

Sgmltexi

An alternative way to write Texinfo documentation
This edition is for Sgmltexi 2003.00.00 for Texinfo 4

Daniele Giacomini <daniele@swlibero.org>

Copyright © 2000-2003 Daniele Giacomini <daniele@swlibero.org>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Short Contents

Introduction to Sgmltexi	1
1 General structure for a Sgmltexi source file	2
2 Sectioning, nodes and menus	14
3 Block and in-line	17
4 Index and cross reference	18
5 Marking words and phrases	20
6 Marking block of text	21
7 List and tables	23
8 Insertions	25
9 Definitions	26
10 Conditional and literal back-end code	29
11 How to use the front-end	33
12 How to install	35
13 Dependencies	36
14 Encoding	37
15 Supported and unsupported Texinfo feature	38
A Supported ISOnum entities: numeric and special graphic . . .	61
B Supported ISOpub entities: publishing	63
C Supported ISOtech entities: general technical	65
D Supported ISOlat1 entities: added latin 1	67
E Supported ISOlat2 entities: added latin 2	69
GNU Free Documentation License	72
Index	78

Introduction to Sgmltexi

Sgmltexi is a DTD with tools to get Texinfo. The idea is to have another way to write Texinfo documents, intended to be a little bit easier, but also with some limitations.

Sgmltexi manages Texinfo nodes automatically, generating a an Info menu at the Top node, and other menus if required. The node names may be generated automatically with string like "chap 1", "app A" and so on, or can be inserted manually, with different names.

The Sgmltexi document has a precise scheme: there may be one or more introductions, there is a body (made probably of chapters), there may be appendixes and indexes. The body is organized into chapters, that may be grouped into parts and also tomes. This way, also big documents are easily managed.

Sgmltexi is a work derived form ALtools and Alml, from the same author. These are the SGML typesetting systems made for the need of an Italian documentation project:

[Appunti di informatica libera.](#)

Obtain Sgmltexi

At the moment, the main distribution source for Sgmltexi is the following URI:

<http://a2.swlibero.org/~daniele/software/sgmltexi/>

What's new about Sgmltexi

The new Sgmltexi is simplified: the derivation system was removed, and the Sgmltexi source files should be "expanded" to avoid the horizontal tabs presence. The expansion is needed when reproducing literal text, otherwise, strings like ‘\011’ will appear, instead of original tabs.

This modification will reduce the pre-elaboration time, and will avoid also some possible troubles with SGML files included inside the main source.

Please consider that this one might be the last release for Sgmltexi, because the Texinfo development is taking different SGML/XML paths.

1 General structure for a Sgmltexi source file

The typical Sgmltexi source start like this:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN">
```

It can be useful to define also some internal entities, like this:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
<!ENTITY EDITION    "2003.10.11" >
...
...
]>
```

All the document is enclosed inside the element `sgmltexi`. Inside, there must be an `head` element, there may be an `intro` element, there must be a `body` element, and there may be an `appendix` element. The space after the `appendix` element may be occupied by some indexes (will be shown later).

```
<sgmltexi>
<head>
...
</head>
<intro>
...
</intro>
<body>
...
</body>
<appendix>
...
</appendix>
</sgmltexi>
```

The element `sgmltexi` has three possible attribute: `lang`, `charset` and `spacing`.

lang [Attribute]

This is a two letter code defining the text language. The use of this attribute generates a `@documentlanguage` command.

charset [Attribute]

This is the input character set, like it can be done with the Texinfo `@documentencoding` command. It is obscured by the ‘`--input-encoding`’ option, that take precedence and generate a pure ISO 646 Texinfo output.

spacing [Attribute]

This is a deprecated feature that help controlling the spacing after sentences. It is deprecated because this action should be controlled with the language specific configuration. This attribute is here only as a last resort. Valid values are: **normal**, **french** and **uniform**. Selecting **french** or **uniform** it is introduced the command `@frenchspacing`.

```
<sgmltexi lang="it" charset="ISO-8859-1" spacing="uniform">
```

1.1 head

The `head` element is the more complicated. It is important to define many informations about the document. Here it is the example from this manual:

```
<head>
  <admin>
    <setfilename content="sgmltexi.info">
    <setttitle content="Sgmltexi">
    <setchapternewpage content="odd">
    <defindex name="sg">
    <syncodeindex from="sg" to="cp">
    <syncodeindex from="vr" to="cp">
    <infodir cat="Texinfo documentation system">
  </admin>
  <titlepage>
    <title>Sgmltexi</title>
    <subtitle>An alternative way to write Texinfo
documentation</subtitle>
    <subtitle>This edition is for Sgmltexi
&EDITION; for Texinfo 4</subtitle>
    <abstract>
      <p>Sgmltexi is an SGML system (DTD and tools) to
make Texinfo documentation using SGML...</p>
      ...
    </abstract>
    <author>Daniele Giacomini
&lt;daniele@swlibero.org&gt;</author>
    <legal>
      <copyright>Copyright &copy; 2000-2003 Daniele Giacomini
&lt;daniele@swlibero.org&gt;</copyright>
      <publishnote>
        <p>Published by...</p>
      </publishnote>
      <license>
        <p>Permission is granted to make and distribute
verbatim copies of this manual...</p>
        ...
      </license>
      <coverart>
        <p>Cover art by ...</p>
      </coverart>
    </legal>
  </titlepage>
  <shortcontents>
```

```
<contents>
</head>
```

This is not all necessary, but it is a good starting point. Looking at this example, can be recognized some important elements: **admin**, for administrative informations, and **titlepage**.

1.2 admin

The **admin** element is used to enclose some empty elements that describe informations that should go inside the Texinfo header. The order for these elements is not important, as Texinfo source will be build in the right one. It follows a summary table and then the detailed description.

Element	Attribute	Content	Description or Texinfo equivalence
setfilename		empty	@setfilename
	content		Info file name
settitle		empty	@settitle
	content		title
setchapternewpage		empty	@setchapternewpage
	content	on, off, odd	tell how to separate chapters
footnotestyle		empty	@footnotestyle
	content	end, separate	where to place footnotes
headings		empty	@headings
	content	on, off, single, double, singleafter, doubleafter	how headings should work
defindex		empty	@defindex
	name		define a two letter index name
defcodeindex		empty	@defcodeindex
	name		define a two letter index name with @code items
synindex		empty	@synindex
	from		source index, as a two letters name
syncodeindex	to		destination index, as a two letters name
		empty	@syncodeindex
infodir	from		source index, as a two letters name
	to		destination index, with @code items
		empty or literal	Info directory information
		@direntry	
	cat		The @dircategory argument

setfilename

[Element]

This element is empty and is used to define the file name for Info, with the Texinfo command **@setfilename**.

content [Attribute]

This attribute is used to assign the file name.

Use this element like this, only with the opening tag:

```
<setfilename content="sgmltexi.info">
```

settitle [Element]

This element is empty and is used to define the title for Info, with the Texinfo command `@settitle`.

content [Attribute]

This attribute is used to assign the title.

Use this element like this, only with the opening tag:

```
<settitle content="Sgmltexi">
```

setchapternewpage [Element]

This element is empty and is used to define the Texinfo command `@setchapternewpage`.

content [Attribute]

This attribute is used to assign the desired value, that can be: **on**, **off** or **odd**.

Use this element like this, only with the opening tag:

```
<setchapternewpage content="on">
```

This element is not essential.

footnotestyle [Element]

This element is empty and is used to define the Texinfo command `@footnotestyle`.

content [Attribute]

This attribute is used to assign the desired value, that can be: **end** or **separate**.

Use this element like this, only with the opening tag:

```
<footnotestyle content="end">
```

This element is not essential.

heading [Element]

This element is empty and is used to define the Texinfo command `@headings`.

content [Attribute]

This attribute is used to assign the desired value, that can be: **on**, **off**, **single**, **double**, **singleafter**, **doubleafter**.

Use this element like this, only with the opening tag:

```
<headings content="double">
```

This element is not essential.

defindex [Element]
defcodeindex [Element]

These two empty elements are used to define the Texinfo commands `@defindex` and `@defcodeindex`.

name [Attribute]
 This attribute is used to define the name of the new user index that is to be created. The name must be a two letter word.

Use these elements like this, only with the opening tag:

```
<defindex name="sg">
<defcodeindex name="xx">
```

These elements are used only as needed, for as many user defined index that are to be created.

synindex [Element]
syncodeindex [Element]

These two elements are used to copy one index into another, like the command `@synindex` and `@syncodeindex` do with Texinfo.

from [Attribute]
to [Attribute]
 The first attribute is used to define the index to copy; the second is the index that receive the first one.

Use this element like this, only with the opening tag:

```
<syncodeindex from="fn" to="cp">
```

This element is used only as needed, for as many user defined index that are to be created.

infodir [Element]
 This element is used to define the Info directory menu item, when the file is installed with `install-info`.

cat [Attribute]
 This attribute defines the category, like the Texinfo command `@dircategory` does.

This element can be used empty, like this:

```
<infodir cat="Texinfo documentation system">
```

But this element can be used also more explicitly, like this:

```
<infodir cat="Texinfo documentation system">
* Sgmltexi: (sgmltexi).          SGML for Texinfo.
* Sgmltexi install: (sgmltexi)install.  Install Sgmltexi.
</infodir>
```

This element is used only if needed. If the element is use empty, only one line inside the `@direntry` environment is inserted, with information already given.

1.3 titlepage

The `titlepage` element is used to enclose all informations that go on the first pages of the document. The order of elements is important. It follows a summary table and then the detailed description.

Element	Attribute	Content	Description or Texinfo equivalence
<code>title</code>		<i>in-line</i>	<code>@title</code>
<code>subtitle</code>		<i>in-line</i>	<code>@subtitle</code>
<code>abstract</code>		<i>block</i>	abstract of the document
<code>author</code>		<i>in-line</i>	<code>@author</code>
<code>frontcovertext</code>		<i>block</i>	text shown on the printed front cover
<code>tpextra</code>		<i>block</i>	extra text shown inside title pages
<code>legal</code>		copyright, publishnote, license, coverart <i>block</i>	legal information
<code>legal</code>		<i>block</i>	simplified legal information
<code>copyright</code>			One copyright owner
<code>publishnote</code>		<i>block</i>	publishing note that should appear before the license
<code>license</code>		<i>block</i>	license or reference to the license, or other conditions related to the document
<code>coverart</code>		<i>block</i>	coverart note that should appear after the license
<code>dedications</code>		<i>block</i>	dedications for a book

title [Element]

This element contains the document title; the title for the "printed" document. It is equivalent to `@title` for Texinfo. Use this element like this:

```
<title>Sgmltexi</title>
```

subtitle [Element]

This element can be used to define one or more subtitles. It is equivalent to `@subtitle` for Texinfo. Use this element like this, after the title:

```
<subtitle>An alternative way to write Texinfo documentation</subtitle>
```

This element is used only as needed, without limitations.

abstract [Element]

This element can be used to define a brief description for the document. The content must be block text, like the element `p`. This text is used for Info typesetting, inside the `Top` node. Use this element like this:

```
<abstract>
```

```
<p>Sgmltexi is an SGML system (DTD and tools) to
make Texinfo documentation using SGML.
The Sgmltexi DTD is designed to...</p>
```

```
<p>...</p>
```

```
</abstract>
```

This element can be used at most one time.

author [Element]

This element can be used to define one of the document authors. This element is equivalent to the Texinfo command `@author`. Use this element like this:

```
<author>Tizio Tizi &lt;tizio@dinkel.brot.dg></author>
```

```
<author>Caio Cai &lt;caio@dinkel.brot.dg></author>
```

This element must be used at least one time.

frontcovertext [Element]

This element can be used to include one or more block of text, that should appear on the front cover, for the printed edition. Use this element like this:

```
<frontcovertext>
```

```
<p>Version 1.2.3</p>
```

```
<p><image name="good-thing" height="5cm"></p>
```

```
</frontcovertext>
```

tpextra [Element]

This element can be used to include one or more block of text, that should appear on different places: before the legal informations; after legal information; after the dedications. The name `tpextra` stand for "title page extra" text. Use this element like this:

```
<tpextra>
```

```
<p><strong>Pinco Pallino</strong> is a very old man...</p>
```

```
<p><strong>Tizio Tizi</strong> studied telecommunication
technology...</p>
```

```
</tpextra>
```

```
<legal>
```

```
...
```

```
...
```

```
</legal>
```

```
<tpextra>
```

```
<p>The front cover picture is made by Sempronio.</p>
```

```
</tpextra>
```

```

<dedications>
    ...
    ...
</dedications>
<tpextra>

```

```

    <p>The software is very important; true free software is much more
    important.</p>

```

```

</tpextra>

```

The use of this element is optional, but notice that the last one can be used only if there is the dedications element before.

legal [Element]

This element is used to enclose other elements describing informations about copyright, license and publication.

copyright [Element]

This element contains a line of text describing the copyright owner for the document.

publishnote [Element]

This optional element, contains block of text that show information about the printed publication.

license [Element]

This element contains block of text to show the license conditions for the document.

coverart [Element]

This optional element, contains block of text that show who is responsible for the cover design.

