.

Although the examples so far have only shown the rotation of text, boxes can also be rotated.

EXAMPLE 27 – Rotating a box.

This code

```
\newsavebox{\foo}
\newlength{\fool}
\settowidth{\fool}{Hurrah for ISO.}
\savebox{\foo}{\parbox{\fool}{Hurrah for ISO. Hurrah for ISO.
                               Hurrah for ISO. Hurrah for ISO.}}
Start
\usebox{\foo}
\&
\begin{turn}{-45}\usebox{\foo}\end{turn}
\&
\begin{turn}{45}\usebox{\foo}\end{turn}
End
```

produces:

Start Hurrah for ISO. Hurrah for ISO. Hurrah for ISO. Hurrah for ISO. & Hurrah for ISO. Hurrah for ISO. Hurrah for ISO. Hurrah for ISO. & Hurrah for ISO. Hurrah for ISO. Hurrah for ISO. Hurrah for ISO. End

Elements can be rotated through arbitrary angles, and also rotated elements can be nested inside other rotated elements.

EXAMPLES

28 – Repeated rotation:

The following example code shows that text can be rotated through any angle. The result is shown in figure 1.

```
\newcount\prwc
```

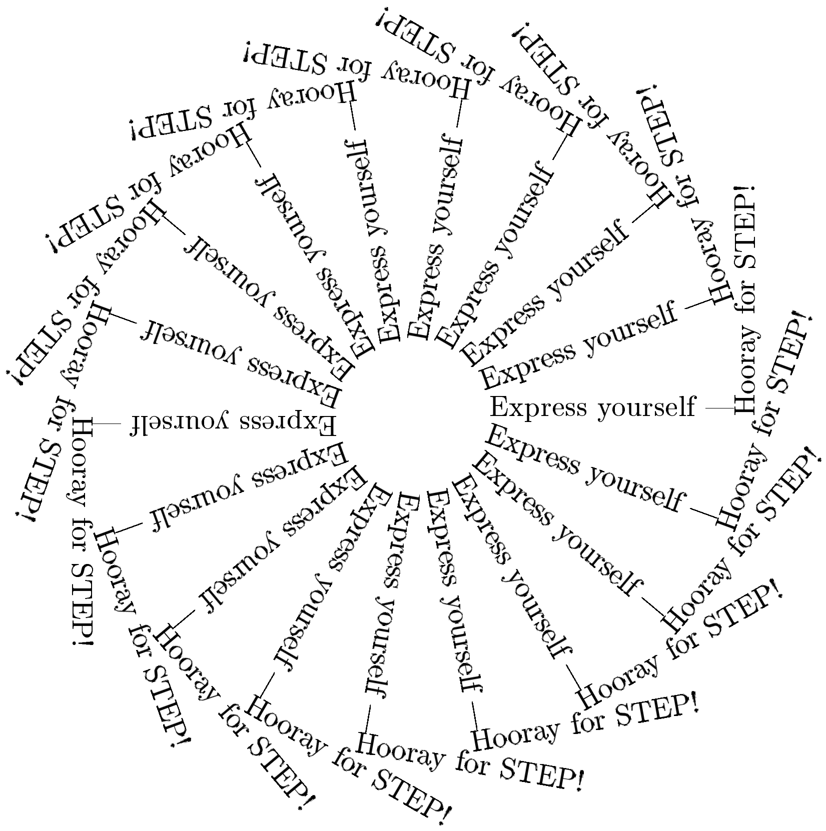


Figure 1 – Example rotation through multiple angles

```
\newsavebox{\prwtext}
\newdimen\prwspace
\def\wheel#1#2{%
  \savebox{\prwtext}{#1\begin{sideways}#2\end{sideways}}%
  \prwspace\wd\prwtext%
  \advance\prwspace by 1cm%
  \centerline{%
  \rule{0pt}{\prwspace}%
  \rule[-\prwspace]{0pt}{\prwspace}%
  \prwc=-180\loop\ifnum\prwc<180
  \rlap{\begin{rotate}{\the\prwc}%
  \rule{1cm}{0pt}\usebox{\prwtext}\end{rotate}}%
  \advance\prwc by 20\repeat}}
\begin{figure}
\wheel{Express yourself ---}{Hooray for STEP!}
\caption{Example rotation through multiple angles}
\label{fig:wheel}
\end{figure}
```

Figures 4 and 5 also show rotations through a range of angles, both positive and negative.

29 – Nested rotations.

This code

```
    Here is some text before a \verb|sideways| environment.
And some more, and more and more garble gobble cluck
click clack clock cluck and so on and on and on.
\begin{center}
\begin{sideways}
\rule{1in}{0pt}
\begin{tabular}{|lr|}
\begin{rotate}{-45}{\em Word}\end{rotate} & \begin{rotate}{-90}%
Occurrences\end{rotate}
\\
\hline
hello & 33 \\
goodbye & 34 \\
\hline
\end{tabular}
\end{sideways}
\end{center}
    Here is some text after a \verb|sideways| environment.
 And some more, and more and more garble gobble cluck
click clack clock cluck and so on and on and on.
```

produces:

Here is some text before a `sideways` environment. And some more, and more and more garble gobble cluck click clack clock cluck and so on and on and on.

Word	Occurrences
hello	33
goodbye	34

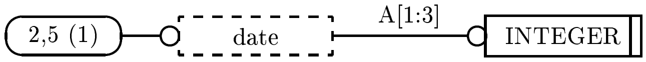
Here is some text after a `sideways` environment. And some more, and more and more garble gobble cluck click clack clock cluck and so on and on and on.