Use this element like this:

```

<legal>
  <copyright>Copyright &copy; 2003 Free Software Foundation,
  Inc.</copyright>
  <publishnote>
    <p>Published by...</p>
  </publishnote>
  <license>
    <p>Permission is granted to copy, distribute and/or
    modify this document under the terms of the GNU Free
    Documentation License, Version 1.1 or any later version
    published by the Free Software Foundation; with no
    Invariant Sections, with no Front-Cover Texts, and with

```

```

        no Back-Cover Texts. A copy of the license is included
        in the section entitled "GNU Free Documentation
        License".</p>
    </license>
    <coverart>
        <p>Cover art by ...</p>
    </coverart>
</legal>

```

This element must be used at least one time.

Actually, this element can have non special internal structure, enclosing simply other block elements. This way, if the standard structure described above is not good for the author intention, it may be used also like this:

```

<legal>
    <p>Copyright &copy; 2003 Free Software Foundation, Inc.</p>

    <p>Published by...</p>

    <p>Permission is granted to copy, distribute and/or modify this
    document under the terms of the GNU Free Documentation License,
    Version 1.1 or any later version published by the Free Software
    Foundation; with no Invariant Sections, with no Front-Cover Texts,
    and with no Back-Cover Texts. A copy of the license is included in
    the section entitled "GNU Free Documentation License".</p>

    <p>Cover art by ...</p>
</legal>

```

dedications [Element]

This optional element, contains block of text that show one or more dedications.

1.4 Table of contents

After the **titlepage**, there is the place for one or more table of contents. To obtain the insertion of these table of contents can be used one of the following empty element.

contents [Element]

The standard table of contents, like the command **@contents** for Texinfo.

shortcontents [Element]

summarycontents [Element]

A reduced table of contents, like the command **@shortcontents** and **@summarycontents** for Texinfo.

1.5 menu

After the contents elements, may appear the `menu` element. This element ask explicitly for the correspondent Texinfo `@menu` command. At this level, the menu is inserted automatically, also without inserting this element. But the `menu` element may be used to define a specific (manual) menu for texinfo. See the following example:

```
<menu>
* introduction ::   Introduction
* structure::      General structure for a Sgmltexi source file
* how to use::     How to use the front-end
* install::        How to install
* GFDL::           GNU Free Documentation License
* index::          Index
</menu>
```

The `menu` may be used also at chapter level and below, and there can be also empty: in this case, an automatic Texinfo menu will be inserted.

1.6 intro

After the `head` element there may be one `intro` element. This is just a way to define a group of chapter that have no numbering. Chapters inside this element are delimited in the same way as for the `body` and `appendix` elements.

```
<intro>
<h1>Introduction to Sgmltexi</h1>

<p>Sgmltexi is a DTD with tools to get Texinfo...</p>

<p>Sgmltexi manage Texinfo nodes automatically,...</p>

</intro>
```

1.7 body

After the `head` and `intro` elements, there must be one `body` element. This is the body of the document.

The body may be divided into chapters, or parts, or tomes, depending on the documentation project that is started. Tomes are delimited with the element `tomeheading`, that contains the tome title; parts are delimited using the element `partheading`, with the same meaning.

Chapters and lower sectioning are a little anonymous, like HTML: `h1`, `h2`, `h3` and `h4`. This is done that way because introduction, body and appendix may have the same method for text sectioning.

```
<body>
<partheader>Networking</partheader>
```

```

<h1>IP protocol history</h1>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h2>ISO/OSI model</h2>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h1>IPv4 and IPv6</h1>

<p>Bla bla bla...</p>

<p>...</p>

</body>

```

Every sectioning element, from tome to last sub-subsection, have an optional attribute: `id`. This attribute may be used to define an identification string that can be the target for cross references.

```

<h1 id="ip history">IP protocol history</h1>

```

Note that due to Texinfo design limitations, cross references labels cannot contain the comma.

1.8 appendix

After the body, it may be placed the **appendix**. This one is used to delimit a group of chapters that must be intended as appendixes. This way, also the appendix is organized in chapter delimited with `h1` heading, and smaller sections.

1.9 Indexes

After the body and after the optional appendix, can be placed one or more analytic index. This is obtained with a special heading element: `indexheading`. This can be used only in conjunction with `printindex` to define the type of index to include. The example should be clear enough:

```

...
<indexheading>Index of functions</indexheading>
<printindex name="fn">
<indexheading>Concept index</indexheading>

```

```
<printindex name="cp">  
...  
</sgmltexi>
```

As can be seen from the example, `printindex` has an attribute, `name`, that tells the type of index to include. In fact, this way is generated the Texinfo command `@printindex`.

2 Sectioning, nodes and menus

To write good documentation with Texinfo, it is required to have control on nodes and menus. With Sgmltexi, nodes and menus can be forgotten, but the Info result may suffer for this choice. Anyway, with Sgmltexi it is possible to choose different levels of manual/automatic node handling.

Headings elements can incorporate some more attributes: **node** and **menu**. The first one define the node name, overriding any automatic determination; the second define the node description on the menu (the automatical choice is otherwise the title).

```
<h1 id="ip history" node="history" menu="History of IP protocol">
IP protocol history</h1>
```

This will generate on the Info menu, the following line:

```
* history::          History of IP protocol
```

The **node** and **menu** attribute may be used independently: the attribute that is not used, will be determined automatically.

Having access to nodes, it is possible to use them for cross reference, without the need for the **id** attribute.

Sgmltexi creates automatically the Top node menu. As already explained before (see [\[top node menu\]](#), page 11), the menu can be explicitly defined. In this way, all nodes and descriptions must be made manually.

Inserting the **menu** element at the bottom of a chapter, or at a lower section, will ask an insertion of a lower Info menu. See the example:

```
<h1>IP protocol history</h1>
```

```
<p>Bla bla bla...</p>
```

```
<p>Bla bla bla...</p>
```

```
<menu>
```

```
<h2>ISO/OSI model</h2>
```

```
<p>Bla bla bla...</p>
```

```
<p>Bla bla bla...</p>
```

```
<h2>More information</h2>
```

```
<p>Bla bla bla...</p>
```

```
<p>...</p>
```

The example shows the insertion of an automatic Info menu before **h2** sections. This menu may otherwise be described completely, like this:

```
<menu>
```

```
* IP layer::      IP ISO/OSI layer model
* more on IP::    More details on IP
</menu>
```

When the menu is described this way, node names must be the same as the one used with the heading elements. That is: when writing the menu, also nodes must be exactly declared, like this:

```
<h1>IP protocol history</h1>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<menu>
* IP layer::      IP ISO/OSI layer model
* more on IP::    More details on IP
</menu>

<h2 node="IP layer">ISO/OSI model</h2>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h2 node="more on IP">More information</h2>

<p>Bla bla bla...</p>

<p>...</p>
```

It is evident that a `menu` attribute (like `<h2 menu="Much more information">`) has no effect in this case.

Element	Attribute	Content	Description or Texinfo equivalence
tomeheading		<i>in-line</i>	title of a tome
parthead			title of a part
h1		<i>in-line</i>	title of a chapter, introduction or appendix
h2		<i>in-line</i>	title of a section
h3		<i>in-line</i>	title of a subsection
h4		<i>in-line</i>	title of a sub-subsection
indexheading		<i>in-line</i>	title of an analytic index
	node		Info node name
	menu		Info node description
	next		Next node name
	prev		Previous node name
	up		Up node name
	type	numbered, unnumbered, heading	type of section, for h1 to h4 elements

2.1 Unnumbered sections and simple headings

Elements `h1`, `h2`, `h3` and `h4` may have an additional attribute: `type`. The keywords `numbered`, `unnumbered` and `heading` may be assigned. `numbered` is the default value; it means that the title will be numbered (with numbers if inside `body`, with letters if inside `appendix`). Assigning `unnumbered`, the titles are not numbered. Assigning `heading`, the titles are not numbered and not annotated inside the table of context.

```
<h1 type="unnumbered">Acknowledgements</h1>
```

...

Inside the `intro` elements, titles cannot be numbered: it is explicitly excluded.

3 Block and in-line

Sgmltexi has a DTD where most of the elements are divided into two categories, block and in-line, with the help of two parameter entities: **block** and **inline** (SGML macro are ‘%block;’ and ‘%inline;’).

A block is something like a paragraph, a list, a table; an in-line is text, text emphasisation, anchors, cross references, and other things that stay inside a text.

Usually, but not necessarily, an in-line element contains text and possibly other in-line elements; but a block element may be made to contain in-line or other blocks. The Sgmltexi DTD don't consider the possibility of block elements that may contain either block or in-line. These kinds of contents are known as "flow" (this name is used inside the HTML DTD) and are rarely useful.

Some block elements, like **example**, may contain block elements or a single **pre** element (a special block element not classified as part of the ‘%block;’ macro). The **pre** element can contain only in-line that is preformatted, that is: it maintains line breaks.

The two basic block element are shown in the following table:

Element	Attribute	Content	Description or Texinfo equivalence
p		<i>in-line</i>	paragraph, or simple block of text
	indent	on , off	first line indentation; default is on
center		<i>in-line</i>	@center

4 Index and cross reference

There are different kind of insertions for making indexes and cross references, that reproduce equivalent Texinfo commands.

4.1 Indexes

Index entries are inserted with a group of empty elements: `cindex`, `findex`, `vindex`, `kindex`, `pindex`, `tindex` and `userindex`. All these elements have the same attribute, `entry`, that define the item text to be inserted. The `userindex` has an additional attribute, to define the user index name (that should be made of two letters).

These elements are a kind of block that may be inserted just after any sectioning title, like this:

```
<h1>IP protocol history</h1>
<cindex entry="IP protocol">
<cindex entry="history">

<p>Bla bla bla...</p>
```

The following table resumes the meaning for so many different index entry elements.

Element	Attribute	Content	Description or Texinfo equivalence
cindex	entry	empty	concept index item
findex	entry	empty	function index item
vindex	entry	empty	variable index item
kindex	entry	empty	keystroke index item
pindex	entry	empty	program index item
tindex	entry	empty	data type index item
userindex	entry	empty	user defined index item
	name		user defined index name (two letters)
printindex	name		print the named index (two letters)

The index is inserted with the element `printindex`, already described. Standard index names are listed in the following table.

Index name	Description
cp	concept index.
ky	keystroke index.
pg	program index.
fn	function index.
vr	variable index.
tp	data type index.

4.2 Cross references

Cross reference elements are all in-line empty elements. All information is given via attributes. As all cross reference elements are implementations of equivalent Texinfo commands, there is only the following table as description. Every attribute description follow its own element. Please note that not all attributes are necessary.

Element	Attribute	Content	Description or Texinfo equivalence
anchor		empty	@anchor
xref	id	empty	anchor identity string
	name		node or anchor name
	title		cross reference name
	info		title or topic
	ptitle		info file name
ref	ptitle	empty	printed manual title
			@ref
	id		node or anchor name
	name		cross reference name
	title		title or topic
pxref	info	empty	info file name
	ptitle		printed manual title
			@pxref
	id		node or anchor name
	name		cross reference name
inforef	title	empty	title or topic
	info		info file name
	ptitle		printed manual title
			@inforef
	id		node or anchor name
uref	name	empty	cross reference name
	info		info file name
			@uref
	uri		URI address
	name		title or description
email	replace	empty	replacement text
			@email
	email		electronic mail address
	name		title or description

Use like this:

```
<p>Sgmltexi creates automatically the Top node menu. As already
explained before (<pxref id="top node menu">), the menu can be
...
```

5 Marking words and phrases

A lot of in-line elements are used to mark words or phrases. Actually, the DTD is very permissive, so that every element of these can contain any in-line element. It is so only to assure compatibility with Texinfo, but may change in the future. The following table list these elements, included `kbdinputstyle`, used to select the style for the `kbd` element.

Element	Attribute	Content	Description or Texinfo equivalence
<code>code</code>		<i>in-line</i>	<code>@code</code>
<code>kbd</code>		<i>in-line</i>	<code>@kbd</code>
<code>kbdinputstyle</code>		empty	<code>@kbdinputstyle</code>
	<code>style</code>	<code>code</code> , <code>example</code> , <code>distinct</code>	how to show keyboard input style; default is <code>distinct</code>
<code>key</code>		<i>in-line</i>	<code>@key</code>
<code>samp</code>		<i>in-line</i>	<code>@samp</code>
<code>var</code>		<i>in-line</i>	<code>@var</code>
<code>env</code>		<i>in-line</i>	<code>@env</code>
<code>file</code>		<i>in-line</i>	<code>@file</code>
<code>command</code>		<i>in-line</i>	<code>@command</code>
<code>option</code>		<i>in-line</i>	<code>@option</code>
<code>dfn</code>		<i>in-line</i>	<code>@dfn</code>
<code>cite</code>		<i>in-line</i>	<code>@cite</code>
<code>acronym</code>		<i>in-line</i>	<code>@acronym</code>
<code>url</code>		<i>in-line</i>	<code>@url</code>
<code>emph</code>		<i>in-line</i>	<code>@emph</code>
<code>strong</code>		<i>in-line</i>	<code>@strong</code>
<code>sc</code>		<i>in-line</i>	<code>@sc</code>
<code>roman</code>		<i>in-line</i>	<code>@r</code>
<code>italic</code>		<i>in-line</i>	<code>@i</code>
<code>bold</code>		<i>in-line</i>	<code>@b</code>
<code>typewriter</code>		<i>in-line</i>	<code>@t</code>

Use like this:

```
<p><strong>Pinco Pallino</strong> is a very old man...</p>
```

```
<p><strong>Tizio Tizi</strong> studied telecommunication  
technology...</p>
```

6 Marking block of text

Some block elements are used to mark other block of text or special kind of in-line text. The DTD is very permissive, to assure maximum compatibility with Texinfo, but this may change in the future. The following table list these elements, included the special element **pre**, used to insert in-line preformatted text, and **exdent**, used inside **pre** to obtain an exdented line.

Element	Attribute	Content	Description or Texinfo equivalence
exdent		<i>in-line</i>	@exdent
pre		<i>in-line</i>	preformatted text
quotation		<i>block</i>	@quotation
display		<i>block or pre</i>	@display
smalldisplay		<i>block or pre</i>	@smalldisplay
example		<i>block or pre</i>	@example
smallexample		<i>block or pre</i>	@smallexample
flushleft		<i>in-line</i>	@flushleft
flushright		<i>in-line</i>	@flushright
lisp		<i>block or pre</i>	@lisp
smalllisp		<i>block or pre</i>	@smalllisp
cartouche		<i>block or pre</i>	@cartouche
format		<i>block or pre</i>	@format
smallformat		<i>block or pre</i>	@smallformat
texinfo		literal Texinfo	embedded literal Texinfo code

Use like this:

```
<example>
<p>Hello everybody</p>
<p>Hello to the world</p>
</example>
```

In this case the **example** element contains preformatted text (please not the use of two SGML entities, **lt** and **gt**):

```
<example>
<pre>
#!/usr/bin/perl
while ($line = &lt;STDIN&gt;)
{
    chomp $line;
    print ("&lt;$line&r&n");
}
</pre>
</example>
```

If it is necessary to include a preformatted literal text, do like this (please note that there is no need to hide ‘<’ and ‘>’):

```
<example>
<pre>
<![CDATA[
```

```
#!/usr/bin/perl
while ($line = <STDIN>)
{
    chomp $line;
    print ("$line\r\n");
}
]]>
</pre>
</example>
```

7 List and tables

List and tables are block elements. The following table list the elements used to implement list and tables.

Element	Attribute	Content	Description or Texinfo equivalence
itemize		<i>item</i> , <i>itemx</i> , <i>block</i>	<code>@itemize</code>
	mark		mark before items; default is ‘•’
enumerate		<i>item</i> , <i>itemx</i> , <i>block</i>	<code>@enumerate</code>
	start		starting number
table		<i>item</i> , <i>itemx</i> , <i>block</i>	<code>@table</code>
	emphasis	<i>asis</i> , <i>code</i> , <i>samp</i> , <i>var</i> , <i>kbd</i> ,	emphasis for the descriptive item
vtable		<i>file</i> <i>item</i> , <i>itemx</i> , <i>block</i>	<code>@vtable</code>
	emphasis	<i>asis</i> , <i>code</i> , <i>samp</i> , <i>var</i> , <i>kbd</i>	emphasis for the descriptive variable item
ftable		<i>item</i> , <i>itemx</i> , <i>block</i>	<code>@ftable</code>
	emphasis	<i>asis</i> , <i>code</i> , <i>samp</i> , <i>var</i> , <i>kbd</i>	emphasis for the descriptive function item
item		<i>in-line</i> or	<code>@item</code>
itemx		empty <i>in-line</i> or	<code>@itemx</code>
multitable		empty <i>columnfraction</i> or <i>columnexample</i> ;	<code>@multitable</code>
columnfraction		<i>raw</i> <i>.n</i>	column fraction, where ". <i>n</i> " is the 0. <i>n</i> fraction of the horizontal space
columnexample		pure text	column fraction, where the inserted text gives the example of the column extension
raw		<i>in-line</i> , <i>tab</i>	row of the table
tab		empty	column separation

It follows some examples; first an unnumbered list:

```
<itemize mark="#">
<item>

    <p>First item.</p>

<item>

    <p>Second item.</p>

</itemize>
```

Here is the result:

First item.

Second item.

Here is a numbered list:

```
<enumerate start="3">  
<item>
```

```
<p>First item.</p>
```

```
<item>
```

```
<p>Second item.</p>
```

```
</enumerate>
```

Here is the result:

3. First item.

4. Second item.

Here is a descriptive list:

```
<table emphasis="code">  
<item>ls</item>  
<itemx>dir</itemx>  
<p>List directory contents.</p>  
<item>cd</item>  
<p>Change directory.</p>  
</table>
```

Here is the result:

ls	
dir	List directory contents.
cd	Change directory.

8 Insertions

Here are described some mixed elements that have not found a better place. There are two types of such insertions: in-line and block insertions. The first of the following tables list the in-line elements and the second the block elements.

Element	Attribute	Content	Description or Texinfo equivalence
dmn		parsed character data	@dmn
math		parsed character data	@math
footnote		<i>in-line</i>	@footnote
image		empty	@image
	name		file name to be inserted (without extension)
	width		image width
	height		image height
whole		<i>in-line</i>	@w (preventing line break)
br		empty	@* (line break)
dh		empty	@- (discretionary hyphen)
hyphenation		empty	@hyphenation
	words		list of hy-phen-a-ted words
Element	Attribute	Content	Description or Texinfo equivalence
sp		empty	@sp
	lines		skip <i>n</i> lines
page		empty	@page
group		<i>block</i>	@group
need		empty	@need
	mils		thousandths of inch

9 Definitions

Here are described the elements implementing Texinfo definitions.

Element	Attribute	Content	Description or Texinfo equivalence
args		<i>in-line</i>	argument of a definition
deffn		args, <i>block</i>	@deffn
	cat		function category
	name		function name
deffnx		empty	@deffnx , attributes like deffn
defun		args, <i>block</i>	@defun
	name		function name
defunx		empty	@defunx , attributes like defun
defmac		args, <i>block</i>	@defmac
	name		macro name
defmacx		empty	@defmacx , attributes like defmac
defspec		args, <i>block</i>	@defspec
	name		special form name
defspecx		empty	@defspecx , attributes like defspec
defvr		<i>block</i>	@defvr
	cat		variable category
	name		variable name
defvrnx		empty	@defvrnx , attributes like defvr
defvar		<i>block</i>	@defvar
	name		variable name
defvarx		empty	@defvarx , attributes like defvar
defopt		<i>block</i>	@defopt
	name		option name
defoptx		empty	@defoptx , attributes like defopt
deftypefn		args, <i>block</i>	@deftypefn
	cat		category
	type		data type
	name		name
deftypefnx		empty	@deftypefnx , attributes like deftypefn
deftypefun		args, <i>block</i>	@deftypefun
	type		data type
	name		name
deftypefunx		empty	@deftypefunx , attributes like deftypefun
deftypevr		<i>block</i>	@deftypevr
	cat		category
	type		data type
	name		name
deftypevrnx		empty	@deftypevrnx , attributes like deftypevr
deftypevar		<i>block</i>	@deftypevar
	data type		type
	name		name
deftypevarx		empty	@deftypevarx , attributes like deftypevar
defcv		<i>block</i>	@defcv

	cat		category	
	class		class	
	name		name	
defcvx		empty	@defcvx, attributes like defcv	
defivar		block	@defivar	
	class		class	
	name		name	
defivarx		empty	@defivarx, attributes like defivar	
deftypeivar		block	@deftypeivar	
	class		class	
	type		type	
	name		name	
deftypeivarx		empty	@deftypeivarx, attributes like	
			deftypeivar	
defop		args, block	@defop	
	cat		category	
	class		class	
	name		name	
defopx		empty	@defopx, attributes like defop	
deftypeop		args, block	@deftypeop	
	cat		category	
	class		class	
	type		type	
	name		name	
deftypeopx		empty	@deftypeopx, attributes like deftypeop	
defmethod		args, block	@defmethod	
	class		class	
	name		name	
defmethodx		empty	@defmethodx, attributes like defmethod	
deftypemethod		args, block	@deftypemethod	
	class		class	
	type		type	
	name		name	
deftypemethodx		empty	@deftypemethodx, attributes like	
			deftypemethod	
deftp		args, block	@deftp	
	cat		category	
	name		name	
deftpx		empty	@deftpx, attributes like deftp	

It follows an example about the use of `defn`:

```
<defn cat="Command" name="sgmltexi">
  <args>[<var>options</var>]... <var>sgml_source</var></args>

  <p>This is the front-end for the SGML to Texinfo system.</p>

</defn>
```

Here is the result:

sgmltexi [*options*]... *sgml_source*

[Command]

This is the front-end for the SGML to Texinfo system.