8.4 Rotations of tables and figures

The previous examples have demonstrated the rotation of textual elements. For instance, example 29 shows that tabular material can be rotated using the `sideways` environment. (Actually, any of the previously mentioned environments may be used.) Two further environments are provided which rotate a LaTeX float through 90 degrees. These are:

- `sidewaystable`, which corresponds to the standard LaTeX `table` environment; and
- `sidewaysfigure`, which corresponds to the standard LaTeX `figure` environment.

Figure 2 – Example figure with a standard caption.



However, the correspondence with the standard environments is not strictly complete as a sideways float is alway placed on a page by itself.

The direction of rotation may be controlled by the `\figuresright` and `\figuresleft` commands.

EXAMPLE 30 – Code for table 5.

```
\begin{sidewaystable}
\centering
\caption{The rotation facilities} \label{tab4}
\begin{tabular}{|l|l|} \hline
{\bf Facility} & {\bf Effect} \\ \hline
\multicolumn{2}{|c|}{{\bf Commands}} \\ \hline
\verb|\rotdriver{<driver>}| &
declare the name of the dvi to Postscript translator (default {\tt dvips}) \\
......
\verb|sidewaysfigure| &
like the \verb|figure| environment, but rotated 90 degrees \\ \hline
\end{tabular}
\end{sidewaystable}
```

8.5 Rotation of float captions and bodies

Sometimes it may be useful to rotate a caption independently of the rotation of a figure or table. The command `\rotcaption` is analogous to the normal `\caption` command, and inserts the caption rotated by 90 degrees. There is also the companion command `\controtcaption` for continuation captions.

EXAMPLES

31 – Float with a regular caption.

Figure 2 is produced by the code below.

```
\begin{figure}
\centering
\caption{Example figure with a standard caption.} \label{fig:nocrot}
\setlength{\unitlength}{0.2in}
\footnotesize
\begin{picture}(17,2)
\thicklines
\put(0,0){\begin{picture}(4,1)
  \put(1.5,0.5){\oval(3,1)}
  \put(1.5,0.5){\makebox(0,0){2,5 (1)}}
```

```
  \put(3,0.5){\line(1,0){1.0}}
  \put(4.25,0.5){\circle{0.5}}
  \end{picture}}
\put(4.5,0){\begin{picture}(8,1)
  \put(0,0){\dashbox{0.25}(4,1){date}}
  \put(4,0.5){\line(1,0){3.5}}
  \put(7.75,0.5){\circle{0.5}}
  \put(6,1){\makebox(0,0){A[1:3]}}
  \end{picture}}
\put(12.5,0){\begin{picture}(4,1)
  \put(0,0){\framebox(4,1){INTEGER}}
  \put(3.75,0){\line(0,1){1}}
  \end{picture}}
\end{picture}
\normalsize
\setlength{\unitlength}{1pt}
\end{figure}
```

32 – Float with a rotated caption.

Figure 3 is produced by the code below.

```
\begin{figure}
\centering
\rotcaption{Figure~\protect\ref{fig:nocrot} with a rotated caption.}
 \label{fig:crot}
\setlength{\unitlength}{0.2in}
\footnotesize
\begin{picture}(17,2)
...
\end{picture}
\normalsize
\setlength{\unitlength}{1pt}
\end{figure}
```

As can be seen from figure 3 the advisability of rotating a caption depends on the size of the body of the float. It may be better in certain cases to leave the caption in its regular position and rotate the body of the float instead.

EXAMPLE 33 – Regular caption and float.

Figure 4 is a regular figure and caption. It is produced by the following code.

```
\def\prwrot#1{%
\settowidth{\fool}{ISOROT}
\savebox{\foo}{\parbox{\fool}{ISOROT ISOROT ISOROT ISOROT}}%
\framebox{---\begin{turn}{#1}\framebox{\usebox{\foo}}\end{turn}---}}%
\def\degrees{{\small$^{o}$}}

\begin{figure}
\centering
```

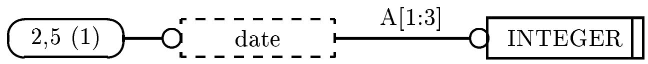


Figure 3 – Figure 2 with a rotated caption.