10 Conditional and literal back-end code

Texinfo has a special handling capability to select code depending on the final typesetting. Things like these can be done with SGML using marked sections, but it is not exactly the same thing. Sgmltexi includes some special elements to implement the conditional code insertion used inside Texinfo, and allow to insert also some piece of literal code.

Element	Attribute	Content	Description or Texinfo equivalence
ifinfo		<i>in-line</i>	@ifinfo ... @end ifinfo
ifinfoblock		<i>block</i>	@ifinfo ... @end ifinfo
iftex		<i>in-line</i>	@iftex ... @end iftex
iftexblock		<i>block</i>	@iftex ... @end iftex
ifhtml		<i>in-line</i>	@ifhtml ... @end ifhtml
ifhtmlblock		<i>block</i>	@ifhtml ... @end ifhtml
ifplaintext		<i>in-line</i>	@ifplaintext ... @end ifplaintext
ifplaintextblock		<i>block</i>	@ifplaintext ... @end ifplaintext
ifxml		<i>in-line</i>	@ifxml ... @end ifxml
ifxmlblock		<i>block</i>	@ifxml ... @end ifxml
ifnotinfo		<i>in-line</i>	@ifnotinfo ... @end ifnotinfo
ifnotinfoblock		<i>block</i>	@ifnotinfo ... @end ifnotinfo
ifnottex		<i>in-line</i>	@ifnottex ... @end ifnottex
ifnottexblock		<i>block</i>	@ifnottex ... @end ifnottex
ifnohtml		<i>in-line</i>	@ifnohtml ... @end ifnohtml
ifnohtmlblock		<i>block</i>	@ifnohtml ... @end ifnohtml
ifnotplaintext		<i>in-line</i>	@ifnotplaintext ... @end ifnotplaintext
ifnotplaintextblock		<i>block</i>	@ifnotplaintext ... @end ifnotplaintext
ifnotxml		<i>in-line</i>	@ifnotxml ... @end ifnotxml
ifnotxmlblock		<i>block</i>	@ifnotxml ... @end ifnotxml
tex		parsed character data	@tex ... @end tex
html		parsed character data	@html ... @end html
texinfo		parsed character data	Raw Texinfo code.

Please note that `ifinfo`, `iftex`, `ifhtml`, `ifplaintext`, `ifxml`, `ifnotinfo`, `ifnottex`, `ifnohtml`, `ifnotplaintext` and `ifnotxml`, are in-line elements that contain in-line text. On the other side, `ifinfoblock`, `iftexblock`, `ifhtmlblock`, `ifplaintextblock`, `ifxmlblock`, `ifnotinfoblock`, `ifnottexblock`, `ifnohtmlblock`, `ifnotplaintextblock` and `ifnotxmlblock`, are block elements that contain block text.

The distinction between `if...` and `if...block` is necessary to avoid trouble with the SGML document type declaration.

The in-line elements `tex`, `html` and `texinfo`, are made to contain in-line literal text, usually enclosed inside ‘<![CDATA[’ and ‘]]>’.

The element `texinfo` has no equivalence inside Texinfo, as it represent just raw Texinfo code. See the following example:

```
<p>The letter <texinfo>@ubaraccent{o}</texinfo> is a special...</p>
```

It may be necessary to force literal reading, as for SGML. In this case, the element content can be enclosed like this:

```
<p>The letter <texinfo><![CDATA[@ubaraccent{o}]]></texinfo> is a...
```

The example doesn't show any real reason to put the text inside a CDATA environment, but there are other situations where it will be necessary.

It follows another example on the use of literal back-end code. The intention is to show a simple mathematical expression: $123 + 10^{-1}$.

```
<p><tex><![CDATA[$123+10^{-1}$]]></tex>
<html><![CDATA[123+10<sup>-1</sup>]]></html>
<ifinfo>123+10^-1</ifinfo>
= 12.3</p>
```

As current typesetting is made for TeX the result is as it follows:

$$123 + 10^{-1} = 12.3$$

10.1 Troubles

Texinfo, like TeX and *roff, distinguishes blocks because these are separated from one or more lines. This way, the distinction between blocks and in-lines is only a matter of vertical space. For example, the following peace of Texinfo source shows three environments `@iftypesetting`, which are part of the same block of text (the same paragraph). That is: in-line.

```
Current typesetting is
@iftex
TeX
@end iftex
@ifhtml
HTML
@end ifhtml
@ifinfo
Info
@end ifinfo
and you can see that...
```

But in a different situation, these environments can became isolated block of text, like this:

```
Current typesetting is:

@iftex
TeX
@end iftex

@ifhtml
HTML
```

```
@end ifhtml
```

```
@ifinfo
Info
@end ifinfo
```

You can see that...

With SGML this is not desirable, and difficult to implement. This is the reason why Sgmltexi distinguishes from `@iftypesetting` or `@ifnottypesetting`, and `@iftypesettingblock` or `@ifnottypesettingblock`.

Sgmltexi try to preserve line breaks inside the SGML source, but in this case, there is a consequence on the use of such conditional in-line environments. This happens because Texinfo requires these commands to lay alone on a single line. That is, if we would like to write something like this:

```
<p>Current typesetting
is <iftex>TeX</iftex><ifhtml>HTML</ifhtml><ifinfo>Info</ifinfo>, so
you know what I mean.</p>
```

we expect that opening and closing tags will introduce proper line breaks. But if it were so, the result would be the following, transforming original in-line into undesired final block:

```
Current typesetting is
@iftex
TeX
@end iftex

@ifhtml
HTML
@end ifhtml

@ifinfo
Info
@end ifinfo
, so you know what I mean.
```

To resolve this problem, these in-line conditional tags don't introduce any initial and final line break: it is up to the author to think about this problem. This way, Sgmltexi must be used in the following way, considering that there is no way to put the comma after the typesetting name:

```
<p>Current typesetting is
<iftex>TeX</iftex>
<ifhtml>HTML</ifhtml>
<ifinfo>Info</ifinfo>
so you know what I mean.</p>
```

The same problem happens with `tex` and `html` elements, but in this case there is no need to distinguish the content, that is meant to be always in-line.

```
<p>
<tex>
```

```


$$\chi^2 = \sum_{i=1}^N \frac{\left( y_i - (a + b x_i) \right)^2}{\sigma_i^2}$$


```

$$\chi^2 = \sum_{i=1}^N \left(\frac{y_i - (a + bx_i)}{\sigma_i} \right)^2$$

When using SGML, inserting literal back-end code is just a last resort that breaks standardization. In other words, if it is necessary to use such expedient, maybe the SGML is the wrong choice for writing documentation.

11 How to use the front-end

When the Sgmltexi source is ready, all that is needed is the program `sgmltexi` to control the SGML parser and other elaborations.

sgmltexi [*options*]... *sgml_source* [Command]

This is the front-end for the SGML to Texinfo system.

`--help` Display a brief help and exit.

`--version` Display version information and exit.

`--verbose` Display processing information.

`--force` Force things.

`--number-sections` Number sections for Info and HTML typesetting.

`--clean` Remove stale files associated with the source name.

`--input-encoding=encoding` Define an input encoding that will be converted automatically into entities. This option take precedence over the attribute `charset` of the `sgmltexi` element. Using this option the Texinfo source will be pure ASCII, and the command `@documentencoding` is used with the argument `'ISO-646'`.
Available encodings are: `'ISO-8859-1'`, `'ISO-8859-2'`, ... `'ISO-8859-10'`. All encodings are not fully supported, because of Texinfo limitations. Unsupported codes are shown with the name of the corresponding standard entity, like this: `'[ETH]'`.

`--sgml-include=parameter-entity` Assign the word `INCLUDE` to an SGML parameter entity.

`--include=parameter-entity` Assign the word `INCLUDE` to an SGML parameter entity.

`--paper=paper` Define the paper format: `letter`, `a4`, `a4wide`, `a4latex`, `small`.

`--setchapternewpage={on|off|odd}` Define the chapter paper feed, overriding the tag `<setchapternewpage content="...">` inside the source.

`--footnotestyle={end|separate}` Define the footnote location, overriding the tag `<footnotestyle content="...">` inside the source.

`--sgml-syntax` Check SGML syntax and report errors.

`--sgml-check` Check SGML syntax and report errors.

`--sp` SP output for debugging purpose.

```
'--texi'  
'--texinfo'      Generate a Texinfo source output.  
  
'--dvi'          Generate a DVI output.  
  
'--ps'  
'--postscript'   Generate a PostScript output.  
  
'--pdf'          Generate a PDF output.  
  
'--info'         Generate an Info output.  
  
'--text'         Generate a text output.  
  
'--html'         Generate an HTML output.  
  
'--xml'          Generate a Texinfo XML output.  
  
'--docbook'      Generate a Docbook XML output.
```

Each elaboration generates a file with `'.diag'` extension. This file contains the messages that may be shown with the option `'--verbose'`.

12 How to install

Sgmltexi is made of two Perl executables: `sgmltexi` and `sgmltexi-sp2texi`. These files can be placed everywhere they can run without giving the path; that is: inside a directory listed by the environment variable `PATH`.

It is needed Perl as `/usr/bin/perl`. If your system is organized differently, you should modify the first line of these executables:

```
#!/usr/bin/perl
#...
```

Sgmltexi expect to find some files:

```
‘/usr/share/sgmltexi/sgmltexi.cat’
    the Sgmltexi SGML catalog;

‘/usr/share/sgmltexi/sgmltexi.dcl’
    the Sgmltexi SGML declaration;

‘/usr/share/sgmltexi/sgmltexi.dtd’
    the Sgmltexi DTD;

‘/usr/share/sgmltexi/entities/’
    the directory containing the standard ISO 8879 entity files.
```

All these files may be exactly where expected or there may be symbolic links to recreate the expected files.

12.1 Gettext

The messages that Sgmltexi shows may be translated. To install the already translated PO files, it is necessary to compile them like this:

```
msgfmt -o sgmltexi.mo it.po
```

In this example the file `it.po` is compiled and it is generated the file `sgmltexi.mo`. This generated file must be copied inside the right directory; in this case, may be `/usr/share/locale/it/LC_MESSAGES/`.

If you don't have installed the Perl-gettext module and you don't want to worry about it, you can comment the following instructions:

```
# We *don't* want to use gettext.
#use POSIX;
#use Locale::gettext;
#setlocale (LC_MESSAGES, "");
#textdomain ("sgmltexi");
```

Then you have to introduce a dummy `gettext()` function:

```
sub gettext
{
    return $_[0];
}
```

13 Dependencies

Sgmltexi depends on the following external software:

Perl the Perl interpreter;

SP

Jade the SGML parser known as SP, that may also be included inside Jade;

Perl-gettext

a Perl module that include Gettext support;

Texinfo

TeX to get a final typesetting.

14 Encoding

Sgmltexi has an incomplete support for ISO 8859-n encodings. It is incomplete because Texinfo is not able to reproduce all characters. There are two ways to define the use of one encoding with Sgmltexi: the option ‘`--input-encoding`’ and the attribute `charset` inside the `sgmltexi` element.

The choice has a different result. The option ‘`--input-encoding`’ generate a transformation of characters into SGML entities, and back to Texinfo code. This way, the Texinfo code is surely pure ASCII (ISO 646), and entities that have no corresponding Texinfo code, are shown like ‘`[ETH]`’. The use of the `charset` attribute results only on the command `@documentencoding`; on some occasions, the result may be good or not. Depending on the better result, it may be used one option or the other.

A good strategy may be the use of the `charset` attribute in any case, adding the option ‘`--input-encoding`’ when Texinfo doesn’t generate a good result alone, usually when typesetting for printing.