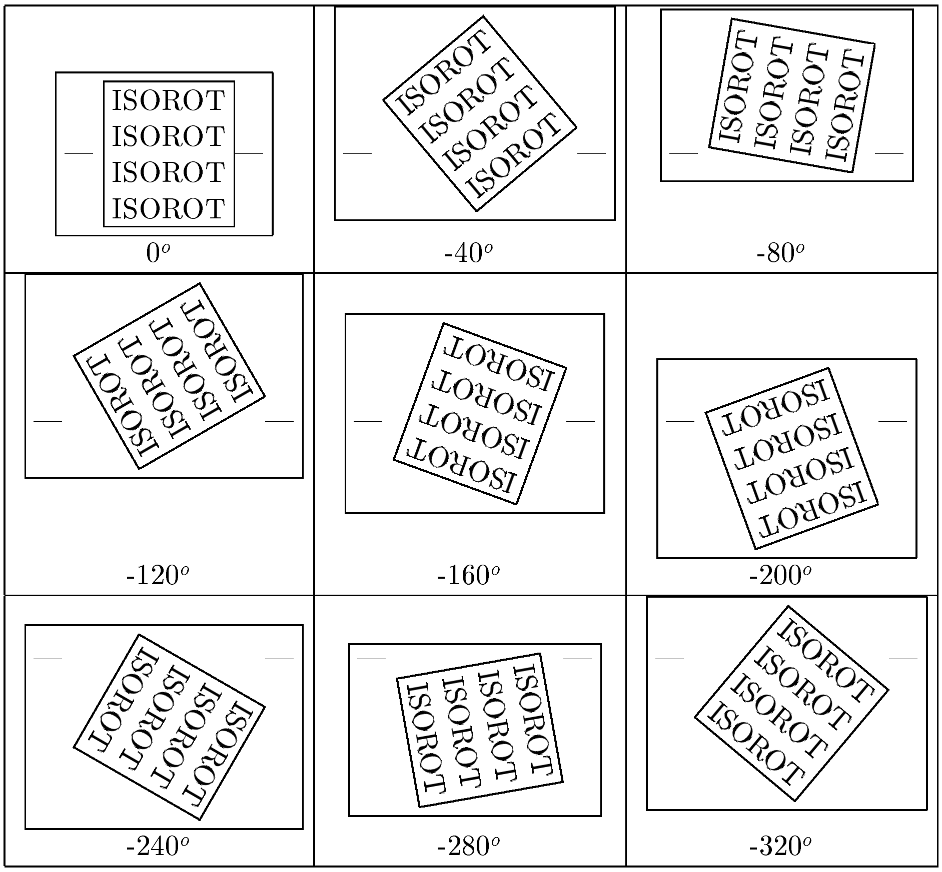


Figure 4 – Rotation of paragraphs between 0 and -320 degrees

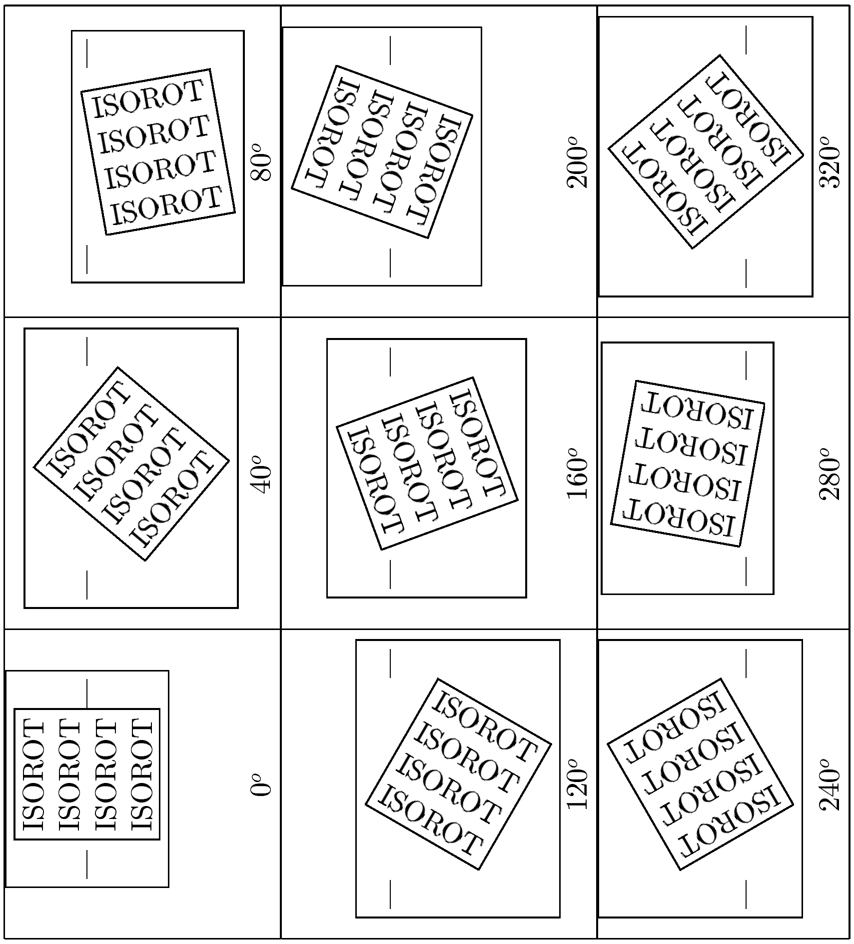


Figure 5 – Rotation of paragraphs between 0 and 320 degrees (with figure body turned sideways)

```
\begin{tabular}{|c|c|c|} \hline
\prwrot{0} &\prwrot{-40}&\prwrot{-80}\\
0\degrees & -40\degrees & -80\degrees \\ \hline
\prwrot{-120}&\prwrot{-160}&\prwrot{-200}\\
-120\degrees & -160\degrees & -200\degrees \\ \hline
\prwrot{-240}&\prwrot{-280}&\prwrot{-320}\\
-240\degrees & -280\degrees & -320\degrees \\ \hline
\end{tabular}
\caption{Rotation of paragraphs between 0 and -320 degrees} \label{fig:angles1}
\end{figure}
```

EXAMPLE 34 – Regular caption and rotated float body.

Figure 5 is a regular figure and caption where the figure contents have been rotated. It was produced by the following code.

```
\begin{figure}
```

```
\centering
\begin{sideways}
\begin{tabular}{|c|c|c|} \hline
\prwrot{0} &\prwrot{40}&\prwrot{80}\\
0\degrees & 40\degrees & 80\degrees \\ \hline
\prwrot{120}&\prwrot{160}&\prwrot{200}\\
120\degrees & 160\degrees & 200\degrees \\ \hline
\prwrot{240}&\prwrot{280}&\prwrot{320}\\
240\degrees & 280\degrees & 320\degrees \\ \hline
\end{tabular}
\end{sideways}
\caption[Rotation of paragraphs between 0 and 320 degrees]%
        {Rotation of paragraphs between 0 and 320 degrees (with figure
         body turned sideways)}\label{fig:angles2}
\end{figure}
```

8.6 Landscaping

LaTeX normally prints in portrait mode. The `landscape` environment prints all the enclosed stuff in landscape mode, except for headers and footers which are not rotated.

EXAMPLE 35 – Landscaping

The source for this part of the document is:

```
\begin{landscape}
\sclause{Landscaping}

    \LaTeX\ normally prints in portrait mode. The ...
...
... long, wide tables.
\end{landscape}
```

The environment starts by clearing the current page and then switches to portrait mode. At the end of the environment the current page is cleared and the next page is back to normal portrait mode.

All the other rotation commands and environments produce boxes and LaTeX will not break a box across a page. The `landscape` environemt does not produce a box and so many pages can be printed in landscape mode with LaTeX taking care of the page breaking for you.

Landscape mode is not particularly useful for normal text as the lines are far too long for comfortable reading. Where it can be useful is where you have a table that is too wide to fit on a portrait page, so needs to be rotated, yet is also too long to fit on the page when it is rotated. The `supertabular`, the `longtable`, and the `xtab` packages provide facilities for automatically breaking long tables across pages. Any of these can be used in conjunction with landscaping to both rotate and automatically page break long, wide tables.

9 The xtab facility

The `xtab` facility is an extension of the `supertabular` package. The extension provides for the specification of a header to go on the last page of a long table.

The commands available are given in table 6.

Table 6 – The xtab facility commands

Command	Effect
`\begin{supertabular}{...}`	This is equivalent to the normal `\begin{tabular}{...}` environment. You supply the specification of the columns just as for the normal tabular environment. All commands that can be used within a tabular environment can also be used within the supertabular environment. Unlike the tabular environment which prevents page breaking within the tabular, the supertabular allows page breaking, so that tabulars can extend automatically across several pages. Supertabular starts off with a tabular environment and checks the amount of space left on the page as it adds each row to the tabular. If the space left on the page is too short for another row, then it ends the current tabular, performs a page break and starts another tabular on the following page. This process is repeated until all the rows have been output. There are special commands for captioning a supertabular as a table, and also stuff can be automatically inserted after each (internal) `\begin{tabular}` and immediately before each `\end{tabular}`. Do not put a supertabular in a table environment, as the table environment keeps its contents on a single page (presumably you are using supertabular because its contents are longer than one page.
	Continued on next page

Table 6 – continued from previous page

Command	Effect
\end{supertabular}	End the supertabular environment.
	Note: If any of the following commands are used, then they should be placed before the particular supertabular environment that they apply to.
\topcaption{...}	A command to provide a caption for the table. The caption is placed at the top of the table.
\bottomcaption{...}	A command to provide a caption for the table. The caption is placed at the bottom of the table.
\tablecaption{...}	A command to provide a caption for the table. The caption is placed at the default position, which is at the top of the table. **Notes:** You cannot use the `caption` command but you can put a label after any of these captioning commands. If you want captioning, the command must be specified before the start of the supertabular environment. The `\...caption{}` command(s) remain in effect until changed by another `\...caption` command.
\tablefirsthead{...}	Defines the contents of the first occurence of the tabular head. The tabular head is some special treatment of the first row in the table. This command is optional. If used, the header must be closed by the end of line command for tabulars (e.g., \\).
\tablehead{...}	Defines the contents of the table head on subsequent pages. For example, you might want to note that this is a continuation of the table on the previous page, as well as repeating any column headings that were given at the start of the supertabular by `\tablefirsthead`.
\tablelasthead{...}	Defines the contents of the table head on the last page of the table. This works by writing to the `.aux` file the number of pages that the supertabular extends over. When the supertabular comes to the last tabular (which it calculates from the information it reads from the `.aux` file) it replaces the contents of `\tablehead` by the contents of `\tablelasthead`.
	Continued on next page

Table 6 – concluded from previous page

Command	**Effect**
	It makes no attempt to measure the amount of space used by the last table head, so if this is more than for `tablehead` then the tabular might be a litle too long. In any case, when using this command, the document has to be LaTeXed at least twice, just as it has to be to resolve references and so on.
`\notablelasthead`	Switches off the last `\tablelasthead`. A `tablelasthead` stays in effect until overwritten by a new `\tablelasthead` or cancelled by this command.
`\tabletail{...}`	The contents of this command are inserted before the `\end{tabular}` on each page except for the last page of the table. For example, you might want to note that the table is continued on the next page.
`\tablelasttail{...}`	The contents of this command are inserted before the final `\end{tabular}` of the table. For example, you might want to note that this is where the table ends.

The table is only broken between rows — a row will not be split across pages. This can lead to some bad page breaks, especially if there are rows with a large vertical height (like some in table 6). It is best to keep rows not too tall. As well, sometimes the first or last page of a table will only have 'headers' and 'footers' with nothing between. To counteract this, either the table must be redesigned (taller or shorter rows) or shifted a bit in the source document so that it starts in a different place on a page.