14.1 Standard and non-standard entities

Sgmltexi DTD include all standard ISO 8879 entities ([Appendix A \[ISOnum\]](#), page 61, [Appendix B \[ISOpub\]](#), page 63, [Appendix C \[ISOtech\]](#), page 65, [Appendix D \[ISolat1\]](#), page 67, [Appendix E \[ISolat2\]](#), page 69). In fact, non all entities are really supported by Texinfo, and when an unsupported entity is used, it is shown on the final typesetting like a name enclosed inside square brackets, like ‘`[ETH]`’.

Sgmltexi uses some non-standard entities, needed for compatibility with Texinfo. These are shown on the following table.

SGML macro	Texinfo command	Appearing	Description
<code>&dots;</code>	<code>@dots{}</code>	...	three dots
<code>&enddots;</code>	<code>@enddots{}</code>	four dots
<code>&TeX;</code>	<code>@TeX{}</code>	T _E X	the name "TeX"
<code>&result;</code>	<code>@result{}</code>	⇒	
<code>&expansion;</code>	<code>@expansion{}</code>	↦	
<code>&print;</code>	<code>@print{}</code>	⊢	
<code>&error;</code>	<code>@error{}</code>	error	
<code>&point;</code>	<code>@point{}</code>	★	
<code>&today;</code>	<code>@today{}</code>	26 January 2003	
<code>&esexcl;</code>	<code>@!</code>	!	ending sentence exclamation mark
<code>&esperiod;</code>	<code>@.</code>	.	ending sentence period
<code>&nes;</code>	<code>@:</code>	:	not ending sentence
<code>&esquest;</code>	<code>@?</code>	?	ending sentence question mark

15 Supported and unsupported Texinfo feature

Some Texinfo features are supported, others are not. Here is the list.

<code>@whitespace</code>	
<code>&emsp;</code>	
<code>@!</code>	<code>&esexcl</code> ; (end sentence exclamation mark)
<code>@"</code>	
<code>@'</code>	Use equivalent standard SGML entities, or ISO 8859-n characters if available.
<code>@*</code>	<code>
</code>
<code>@,{c}</code>	
<code>@,{C}</code>	Use equivalent standard SGML entities, or ISO 8859-n characters if available.
<code>@-</code>	<code><dh></code>
<code>@.</code>	<code>&esperiod</code> ; (end sentence period)
<code>@:</code>	<code>&nes</code> ; (not ending sentence)
<code>@=</code>	Unsupported. ¹
<code>@?</code>	<code>&esquest</code> ; (end of sentence question mark)
<code>@@</code>	Simply @.
<code>@^</code>	
<code>@‘</code>	Use equivalent standard SGML entities, or ISO 8859-n characters if available.
<code>@{</code>	
<code>@}</code>	Simply { and }.
<code>@~</code>	
<code>@AA{}</code>	
<code>@aa{}</code>	Use equivalent standard SGML entities, or ISO 8859-n characters if available.
<code>@acronym{abbrev}</code>	<code><acronym>abbrev</acronym></code>
<code>@AE{}</code>	
<code>@ae{}</code>	Use equivalent standard SGML entities, or ISO 8859-n characters if available.
<code>@afivepaper</code>	
<code>@afourpaper</code>	
<code>@afourlatex</code>	
<code>@afourwide</code>	Use command line option: ‘ <code>--paper=a5</code> ’, ‘ <code>--paper=a4</code> ’, ‘ <code>--paper=a4latex</code> ’, ‘ <code>--paper=a4wide</code> ’.
<code>@alias new=existing</code>	Unsupported. May be used inside the <code>texinfo</code> element.

¹ Are there standard SGML entities? I need advice.

`@anchor{name}`
 `<anchor id="name">`

`@appendix title`
`@appendixsec title`
`@appendixsection title`
`@appendixsubsec title`
`@appendixsubsection title`
`@appendixsubsubsec title`
`@appendixsubsubsection title`
 Appendices are enclosed inside the `appendix` element.

`@asis` The word `asis` is used as an argument for the `emphasis` attribute for `table`, `vtable` and `ftable` elements.

`@author author`
 `<author>author</author>`

`@b{text}` `<bold>text</bold>`

`@bullet{}`
 `•`

`@bye` `</sgmltexi>`

`@c comment`
`@comment comment`
 Unsupported, as SGML has its own comment symbol. May be used inside the `texinfo` element.

`@cartouche`
 `<cartouche>text-block</cartouche>`

`@center line-of-text`
 `<center>line-of-text</center>`
 Unsupported on title page.

`@centerchap title`
 Unsupported.

`@chapheading title`
 `<h1 type="heading">title</h1>`

`@chapter title`
 `<h1>title</h1>`

`@cindex entry`
 `<cindex entry="entry">`

`@cite{reference}`
 `<cite>reference</cite>`

`@clear flag`
 Unsupported. May be used inside the `texinfo` element.

```

@code{sample}
    <code>sample</code>

@command{command-name}
    <command>command-name</command>

@contents
    <contents>

@copyright{}
    Use equivalent standard SGML entities, or ISO 8859-n characters if available.

@defcodeindex index-name
    <defcodeindex>index-name</defcodeindex>

@defcv category class name
@defcvx category class name
    <defcv cat="category" class="class" name="name">
    [<defcvx cat="category" class="class" name="name">]...
    ...
    ...
    </defcv>

@deffn category name argument...
@deffnx category name argument...
    <deffn cat="category" name="name">
    <args>argument...</args>
    [<deffnx cat="category" name="name">
    <args>argument...</args>]...
    ...
    ...
    </deffn>

@defindex index-name
    <defindex>index-name</defcodeindex>

@definfoenclose newcmd, before, after,
    Unsupported. May be used inside the texinfo element.

@defivar class instance-variable-name
@defivarx class instance-variable-name
    <defivar class="class" name="instance-variable">
    [<defivarx class="class" name="instance-variable">]...
    ...
    ...
    </defivar>

@defmac macro-name argument...
@defmacx macro-name argument...
    <defmac name="macro-name">
    <args>argument...</args>
    [<defmacx name="macro-name">

```

```

        <args>argument...</args>] ...
        ...
        ...
    </defmac>

@defmethod class method-name argument...
@defmethodx class method-name argument...
    <defmethod class="class" name="method-name">
        <args>argument...</args>
    [<defmethodx class="class" name="method-name">
        <args>argument...</args>] ...
        ...
        ...
    </defmethod>

@defop category class name argument...
@defopx category class name argument...
    <defop cat="category" class="class" name="name">
        <args>argument...</args>
    [<defopx cat="category" class="class" name="name">
        <args>argument...</args>] ...
        ...
        ...
    </defop>

@defopt option-name
@defoptx option-name
    <defopt name="option-name">
    [<defoptx name="option-name">] ...
        ...
        ...
    </defopt>

@defspec special-form-name argument...
@defspecx special-form-name argument...
    <defspec name="special-form-name">
        <args>argument...</args>
    [<defspecx name="special-form-name">
        <args>argument...</args>] ...
        ...
        ...
    </defspec>

@deftp category name-of-type attribute...
@deftpx category name-of-type attribute...
    <deftp cat="category" name="name-of-type">
        <args>attribute...</args>
    [<deftpx cat="category" name="name-of-type">
        <args>attribute...</args>] ...
        ...

```

```

    ...
  </deftp>

@deftypefn classification data-type name argument...
@deftypefnx classification data-type name argument...
  <deftypefn cat="classification" type="data-type" name="name">
    <args>argument...</args>
  [<deftypefnx cat="classification" type="data-type" name="name">
    <args>argument...</args>] ...
    ...
  </deftypefn>

@deftypefun data-type function-name argument...
@deftypefunx data-type function-name argument...
  <deftypefun type="data-type" name="function-name">
    <args>argument...</args>
  [<deftypefunx type="data-type" name="function-name">
    <args>argument...</args>] ...
    ...
  </deftypefun>

@deftypeivar class data-type variable-name
@deftypeivarx class data-type variable-name
  <deftypeivar class="class" type="data-type" name="variable-name">
    [<deftypeivarx class="class" type="data-type" name="variable-
name">] ...
    ...
  </deftypeivar>

@deftypemethod class data-type method-name argument...
@deftypemethodx class data-type method-name argument...
  <deftypemethod class="class" type="data-type" name="method-name">
    <args>argument...</args>
  [<deftypemethodx class="class" type="data-type" name="method-
name">
    <args>argument...</args>] ...
    ...
  </deftypemethod>

@deftypeop category class data-type name argument...
@deftypeopx category class data-type name argument...
  <deftypeop cat="category" class="class" type="data-type" name="name">
    <args>argument...</args>
  [<deftypeopx cat="category" class="class" type="data-type" name="name">
    <args>argument...</args>] ...
    ...

```

```

    ...
  </deftypeop>

@deftypevar data-type variable-name
@deftypevarx data-type variable-name
    <deftypevar type="data-type" name="variable-name">
    [<deftypevarx type="data-type" name="variable-name">] ...
    ...
    ...
  </deftypevar>

@deftypevr classification data-type name
@deftypevr classification data-type name
    <deftypevr cat="classification" type="data-type" name="name">
    [<deftypevr cat="classification" type="data-type" name="name">] ...
    ...
    ...
  </deftypevr>

@defun function-name argument...
@defunx function-name argument...
    <defun name="function-name">
    <args>argument...</args>
    [<defunx name="function-name">
    <args>argument...</args>] ...
    ...
    ...
  </defun>

@defvar variable-name
@defvar variable-name
    <defvar name="variable-name">
    [<defvarx name="variable-name">] ...
    ...
    ...
  </defvar>

@defvr category name
@defvr category name
    <defvr cat="category" name="name">
    [<defvr cat="category" name="name">] ...
    ...
    ...
  </defvr>

@detailmenu
    <menu>
    ...
    ...
  <detailmenu>

```

```

        ...
        ...
    </detailmenu>
</menu>

@dfn{term}
    <dfn>term</dfn>

@dircategory dirpart
@direntry
    <infodir cat="dirpart">
        ...
        ...
    </infodir>

@display <display>text-block</display>

@dmn{dimension}
    <dmn>dimension</dmn>

@documentdescription description @end documentdescription
    <documentdescription content="description">

@documentencoding enc
    <sgmltexi charset="enc">
    This is the input character set, like it can be done with the Texinfo
    @documentencoding command. It is obscured by the ‘--input-encoding’
    option, that take precedence and generate a pure ISO 646 Texinfo output.

@documentlanguage cc
    <sgmltexi lang="cc">

@dotaccent{c}
    Use equivalent standard SGML entities, or ISO 8859-n characters if available.

@dots{}    &dots;

@email{address, displayed-text}
    <email email="address" name="displayed-text">

@emph{text}
    <emph>text</emph>

@env{environment-variable}
    <env>environment-variable</env>

@enddots{}
    &enddots;

@enumerate [number-or-letter]
    <enumerate [start="number-or-letter"]>
    <item>
        ...
        ...

```

```

    <item>
      ...
    </enumerate>

```

`@equiv{}` `≡`

`@error{}` `&error;`

`@evenfooting`

`@evenheading`

`@everyfooting`

`@everyheading`

Unsupported. May be used inside the `texinfo` element.

`@example`

```

    <example>
      ...
      block
      ...
    </example>

```

preformatted:

```

    <example>
    <pre>
      ...
      in-line
      ...
    </pre>
    </example>

```

literal:

```

    <example>
    <pre>
    <![CDATA[
      ...
      in-line
      ...
    ]]>
    </pre>
    </example>

```

`@exampleindent`

Unsupported. May be used inside the `texinfo` element.

`@exlamdown`

Use equivalent standard SGML entities, or ISO 8859-n characters if available.

`@exdent`

```

    <pre>
      ...