EXAMPLE 36 – Table 6 was produced by the following code:

```
\topcaption{The xtab facility commands} \label{tab:xtab}
\tablefirsthead{\hline \multicolumn{1}{|c|}{{\bf Command}} &
                      \multicolumn{1}{c|}{{\bf Effect}} \\ \hline }
\tablehead{\multicolumn{2}{c}%
           {{\bf \tablename\ \thetable{} -- continued from previous page}} \\
  \hline   \multicolumn{1}{|c|}{{\bf Command}} &
           \multicolumn{1}{c|}{{\bf Effect}} \\ \hline }
\tablelasthead{\multicolumn{2}{c}%
           {{\bf \tablename\ \thetable{} -- concluded from previous page}} \\
  \hline   \multicolumn{1}{|c|}{{\bf Command}} &
           \multicolumn{1}{c|}{{\bf Effect}} \\ \hline }
\tabletail{\hline \multicolumn{2}{|r|}{{Continued on next page}} \\ \hline}
\tablelasttail{\hline \hline}

\begin{center}
\begin{supertabular}{|l|p{0.5\textwidth}|}
\verb|\begin{supertabular}{...}| & This is equivalent to the normal
                        \verb|\begin{tabular}{...}| environment.
```

```
                          You supply the specification of the columns
                          just as for the normal tabular environment.
                          All commands that can be used within a tabular
                          environment can also be used within
                          the supertabular environment.
 \\
 &
    Unlike the tabular environment which prevents page breaking
within the tabular, the supertabular allows page breaking, so that
tabulars can extend automatically across several pages. Supertabular
... ... ...
\verb|\tablelasttail{...}| & The contents of this command are inserted before
                          the final \verb|\end{tabular}| of the table.
 \\
 &
    For example, you might want to note that  this is where
the table ends.
 \\
\end{supertabular}
\end{center}
```

10 The askinc facility

The askinc facility contains LaTeX code to enable the interactive input of files. This functionality is a cross between the LaTeX `\include` and `\includeonly` commands, and the `\input` command.

If using the current LaTeX system, put the command `\usepackage{askinc}` in the document preamble. Otherwise call askinc as an option to the `\documentstyle` command.

In the body of the root source document, use the command `\infile{`⟨*file*⟩`}` for each ⟨*file*⟩ that comprises the document. That is, the command `\infile` is similar to the `\input` and `\include` commands.

When LaTeXing the root document, you will be asked to provide a comma-separated list of the `\infile`s to be processed (similar to the argument to the `\includeonly` command). If you want all the files to be processed, just hit the `<RETURN>` key (or its equivalent).

Like `\include`d files, a file that is `\infile`d into a document shall not itself contain another `\infile`d file.

EXAMPLE 37 – The following root file has three files that are `\infile`d.

```
\documentclass[...]{...}
\usepackage{askincv}
% other preamble stuff
\begin{document}
% perhaps some stuff
\infile{file1}
\infile{file2}
\infile{file3}
% perhaps more stuff
```

```
\end{document}
```

Annex A
(normative)
Additional commands

A.1 Language configuration commands

There is an additional set of commands in the iso facility that are specified here. This set has been provided to enable the style to be easily configured for a non-English language.

The modified commands may be put in the document preamble or, preferably, placed in a separate .sty file and called as a package. This latter option improves reuseability.

A.1.1 Words and phrases

To produce a non-English version of the iso facility, the relevant commands from the following list may require re-definition.

NOTE 1 – In the listing, the default values are printed *in this font* to distinguish them from the explanatory text.

– `\annexname`: Header for an annex. Default value is: *Annex*

– `\contentsname`: Header for table of contents listing. Default value is: *Contents*

– `\copyrightname`: The copyright owner. Default value is: *ISO*

– `\examplename`: Identification of an example. Default value is: *EXAMPLE*

– `\examplesname`: Header for a list of examples. Default value is: *EXAMPLES*

– `\figurename`: Start of the caption for a figure. Default value is: *Figure*

– `\forewordname`: Title of the Foreword. Default value is: *Foreword*

– `\indexname`: Header for the index. Default value is: *Index*

– `\informativename`: Identification of an informative annex. Default value is: *informative*

– `\inscopename`: Introduction to in-scope listing. Default value is: *The following are within the scope of this*

– `\ISname`: 'INTERNATIONAL STANDARD' as used in the header for an IS title page. Default value is: *INTERNATIONAL STANDARD*

– `\introductionname`: Title of the Introduction. Default value is: *Introduction*

- `\listannexname`: Header for list of annexes listing. Default value is: *Annexes*
- `\listfigurename`: Header for list of figures listing. Default value is: *Figures*
- `\listtablename`: Header for list of tables listing. Default value is: *Tables*
- `\normativename`: Identification of a normative annex. Default value is: *normative*
- `\normrefsname`: Title of normative references clause. Default value is: *Normative references*
- `\notename`: Identification of a note. Default value is: *NOTE*
- `\notesname`: Header for a list of notes. Default value is: *NOTES*
- `\outofscopename`: Introduction to out-of-scope listing. Default value is: *The following are outside the scope of this*
- `\pagename`: The word for the page header in the table of contents. Default value is: *Page*
- `\scopename`: Title of the Scope. Default value is: *Scope*
- `\sectionname`: The word for a 'section'. Default value is: *Section*
- `\tablename`: Start of the caption for a table. Default value is: *Table*
- `\tbpname`: Footnote text for 'to be published.'. Default value is: *To be published.*

The following commands provide the names for referenced document elements.

- `\annexrefname`: Reference to an annex. Default value is: *annex*
- `\clauserefname`: Reference to a clause. Default value is: *clause*
- `\examplerefname`: Reference to an example. Default value is: *example*
- `\figurerefname`: Reference to a figure. Default value is: *figure*
- `\tablerefname`: Reference to a table. Default value is: *table*

NOTES

2 – The above commands, may be changed via the LaTeX `\renewcommand`.

3 – The LaTeX command `\today` will probably also require modification. This is not something for the casual user to attempt.

EXAMPLE 38 – The following is a partial list of the commands to convert to a French language style.

```
\renewcommand{\annexname}{Annexe}
\renewcommand{\contentsname}{Sommaire}
\renewcommand{\examplename}{EXEMPLE}
\renewcommand{\examplesname}{EXEMPLES}
\renewcommand{\forewordname}{Avant-propos}
\renewcommand{\ISname}{NORME INTERNATIONALE}
\renewcommand{\listtablename}{Tableaux}
\renewcommand{\scopename}{Domaine d'application}
\renewcommand{\tablename}{Tableau}
```

A.1.2 Boilerplate

Some commands print boilerplate text; again, the default text is in English. For iso there are three such commands. The first is `\copyrightnotice` which contains the text of the copyright notice for an International Standard. This can be changed via the `\renewcommand` command.

The second is `\normrefbp` which prints the boilerplate for the introduction to the Normative references clause. Like the `\copyrightnotice` command, this can be redefined using the `\renewcommand`.

The other is the `\fwdbp` command which `\input`s the boilerplate text from a file called isofwdbp.tex. For non-English text either modify the contents of this file or create a new file and modify the `\fwdbp` command to call in the new file.

EXAMPLES

39 – This is how `\normrefbp` could be written for the French language.

```
\renewcommand{\normrefbp}[1]{%
    Les normes suivantes contiennent des dispositions qui, par
suite de la r\'{e}f\'{e}nce qui en est faite, constituent des
dispositions valables pour la pr\'{e}sente #1.
Au moment de la publication, les \'{e}ditions indiqu\'{e}es
\'{e}taient en viguer. Toute norms est sujette \`{a} r\'{e}vision et
les parties prenantes des accords fond\'{e}s sur la pr\'{e}sente #1
sont invit\'{e}es \`{a} rechercher la possibilit\'{e} d'appliquer
les \'{e}ditions les plus r\'{e}centes des normes indiqu\'{e}es
ci-apr\`{e}s. Les membres de la CEI et de l'ISO poss\`{e}dent
le registre des Normes Internationales en vigueur \`{a} un
moment donn\'{e}.
}% end renewcommand
```

40 – Given that `\normrefbp` has been redefined as in example 39, then the command `\normrefbp{norme internationale}` will print:

Les normes suivantes contiennent des dispositions qui, par suite de la réfénce qui en est faite, constituent des dispositions valables pour la présente norme internationale. Au moment de la publication, les éditions indiquées étaient en viguer. Toute norms est sujette à révision et les parties prenantes des accords fondés sur la présente norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes Internationales en vigueur à un moment donné.

Annex B
(normative)

Ordering of LaTeX commands

The LaTeX commands to produce an ISO standard document are:

```
%             either
\documentclass[<options>]{iso}           % for LaTeX 2e
\usepackage{<name>}                      % additional packages (LaTeX 2e)
%              or
\documenstyle[<options>]{iso}            % for LaTeX v2.09

\standard{<standard number>}
\yearofedition{<year>}
\languageofedition{<code>}
  % other preamble commands
\begin{document}
\foreword                           % start Foreword
  \fwdbp                            % boilerplate
  % other text
\introduction                       % start Introduction
  % text
\title{<intro>}{<main>}{<compl>}    % the title
\scopeclause                        % start scope clause
   % text of scope
\normrefsclause                     % start norm refs clause
  \normrefbp                        % boilerplate
  \begin{nreferences}
    % \item list of normative references
  \end{nreferences}
\defclause                          % definitions clause
  % definitions
\symabbclause                       % Symbols & abbreviations clause
  % symbol lists
% THE BODY OF THE DOCUMENT
\bibannex                           % optional; the final Bibliography annex
  % bibliography listing
% the index
\end{document}
```

NOTE 1 – Remember that in the above `iso` should be replaced by the current iso version for the LaTeX system being used.

Annex C
(informative)
Creating an index

LaTeX, although providing some assistance in preparing the data for an index, only does part of the job. The command `\index{text}` writes out `text` to an .idx file in the following format:

```
\indexentry{text}{pg}
```

where `pg` is the page number in the document where the `\index` command occurred.

The `theindex` environment is used for printing an index. The format of this is:

```
\begin{theindex}
\item <text and page numbers>
  \subitem <text and page numbers>
    \subsubitem <text and page numbers>
.
.
\end{theindex}
```

where `\item` is a major topic entry, `\subitem` is a sub-topic entry, and `\subsubitem` is a sub-subtopic entry. The command `\indexspace` can be used to add space between the entries. Other text and commands can also occur within the environment.

LaTeX provides no help in going from an .idx file to the `theindex` environment [1]. The data in the file has to be sorted, duplicate page numbers deleted, etc, and then re-written in the desired environment format.

The typical process for producing a document with an index is:

a) Prepare the source file, say fred.tex, with the command `\makeindex` in the preamble, and `\index` commands within the body of the text.

b) Run LaTeX. As well as producing the usual output files, it will also produce the file fred.idx.

c) By some means produce a file, let's call it fredidx.tex, from the data in fred.idx that contains the appropriate `theindex` formatted data.

d) Run LaTeX again on fred.tex which now has to contain (either via `\input` or `\include`) the file fredidx.tex.

Chen and Harrison [4] discuss the problems of creating an index in their paper *Index preparation and processing* and also describe the MakeIndex program. Goosens, Mittelbach and Samarin [3] also describe how to use the MakeIndex program for producing indexes for LaTeX documents.