```

```

        <exdent>some exdented text</exdent>
        ...
    </pre>

@expansion{
    &expansion;

@file{filename}
    <file>filename</file>

@finalout
    Unsupported. Actually it is always included.

@index entry
    <index entry="entry">

@flushleft
    <flushleft>text</flushleft>

@flushright
    <flushright>text</flushright>

@footnote{text-of-footnote}
    <footnote>text-of-footnote</footnote>

@footnotestyle style
    <footnotestyle content="style">, or use command line option:
    '--footnotestyle=style'.

@format
    <format>
    <pre>
        ...
        ...
    </pre>
    </format>
literal:
    <format>
    <pre>
    <![CDATA[
        ...
        ...
    ]]>
    </pre>
    </format>

@frenchspacing
    <sgmltexi spacing="french">

@ftable formatting-command
    <ftable emphasis="command">
    <item>descriptive-item</item>

```

```

        [<itemx>descriptive-item</itemx>]...
        text-block...
        ...
        ...
    <item>descriptive-item</item>
    [<itemx>descriptive-item</itemx>]...
    text-block...
    ...
    ...
</ftable>

@group    <group>text-block</group>
@H{c}     Use equivalent standard SGML entities, or ISO 8859-n characters if available.
@heading title
    <h2 type="heading">title</h2>
@headings on-off-single-double
    <headings content="on-off-single-double-singleafter-
doubleafter">, or use command line option: '--headings=on-off-
single-double-singleafter-doubleafter'.
@html     <html>html_back_end_code</html>
@hyphenation{hy-phen-a-ted words}
    <hyphenation words="hy-phen-a-ted words">
@i{text}  <italic>text</italic>
@ifclear flag
    Unsupported. May be used inside the texinfo element.
@ifhtml   There are two possibilities: in-line and block conditionals.
    <ifhtml>in-line</ifhtml>
    <ifhtmlblock>
        block
        ...
    </ifhtmlblock>

```

SGML gives the possibility to use marked sections. These can be controlled with Sgmltexi with the option '--sgml-include' at the command line. For example, the SGML source may be like this:

```

<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
  <!ENTITY % HTML          "IGNORE">
  <!ENTITY % INFO          "IGNORE">
  <!ENTITY % TEX           "IGNORE">
  ...
  ...
]>
<sgmltexi>

```

```

...
...
<![%HTML;[
    <p>Here it is some text that is meant to appear only inside
    the HTML typesetting.</p>
]]>
<![%INFO;[
    <p>Here it is some other text that is meant to appear only
    inside the Info typesetting.</p>
]]>
<![%TEX;[
    <p>This text is meant to appear only inside the TeX
    typesetting.</p>
]]>
...
...
</sgmltexi>

```

Then, when typesetting for HTML, the option ‘--sgml-include=HTML’ must be used:

```
sgmltexi --sgml-include=HTML --html my_file.sgml
```

When typesetting for Info, the option ‘--sgml-include=INFO’ must be used:

```
sgmltexi --sgml-include=INFO --info my_file.sgml
```

The same way, when typesetting for TeX, the option ‘--sgml-include=TEX’ must be used:

```
sgmltexi --sgml-include=TEX --info my_file.sgml
```

@ifinfo There are two possibilities: in-line and block conditionals.

```

<ifinfo>in-line</ifinfo>
<ifinfoblock>
    block
...
</ifinfoblock>

```

SGML gives also the possibility to use marked sections. See **@ifhtml** for the explanation.

@ifnohtml

There are two possibilities: in-line and block conditionals.

```

<ifnohtml>in-line</ifnohtml>
<ifnohtmlblock>
    block
...
</ifnohtmlblock>

```

SGML gives the possibility to use marked sections. These can be controlled with Sgmltexi with the option ‘--sgml-include’ at the command line. For example, the SGML source may be like this:

```

<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
<!ENTITY % NOTHTML          "IGNORE">
<!ENTITY % NOTINFO          "IGNORE">
<!ENTITY % NOTTEX           "IGNORE">
...
...
]>
<sgmltexi>
...
...
<![%NOTHTML;[
    <p>Here it is some text that is meant to appear only outside
    the HTML typesetting.</p>
]]>
<![%NOTINFO;[
    <p>Here it is some other text that is meant to appear only
    outside the Info typesetting.</p>
]]>
<![%NOTTEX;[
    <p>This text is meant to appear only outside the TeX
    typesetting.</p>
]]>
...
...
</sgmltexi>

```

Then, when typesetting for HTML, the options ‘--sgml-include=NOTINFO’ and ‘--sgml-include=NOTTEX’ must be used:

```
sgmltexi --sgml-include=NOTINFO --sgml-include=NOTTEX --html my_file.sgml
```

When typesetting for Info, the options ‘--sgml-include=NOTHTML’ and ‘--sgml-include=NOTTEX’ must be used:

```
sgmltexi --sgml-include=NOTHTML --sgml-include=NOTTEX --info my_file.sgml
```

The same way, when typesetting for TeX, the options ‘--sgml-include=NOTINFO’ and ‘--sgml-include=NOTHTML’ must be used:

```
sgmltexi --sgml-include=NOTINFO --sgml-include=NOTHTML --tex my_file.sgml
```

@ifnotinfo

There are two possibilities: in-line and block conditionals.

```

<ifnotinfo>in-line</ifnotinfo>

<ifnotinfoblock>
    block
    ...
</ifnotinfoblock>

```

SGML gives also the possibility to use marked sections. See @ifnohtml for the explanation.

@ifnotplaintext

There are two possibilities: in-line and block conditionals.

```
<ifnotplaintext>in-line</ifnotplaintext>
<ifnotplaintextblock>
  block
  ...
</ifnotplaintextblock>
```

SGML gives also the possibility to use marked sections. See **@ifnohtml** for the explanation.

@ifnottex

There are two possibilities: in-line and block conditionals.

```
<ifnottex>in-line</ifnottex>
<ifnottexblock>
  block
  ...
</ifnottexblock>
```

SGML gives also the possibility to use marked sections. See **@ifnohtml** for the explanation.

@ifnotxml

There are two possibilities: in-line and block conditionals.

```
<ifnotxml>in-line</ifnotxml>
<ifnotxmlblock>
  block
  ...
</ifnotxmlblock>
```

SGML gives also the possibility to use marked sections. See **@ifnohtml** for the explanation.

@ifplaintext

There are two possibilities: in-line and block conditionals.

```
<ifplaintext>in-line</ifplaintext>
<ifplaintextblock>
  block
  ...
</ifplaintextblock>
```

SGML gives also the possibility to use marked sections. See **@ifhtml** for the explanation.

@ifset flag

Unsupported. May be used inside the **texinfo** element.

@iftex

There are two possibilities: in-line and block conditionals.

```
<iftex>in-line</iftex>
<iftexblock>
  block
```

```

    ...
</iftexblock>

```

SGML gives also the possibility to use marked sections. See `@ifhtml` for the explanation.

`@ifxml` There are two possibilities: in-line and block conditionals.

```

<ifxml>in-line</ifxml>

<ifxmlblock>
  block
  ...
</ifxmlblock>

```

SGML gives also the possibility to use marked sections. See `@ifhtml` for the explanation.

`@ignore` Unsupported. May be used inside the `texinfo` element. If it is not necessary to have the text included inside the generated Texinfo source, standard SGML comments may be used:

```

<!--
  This text is a comment, that is not reported
  inside the Texinfo generated source.
  ...
  ...
-->

```

`@image{filename, [width], [height]}`

```

<image name="filename" width="width" height="height">

```

`@include` Unsupported (see below). Use SGML mechanism instead, like this:

```

<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
  <!ENTITY GPL      SYSTEM "licenses/gpl.sgml">
  <!ENTITY BSD      SYSTEM "licenses/bsd.sgml">
  ...
  ...
]>
<sgmltexi>
  ...
  ...
<appendix>
  &GPL;
  &BSD;
  ...
  ...
</appendix>
  ...
</sgmltexi>

```

As it can be seen, the insertion of ‘licenses/gpl.sgml’ and ‘licenses/bsd.sgml’ happens when the SGML macro ‘&GPL;’ and ‘&BSD;’ appear inside the source.

If it is necessary to include a Texinfo file, the element `texinfo` may be used like this:

```
<p><texinfo>
  @include example.texi
</texinfo></p>
```

Please remember that `texinfo` is an in-line element.

```
@inforef{node-name, [entry-name], info-file-name}
  <inforef id="node-name" name="entry-name" info="info-file-name">
```

```
\input macro-definition-file
  Unsupported.
```

```
@item      See @table, @ftable, @vtable, @itemize, @enumerate and @multitable.
```

```
@itemize [mark]
  <itemize [mark="mark"]>
  <item>
    ...
    ...
  </item>
  ...
  ...
</itemize>
```

```
@itemx     See @table, @ftable and @vtable.
```

```
@kbd{keyboard-characters}
  <kbd>keyboard-characters</kbd>
```

```
@kbdinputstyle style
  <kbdstyle style="style">
```

```
@key{key-name}
  <key>key-name</key>
```

```
@kindex entry
  <kindex entry="entry">
```

```
@L{}
```

```
@l{}      Use equivalent standard SGML entities, or ISO 8859-n characters if available.
```

```
@lisp
  <lisp>
    ...
    block
    ...
  </lisp>
```

preformatted:

```

<lisp>
<pre>
    ...
    in-line
    ...
</pre>
</lisp>

```

literal:

```

<lisp>
<pre>
<![CDATA[
    ...
    in-line
    ...
]]>
</pre>
</lisp>

```

@lowersections

Unsupported.

@macro *macro-name* {*params*}

Unsupported. May be used inside the `texinfo` element.

@majorheading *title*

Unsupported. Maybe when Texinfo introduce @part.

@math{*mathematical-expression*}

$*mathematical-expression*$

@menu <menu>[*info-menu*</menu>]

@minus{} −

@multitable *column-width-spec*

```

<multitable>
<columnfraction>columnfraction</columnfraction>...
<raw>element [<tab>element]...</raw>...
...
</multitable>

<multitable>
<columnexample>text-example</columnexample>...
<raw>element [<tab>element]...</raw>...
...
</multitable>

```

@need *n* <need mils="*n*">

@node *name*, *next*, *previous*, *up*

Use like this for standard manual node handling:

```
<hn node="name">title</hn>
```

If it is required a complete control over nodes, also *next*, *previous* and *up* nodes may be specified, like this:

```
<hn node="name" next="next" prev="previous" up="up" >title</hn>
```

Sgmltexi doesn't make any validity check over manual node insertions.

@noindent

```
<p indent="off">
```

@novalidate

Unsupported. May be used inside the **texinfo** element.

0{}

o{} Use equivalent standard SGML entities, or ISO 8859-n characters if available.

@oddfooting

@oddheading

Unsupported. May be used inside the **texinfo** element.

@option{option-name}

```
<option>option-name</option>
```

@page <page>

@pagesizes [width][, height]

Unsupported.

@paragraphindent indent

Unsupported.

@pindex entry

```
<pindex entry="entry">
```

@point{} &point;

@pounds{}

Use equivalent standard SGML entities, or ISO 8859-n characters if available.

@print{} &print;

@printindex index-name

```
<printindex name="index-entry">
```

@pxref{node-name, [entry], [topic-or-title], [info-file], [manual]}

```
<pxref id="node-name" name="entry" title="topic-or-title"
info="info-file" ptitle="manual">
```

@questiondown{}

Use equivalent standard SGML entities, or ISO 8859-n characters if available.

@quotation

```
<quotation>
```

```
...
```

```
in-line
```

```
...
```

```
</quotation>
```

```

@r{text}  <roman>text</roman>

@raisesections
    Unsupported.

@ref{node-name, [entry], [topic-or-title], [info-file], [manual]}
    <ref id="node-name" name="entry" title="topic-or-title"
    info="info-file" ptitle="manual">

@refill    Unsupported.

@result{}
    &result;

@ringaccent{c}
    Use equivalent standard SGML entities, or ISO 8859-n characters if available.

@samp{text}
    <samp>text</samp>

@sc{text}
    <sc>text</sc>

@section title
    <h2>title</h2>

@set flag string
    Unsupported. May be used inside the texinfo element.

@setchapternewpage on-off-odd
    <setchapternewpage content="on-off-odd">, or use command line option:
    ‘--setchapternewpage=on-off-odd’.

@setcontentsaftertitlepage
    Unsupported.

@setfilename info-file-name
    <setfilename content="info-file-name">

@setshortcontentsaftertitlepage
    Unsupported.

@settitle title
    <settitle content="title">

@shortcontents
    <shortcontents>

@shorttitlepage title
    Unsupported.

@smallbook
    Use command line option: ‘--paper=small’.

@smalldisplay
    <smalldisplay>text-block</smalldisplay>

```

```
@smallexample
    <smallexample>
        ...
        block
        ...
    </smallexample>
```

preformatted:

```
<smallexample>
<pre>
    ...
    in-line
    ...
</pre>
</smallexample>
```

literal:

```
<smallexample>
<pre>
<![CDATA[
    ...
    in-line
    ...
]]>
</pre>
</smallexample>
```

```
@smallformat
```

```
<smallformat>
<pre>
    ...
    ...
</pre>
</smallformat>
```

literal:

```
<smallformat>
<pre>
<![CDATA[
    ...
    ...
]]>
</pre>
</smallformat>
```

```
@smalllisp
```

```
<smalllisp>
    ...
    block
    ...
```

```

        </smalllisp>
preformatted:
    <smalllisp>
    <pre>
        ...
        in-line
        ...
    </pre>
    </smalllisp>
literal:
    <smalllisp>
    <pre>
    <![CDATA[
        ...
        in-line
        ...
    ]]>
    </pre>
    </smalllisp>

@sp lines
    <sp lines="lines">

@ss{}      Use equivalent standard SGML entities, or ISO 8859-n characters if available.

@strong{text}
    <strong>text</strong>

@subheading title
    <h3 type="heading">title</h3>

@subsection title
    <h3>title</h3>

@subsubheading title
    <h4 type="heading">title</h4>

@subsubsection title
    <h4>title</h4>

@subtitle subtitle
    <subtitle>subtitle</subtitle>

@summarycontents
    <summarycontents>

@syncodeindex from-index into-index
    <syncodeindex from="from-index" to="into-index">

@synindex from-index into-index
    <synindex from="from-index" to="into-index">

@t{text}  <typewriter>text</typewriter>

```

```

@tab      See @multitable.

@table formatting-command
      <table emphasis="command">
      <item>descriptive-item</item>
      [<itemx>descriptive-item</itemx>] ...
      text-block...
      ...
      ...
      <item>descriptive-item</item>
      [<itemx>descriptive-item</itemx>] ...
      text-block...
      ...
      ...
    </table>

@TeX{}    &TeX;

@tex      <tex>tex_back_end_code</tex>

@thischapter
@thischaptername
@thisfile
@thispage
@thistitle
      Unsupported. May be used inside the texinfo element.

@tie{}    &nbsp;

@tieaccent{cc}
      Unsupported.2

@index entry
      <tindex entry="entry">

@title title
      <title>title</title>

@titlefont{text}
      Unsupported.

@titlepage
      Unsupported. See Sgmltexi structure.

@today    &today;

@top      Generated automatically; in other words, it is hidden to the user.

@u{c}
@ubaraccent{c}
@udotaccent{c}
      Unsupported.3

```

² Is there a standard SGML entity? I need advice.

³ Are there standard SGML entities? I need advice.

```

@unnumbered title
    <h1 type="unnumbered">title</h1>

@unnumberedsec title
    <h2 type="unnumbered">title</h2>

@unnumberedsubsec title
    <h3 type="unnumbered">title</h3>

@unnumberedsubsubsec title
    <h4 type="unnumbered">title</h4>

@uref{url, [displayed-text], [replacement]}
    <uref uri="url" name="displayed-text" replace="replacement">

@url{url}
    <url>url</url>

@v{c}      Unsupported.4

@value{flag}
    Unsupported. May be used inside the texinfo element.

@var{metasyntactic-variable}
    <var>metasyntactic-variable</var>

@verb{xliteral_textx}
    <verb char="x"><![CDATA[literal_text]]></verb>

@verbatim literal_text @end verbatim
    <verbatim>
    <[CDATA[
    literal_text
    ]]>
    </verbatim>

@verbatiminclude file
    <verbatiminclude file="file">

@vindex entry
    <vindex entry="entry">

@vskip amount
    Unsupported. May be used inside the texinfo element.

@vtable formatting-command
    <vtable emphasis="command">
    <item>descriptive-item</item>
    [<itemx>descriptive-item</itemx>] ...
    text-block...
    ...
    ...
    <item>descriptive-item</item>

```

⁴ Are there standard SGML entities? I need advice.

```

        [<itemx>descriptive-item</itemx>]...
        text-block...
        ...
        ...
    </vtable>

@w{text}  <whole>text</whole>

@xref{node-name, [entry], [topic-or-title], [info-file], [manual]}
    <xref id="node-name" name="entry" title="topic-or-title"
    info="info-file" ptitle="manual">

```

Appendix A Supported ISOnum entities: numeric and special graphic

The following table shows the current state of supported ISOnum entities.

SGML macro	Appearing	Description
½	1/2	fraction one-half
½	1/2	fraction one-half
¼	1/4	fraction one-quarter
¾	3/4	fraction three-quarters
⅛	1/8	fraction one-eighth
⅜	3/8	fraction three-eighths
⅝	5/8	fraction five-eighths
⅞	7/8	fraction seven-eighths
¹	[sup1]	superscript one
²	[sup2]	superscript two
³	[sup3]	superscript three
+	+	plus sign
±	+/-	plus-or-minus sign
<	<	less-than sign
=	=	equals sign
>	>	greater-than sign
÷	[divide]	divide sign
×	[times]	multiply sign
¤	[curren]	general currency sign
£	£	pound sign
$	\$	dollar sign
¢	[cent]	cent sign
¥	[yen]	yen sign
#	#	number sign
%	%	percent sign
&	&	ampersand
*	*	asterisk
@	@	commercial at
[[left square bracket
\	\	reverse solidus
]]	right square bracket
{	{	left curly bracket
―	[horbar]	horizontal bar
|		vertical bar
}	}	right curly bracket
µ	[micro]	micro sign
Ω	[ohm]	ohm sign
°	[deg]	degree sign
º	[ordm]	ordinal indicator, masculine
ª	[ordf]	ordinal indicator, feminine
§	[sect]	section sign
¶	[para]	pilcrow (paragraph sign)
·	•	middle dot

←	[larr]	leftward arrow
→	[rarr]	rightward arrow
↑	[uarr]	upward arrow
↓	[darr]	downward arrow
©	©	copyright sign
®	[reg]	registered sign
™	[trade]	trade mark sign
¦	[brvbar]	broken (vertical) bar
¬	[not]	not sign
♪	[sung]	music note (sung text sign)
!	!	exclamation mark
&ixcl;	¡	inverted exclamation mark
"	"	quotation mark
'	'	apostrophe
((left parenthesis
))	right parenthesis
,	,	comma
_	-	low line
‐	-	hyphen
.	.	full stop, period
/	/	solidus
:	:	colon
;	;	semicolon
?	?	question mark
¿	¿	inverted question mark
«	[laquo]	angle quotation mark, left
»	[raquo]	angle quotation mark, right
‘	[lsquo]	single quotation mark, left
’	[rsquo]	single quotation mark, right
“	[ldquo]	double quotation mark, left
”	[rdquo]	double quotation mark, right
 		no break (required) space
­	[shy]	soft hyphen

Appendix B Supported ISOpub entities: publishing

The following table shows the current state of supported ISOpub entities.

SGML macro	Appearing	Description
 		em space
 	[ensp]	en space (1/2-em)
 	[emsp3]	1/3-em space
 	[emsp4]	1/4-em space
 	[numsp]	digit space (width of a number)
 	[puncsp]	punctuation space (width of comma)
 	[thinsp]	thin space (1/6-em)
 	[hairsp]	hair space
—	[mdash]	em dash
–	[ndash]	en dash
‐	[dash]	hyphen (true graphic)
␣	[blank]	significant blank symbol
…	[hellip]	ellipsis (horizontal)
‥	[nldr]	double baseline dot (en leader)
⅓	1/3	fraction one-third
⅔	2/3	fraction two-thirds
⅕	1/5	fraction one-fifth
⅖	2/5	fraction two-fifths
⅗	3/5	fraction three-fifths
⅘	4/5	fraction four-fifths
⅙	1/6	fraction one-sixth
⅚	5/6	fraction five-sixths
℅	[incare]	in-care-of symbol
█	[block]	full block
▀	[uhblk]	upper half block
▄	[lhblk]	lower half block
░	[blk14]	25% shaded block
▒	[blk12]	50% shaded block
▓	[blk34]	75% shaded block
▮	[marker]	histogram marker
○	[cir]	circle, open
□	[squ]	square, open
▭	[rect]	rectangle, open
▵	[utri]	up triangle, open
▿	[dtri]	down triangle, open
☆	[star]	star, open
•	•	round bullet, filled
▪	[squf]	sq bullet, filled
▴	[utrif]	up tri, filled
▾	[dtrif]	dn tri, filled
◂	[ltrif]	l tri, filled
▸	[rtrif]	r tri, filled
♣	[clubs]	club suit symbol

♦	[diams]	diamond suit symbol
♥	[hearts]	heart suit symbol
♠	[spades]	spades suit symbol
✠	[malt]	maltese cross
†	[dagger]	dagger
‡	[Dagger]	double dagger
✓	[check]	tick, check mark
✗	[ballot]	ballot cross
♯	[sharp]	musical sharp
♭	[flat]	musical flat
♂	[male]	male symbol
♀	[female]	female symbol
☎	[phone]	telephone symbol
⌕	[telrec]	telephone recorder symbol
℗	[copysr]	sound recording copyright sign
⁁	[caret]	caret (insertion mark)
‚	[lsquor]	rising single quote, left (low)
„	[ldquor]	rising dbl quote, left (low)
ﬀ	[fflig]	mall ff ligature
ﬁ	[filig]	mall fi ligature
fj	[fjlig]	mall fj ligature
ﬃ	[ffilig]	mall ffi ligature
ﬀ	[fflig]	mall ffl ligature
&flig;	[flig]	mall fl ligature
…	[mldr]	m leader
”	[rdquor]	rising dbl quote, right (high)
’	[rsquor]	rising single quote, right (high)
⋮	[vellip]	vertical ellipsis
⁃	[hybull]	rectangle, filled (hyphen bullet)
◊	[loz]	lozenge - lozenge or total mark
⧫	[lozf]	blacklozenge - lozenge, filled
◃	[ltri]	l triangle, open
▹	[rtri]	r triangle, open
★	[starf]	bigstar - star, filled
♮	[natur]	natural - music natural
℞	[rx]	pharmaceutical prescription (Rx)
✶	[sext]	sextile (6-pointed star)
⌖	[target]	register mark or target
⌍	[dlcrop]	downward left crop mark
⌌	[drcrop]	downward right crop mark
⌏	[ulcrop]	upward left crop mark
⌎	[urcrop]	upward right crop mark

Appendix C Supported ISOtech entities: general technical

The following table shows the current state of supported ISOtech entities.