C.1 The index command

The `\index` command is one of the standard LaTeX commands. The command format is `\index{<str>}`, where `<str>` is any string of characters, and it writes an entry to the .idx file in the form `\indexentry{<str>}{pg}`, where `pg` is the page number in the document where the command is called.

Some points to note:

- The `\index` command is, in LaTeX terminology, *fragile*. That is, if it appears in a moving argument (like the caption to a table or figure) it must be preceded by the LaTeX `\protect` command.

- Any of the ten LaTeX special characters (i.e., `#`, `$`, `%`, `&`, `~`, `_`, `^`, `\`, `{` and `}`) may appear within the argument, with the proviso that it must have no unmatched braces (the braces in `\{` and `\}` are counted in the matching process).

- The `\index` command must not appear inside another command's argument (e.g., within the `\footnote` command) unless the argument of the `\index` command contains only letters, digits, and/or punctuation characters. In particular, it should not contain any of the special characters.

  NOTE 1 – This means that the argument of the `\ix` command should not contain any special characters. Remember that the `\ix` command prints its argument in the body of the text and also calls `\index` to place its argument into the .idx file.

NOTE 2 – Under some circumstances, an `\index` command appearing in another command's argument may contain special characters, provided they are `\protect`ed. Determination of when this is satisfactory is a process of trial and error.

EXAMPLE 41 – The command `\ix{an\protect\_underscore}` will print the characters *an_undescore* in the text and also write the following to the .idx file:

```
\indexentry{an\_ underscore}{pg}
```

Notice that there is a space between the underscore character and the word 'underscore' in the .idx file (but there is no space in the printed body of the document text). This extraneous space may have to be edited out from the final index.

C.2 The program GenIndex

GenIndex is a C program that converts .idx data to `theindex` data. Source code for GenIndex is freely available from the NIST SOLIS system. The GenIndex program is not as sophisticated as MakeIndex but does a reasonable job.[4]

C.2.1 Input

GenIndex reads lines of data of the form:

```
\indexentry{IndexData}{Page}
```

[4] These two programs are not completely compatible but do have a common subset of commands. The common subset excludes the `SeeAlso` and `See` commands. At some later time GenIndex may be rewritten to be compatible with MakeIndex. It is probably advisable, therefore, not to use the GenIndex `SeeAlso` and `See` commands.

`Page` is a character string representing a page number. GenIndex only recognizes strings that represent integer values greater than or equal to zero, or (upper and lower case) roman numerals. For any other numbering system, the page number is treated as zero.

`IndexData` is a string of characters and command characters that specify the data that is to be indexed. In the simplest case, this is just the word or phrase to appear in the index, but much more can be done than this, as is described below.

The general form of `IndexData` is given by the following partial grammar:

```
IndexData  = MainData ['!' SubData ['!' SubSubData ] ] .
MainData   = Data .
SubData    = Data .
SubSubData = Data .
Data       = Key [ Entry ] [ SeeAlso ] [ See ] .
Key        = any string not containing the @, !, + or * characters .
Entry      = '@' any string not containing the !, + or * characters
                 (unless enclosed in braces {} ) .
SeeAlso    = '+' any string not containing the ! or * characters
                 (unless enclosed in braces {} ) .
See        = '*' any string not containing the ! character
                 (unless enclosed in braces {} ) .
```

Upto three levels of indexing are enabled — a main topic entry, an optional sub-topic, and an optional sub-subtopic. The sub-topics for an entry are introduced by the `!` character.

Apart from `Key`, braces within a string must be matched. That is, they must appear in pairs of an opening and a closing brace. Spaces are significant within the `Key` string, but not in the others. That is, `\indexentry{A}` differs from `\indexentry{ A}` which differs from `\indexentry{A }`.

EXAMPLES

42 – Here is an index entry for a simple topic:

```
\indexentry{Topic}{27}
```

43 – And here is one where the key and the main entry are different, a sub-topic is being indexed, and the page is in roman numerals:

```
\indexentry{main@{\bf Main}!sub topic}{xviii}
```

44 – This is how the characters `@`, `!`, `+` and `*` characters are indexed in this document.

```
\index{(64@{\verb|@|}}
\index{(33@{\verb|!|}}
\index{(43@{\verb|+|}}
\index{(42@{\verb|*|}}
```

C.2.1.1 Key

The indexed entries are sorted alphabetically on the value of `Key`. Sorting is case-insensitive. A `Key` value is required.

Table C.1 – The LaTeX ASCII character set

	0	1	2	3	4	5	6	7	8	9
30			␣	!	"	#	$	%	&	’
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	‘	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~			

More precisely, the key entries are sorted according to the C language implementation collating sequence, which is usually ASCII. Table C.1 gives the ASCII collating sequence for the LaTeX character set.

C.2.1.2 Entry

`Entry` is optional and is introduced by the `@` character. If present, this will form the indexed string. If absent, then the value of `Key` is used instead.

C.2.1.3 See also

`SeeAlso` is optional and is introduced by the `+` character. Within a document, it should only be used once per entry. If used more than once, then only the last value is taken.

This is used to produce an entry that refers to another indexed entry in a *see also ...* style.

C.2.1.4 See

`See` is optional and is introduced by the `*` character. Within a document, it should be used only once per entry. If used more than once, then only the last value is taken.

This is used to produce an entry that refers to another entry in a *see ...* style. Use of this option prohibits printing page numbers for the entry.