SGML macro	Appearing	Description
ℵ	[aleph]	aleph, Hebrew
∧	[and]	logical and
&ang90;	[ang90]	right (90 degree) angle
∢	[angsph]	angle-spherical
≈	[ap]	approximate
∵	[becaus]	because
⊥	[bottom]	perpendicular
∩	[cap]	intersection
≅	[cong]	congruent with
∮	[conint]	contour integral operator
∪	[cup]	union or logical sum
≡	\equiv	identical with
∃	[exist]	at least one exists
∀	[forall]	for all
ƒ	[fnof]	function of (italic small f)
≥	[ge]	greater-than-or-equal
⇔	[iff]	if and only if
∞	[infin]	infinity
∫	[int]	integral operator
∈	[isin]	set membership
⟨	[lang]	left angle bracket
⇐	[lArr]	is implied by
≤	[le]	less-than-or-equal
−	—	minus sign
∓	[mnplus]	minus-or-plus sign
∇	[nabla]	del, Hamilton operator
≠	[ne]	not equal
∋	[ni]	contains
∨	[or]	logical or
∥	[par]	parallel
∂	[part]	partial differential
‰	[permil]	per thousand
⊥	[perp]	perpendicular
′	[prime]	prime or minute
″	[Prime]	double prime or second
∝	[prop]	is proportional to
√	[radic]	radical
⟩	[rang]	right angle bracket
⇒	[rArr]	implies
∼	[sim]	similar
≃	[sime]	similar, equals
□	[square]	square
⊂	[sub]	subset or is implied by

⊆	[sube]	subset, equals
⊃	[sup]	superset or implies
⊇	[supe]	superset, equals
∴	[there4]	therefore
‖	[Verbar]	dbl vertical bar
Å	[angst]	capital A, ring
ℬ	[bernou]	bernoulli function (script capital B)
∘	[compfn]	composite function (small circle)
¨	[Dot]	dieresis or umlaut mark
⃜	[DotDot]	four dots above
ℋ	[hamilt]	hamiltonian (script capital H)
ℒ	[lagran]	lagrangian (script capital L)
∗	[lowast]	low asterisk
∉	[notin]	negated set membership
ℴ	[order]	order of (script small o)
ℳ	[phmmat]	physics M-matrix (script capital M)
⃛	[tdot]	three dots above
‴	[tprime]	triple prime
≙	[wedgeq]	corresponds to (wedge, equals)

Appendix D Supported ISolat1 entities: added latin 1

The following table shows the current state of supported ISolat1 entities.

SGML macro	Appearing	Description
á	á	small a, acute accent
Á	Á	capital A, acute accent
â	â	small a, circumflex accent
Â	Â	capital A, circumflex accent
à	à	small a, grave accent
À	À	capital A, grave accent
å	å	small a, ring
Å	Å	capital A, ring
ã	ã	small a, tilde
Ã	Ã	capital A, tilde
ä	ä	small a, dieresis or umlaut mark
Ä	Ä	capital A, dieresis or umlaut mark
æ	æ	small ae diphthong (ligature)
Æ	Æ	capital AE diphthong (ligature)
ç	ç	small c, cedilla
Ç	Ç	capital C, cedilla
ð	[eth]	small eth, Icelandic
Ð	[ETH]	capital Eth, Icelandic
é	é	small e, acute accent
É	É	capital E, acute accent
ê	ê	small e, circumflex accent
Ê	Ê	capital E, circumflex accent
è	è	small e, grave accent
È	È	capital E, grave accent
ë	ë	small e, dieresis or umlaut mark
Ë	Ë	capital E, dieresis or umlaut mark
í	í	small i, acute accent
Í	Í	capital I, acute accent
î	î	small i, circumflex accent
Î	Î	capital I, circumflex accent
ì	ì	small i, grave accent
Ì	Ì	capital I, grave accent
ï	ï	small i, dieresis or umlaut mark
Ï	Ï	capital I, dieresis or umlaut mark
ñ	ñ	small n, tilde
Ñ	Ñ	capital N, tilde
ó	ó	small o, acute accent
Ó	Ó	capital O, acute accent
ô	ô	small o, circumflex accent
Ô	Ô	capital O, circumflex accent
ò	ò	small o, grave accent
Ò	Ò	capital O, grave accent
ø	ø	small o, slash

Ø	Ø	capital O, slash
õ	õ	small o, tilde
Õ	Õ	capital O, tilde
ö	ö	small o, dieresis or umlaut mark
Ö	Ö	capital O, dieresis or umlaut mark
ß	ß	small sharp s, German (sz ligature)
þ	[thorn]	small thorn, Icelandic
Þ	[THORN]	capital THORN, Icelandic
ú	ú	small u, acute accent
Ú	Ú	capital U, acute accent
û	û	small u, circumflex accent
Û	Û	capital U, circumflex accent
ù	ù	small u, grave accent
Ù	Ù	capital U, grave accent
ü	ü	small u, dieresis or umlaut mark
Ü	Ü	capital U, dieresis or umlaut mark
ý	ý	small y, acute accent
Ý	Ý	capital Y, acute accent
ÿ	ÿ	small y, dieresis or umlaut mark

Appendix E Supported ISolat2 entities: added latin 2

The following table shows the current state of supported ISolat2 entities.

SGML macro	Appearing	Description
ă	[abreve]	small a, breve
Ă	[Abreve]	capital A, breve
ā	[amacr]	small a, macron
Ā	[Amacr]	capital A, macron
ą	[aogon]	small a, ogonek
Ą	[Aogon]	capital A, ogonek
ć	ć	small c, acute accent
Ć	Ć	capital C, acute accent
č	[ccaron]	small c, caron
Č	[Ccaron]	capital C, caron
ĉ	ĉ	small c, circumflex accent
Ĉ	Ĉ	capital C, circumflex accent
ċ	ċ	small c, dot above
Ċ	Ċ	capital C, dot above
ď	[dcaron]	small d, caron
Ď	[Dcaron]	capital D, caron
đ	.	small d, stroke
Đ	[Dstrok]	capital D, stroke
ě	[ecaron]	small e, caron
Ě	[Ecaron]	capital E, caron
ė	ė	small e, dot above
Ė	Ė	capital E, dot above
ē	[emacr]	small e, macron
Ē	[Emacr]	capital E, macron
ę	[eogon]	small e, ogonek
Ę	[Eogon]	capital E, ogonek
ǵ	ǵ	small g, acute accent
ğ	[gbreve]	small g, breve
Ğ	[Gbreve]	capital G, breve
Ģ	Ġ	capital G, cedilla
ĝ	ġ	small g, circumflex accent
Ĝ	Ġ	capital G, circumflex accent
ġ	ḡ	small g, dot above
Ġ	Ḡ	capital G, dot above
ĥ	ĥ	small h, circumflex accent
Ĥ	Ĥ	capital H, circumflex accent
ħ	[hstrok]	small h, stroke
Ħ	"	capital H, stroke
İ	İ	capital I, dot above
Ī	[Imacr]	capital I, macron
ī	[imacr]	small i, macron
ĳ	[ijlig]	small ij ligature
Ĳ	[IJlig]	capital IJ ligature

ı	[inodot]	small i without dot
į	[iogon]	small i, ogonek
Į	[Iogon]	capital I, ogonek
ĩ	ï	small i, tilde
Ĩ	Ï	capital I, tilde
ĵ	ĵ	small j, circumflex accent
Ĵ	Ĵ	capital J, circumflex accent
ķ	ķ	small k, cedilla
Ķ	Ķ	capital K, cedilla
ĸ	[kgreen]	small k, Greenlandic
ĺ	ĺ	small l, acute accent
Ĺ	Ł	capital L, acute accent
ľ	[lcaron]	small l, caron
Ľ	[Lcaron]	capital L, caron
ļ	ļ	small l, cedilla
Ļ	Ļ	capital L, cedilla
ŀ	[lmidot]	small l, middle dot
Ŀ	[Lmidot]	capital L, middle dot
&lstroke;	ł	small l, stroke
&Lstroke;	Ł	capital L, stroke
ń	ń	small n, acute accent
Ń	Ń	capital N, acute accent
ŋ	[eng]	small eng, Lapp
Ŋ	[ENG]	capital ENG, Lapp
ŉ	[napos]	small n, apostrophe
ň	[ncaron]	small n, caron
Ň	[Ncaron]	capital N, caron
ņ	ņ	small n, cedilla
Ņ	Ņ	capital N, cedilla
ő	ö	small o, double acute accent
Ő	Ö	capital O, double acute accent
Ō	[Omacr]	capital O, macron
ō	[omacr]	small o, macron
œ	œ	small oe ligature
Œ	Œ	capital OE ligature
ŕ	ř	small r, acute accent
Ŕ	Ř	capital R, acute accent
ř	[rcaron]	small r, caron
Ř	[Rcaron]	capital R, caron
ŗ	ŗ	small r, cedilla
Ŗ	Ř	capital R, cedilla
ś	ś	small s, acute accent
Ś	Ś	capital S, acute accent
š	[scaron]	small s, caron
Š	[Scaron]	capital S, caron
ş	š	small s, cedilla

Ş	Š	capital S, cedilla
ŝ	š	small s, circumflex accent
Ŝ	Ŝ	capital S, circumflex accent
ť	[tcaron]	small t, caron
Ť	[Tcaron]	capital T, caron
ţ	ţ	small t, cedilla
Ţ	Ț	capital T, cedilla
ŧ		small t, stroke
Ŧ	[Tstrok]	capital T, stroke
ŭ	[ubreve]	small u, breve
Ŭ	[Ubreve]	capital U, breve
ű	ů	small u, double acute accent
Ű	Ů	capital U, double acute accent
ū	[umacr]	small u, macron
Ū	[Umacr]	capital U, macron
ų	[uogon]	small u, ogonek
Ų	[Uogon]	capital U, ogonek
ů	ü	small u, ring
Ů	Ü	capital U, ring
ũ	ũ	small u, tilde
Ũ	Ũ	capital U, tilde
ŵ	ŵ	small w, circumflex accent
Ŵ	Ŵ	capital W, circumflex accent
ŷ	ÿ	small y, circumflex accent
Ŷ	Ÿ	capital Y, circumflex accent
Ÿ	ÿ	capital Y, dieresis or umlaut mark
ź	ź	small z, acute accent
Ź	Ź	capital Z, acute accent
ž	[zcaron]	small z, caron
Ž	[Zcaron]	capital Z, caron
ż	ẏ	small z, dot above
Ż	Ẑ	capital Z, dot above

GNU Free Documentation License

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of

the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations

requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

A

abstract 7
 admin 4
 appendix 12
 author 7, 8

B

block 17
 body 11

C

cat 6
 charset 2
 content 4, 5
 contents 10
 copyright 7, 9
 coverart 7, 9

D

dedications 10
 defcodeindex 6
 defindex 4, 6
 dependencies 36

E

encoding 37
 entities 37

F

footnotestyle 4, 5
 from 6
 frontcovertext 7, 8

G

Gettext 35

H

h1 11
 h2 11
 h3 11
 h4 11
 head 3
 heading 5
 how to use Sgmltexi 33

I

in-line 17
 indexheading 12
 infodir 4, 6
 installation 35
 intro 11
 ISOlat1 67
 ISOlat2 69
 ISOnum 61
 ISOpub 63
 ISotech 65

L

lang 2
 legal 7, 9
 license 7, 9

M

menu 11
 menu 14

N

name 6
 node 14

P

partheading 11
 PATH 35
 printindex 12
 publishnote 7, 9

S

setchapternewpage 4, 5
 setfilename 4
 settitle 4, 5
 shortcontents 10
 spacing 2
 standard entities 37
 structure 2
 subtitle 7
 summarycontents 10
 supported 38
 syncodeindex 4, 6
 synindex 4, 6

T

title 7
 titlepage 7
 to 6
 tomeheading 11
 tpextra 8

U

unsupported 38