C.2.2 Output

The program GenIndex sorts the entries into alphabetical order (based on the `Key` values), and produces a `theindex` for the input data. Several formatting commands are included in the output file to enable adjustment of the appearance of the printed index. These are shown in table C.2.

NOTES

Table C.2 – GenIndex formatting commands

Command	Effect
`\indexfill`	spacing style between main topic and page numbers
`\sindexfill`	spacing style between subtopic and page numbers
`\ssindexfill`	spacing between sub-subtopic and page numbers
`\indexsee{text}`	produces `text` as a *see* entry
`\indexseealso{text}`	produces `text` as a *see also* entry
`\otherindexspace{c}`	heading for non-alphabetic entry
`\alphaindexspace{c}`	heading for an alphabetic entry

1 – The commands in table C.2 are defined in the iso file.

2 – If the commands are not defined in your system, then you can define them using the LaTeX `\newcommand` command. On the other hand, if they are defined in your system, you can change them using the LaTeX `\renewcommand` command.

EXAMPLE 45 – This input file:

```
\indexentry{Freddy+Fred}{27}
\indexentry{Frederick*Fred}{29}
\indexentry{Fred}{42}
\indexentry{Fred}{52}
\indexentry{Fred}{43}
```

will produce output like:

```
\begin{theindex}
\alphaindexspace{F}
\item Fred \indexfill 42--43, 52
\item Frederick \indexsee{Fred}
\item Freddy \indexfill 27 \indexseealso{Fred}
\end{theindex}
```

C.2.2.1 Indexfill commands

The commands `\indexfill`, `\sindexfill` and `\ssindexfill` control the style of spacing between an indexed entry and its corresponding page numbers.

EXAMPLE 46 – If you want to have the page numbers right justified, with lines between the main topics and their numbers, dots between sub-topics and pages, and sub-subtopic numbers right justified, then you could define these commands as:

```
\newcommand{\indexfill}{\hrulefill}
\newcommand{\sindexfill}{\dotfill}
\newcommand{\ssindexfill}{\hfill}
```

NOTE – The usual LaTeX style is to have a small gap. If this is what you want, then define the commands as:

```
\newcommand{\indexfill}{}
\newcommand{\sindexfill}{}
\newcommand{\ssindexfill}{}
```

C.2.2.2 Indexsee and indexseealso commands

The `\indexsee{text}` and `\indexseealso{text}` commands control the appearance of the *see* and *see also* text. Each of these takes `text` as a parameter.

EXAMPLE 47 – These commands could be defined as:

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

C.2.2.3 Index space commands

The commands `\alphaindexspace{c}` and `\otherindexspace{c}` control the amount of space between blocks of index entries. These correspond to the LaTeX `\indexspace` command, which just inserts some vertical space into the index listing. Both these commands take a single parameter, which is typically a single character.

NOTE – GenIndex looks at the first character (call it `c`) of the `Key` and if this changes as it processes the ordered list of main topics, it puts that character as the parameter for these commands. It writes `\alphaindexspace{c}` if the character is alphabetic (and `c` is put into upper-case), otherwise it writes `\otherindexspace{c}`.

EXAMPLE 48 – These commands could be defined as:

```
\newcommand{\alphaindexspace}[1]{\indexspace
                          {\bf #1}}
\newcommand{\otherindexspace}[1]{}
```

which would result in the printing of a vertical space and a bold font text for an alphabetic header, or nothing for anything else.

C.2.3 Running GenIndex

To run GenIndex, first obtain a copy of the program, and compile it if necessary. Execute the program on your data.

GenIndex takes zero, one or two file names as parameters. If no files are given then input and output is from and to `stdin` and `stdout` respectively.

If one file is given, then input is taken from this file, and output is to `stdout`.

If two file names are given, then input is taken from the first and output is to the second.

EXAMPLE 49 – A typical usage would be:

```
GenIndex fred.idx fredidx.tex
```

which would read the fred.idx file and write the resulting index data to file fredidx.tex.

Annex D
(informative)
The relationship with SGML

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

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

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

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

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

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

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

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

Annex E
(informative)
Obtaining LaTeX and friends

LaTeX is a freely available document typesetting system. There are many public domain additions to the basic system. The information below gives pointers to where you can obtain LaTeX etc., from the Internet. General information on Internet access is obtainable from many sources, for example reference [12].

E.1 CTAN

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

– `ftp.dante.de` in Germany;

– `ftp.shsu.edu` in the USA;

– `ftp.tex.ac.uk` in the UK.

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

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

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

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

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

E.2 SOLIS

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

There are several ways of accessing SOLIS. These are detailed by Ressler [10] and Rinaudot [11]. Copies of these reports may be obtained by telephoning the IPO Office at `+1 (301) 975-3983`. Abbreviated instructions are given below and are current as of September 1994.

E.2.1 Anonymous ftp

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

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

cd pub/step

to get to the top level directory.

E.2.2 Kermit

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

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

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

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

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

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

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

E.2.3 Electronic mail

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

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

E.2.4 Gopher

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

E.2.4.1 Gopher client

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

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

`gopher elib.cme.nist.gov`

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

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

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

E.2.4.2 World Wide Web

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

– `gopher://elib.cme.nist.gov` or

– `http://elib.cme.nist.gov:70`

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

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

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

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

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

Annex F
(informative)

Bibliography

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

[2] RAHTZ, S., and BARROCA, L., *A style option for rotated objects in LaTeX,* TUGBoat, volume 13, number 2, pp 156–180, July 1992.

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

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

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

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

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

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

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

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

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

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

Index