

Link

[OS B.4.4.22, Tutorial 1–7]

The `link` category is used to define a hyperlink in the Edit window, HTML output, and PDF output. The link can be hard-coded in the FOSI, or an attribute in the document can provide an ID or URI.

OutSpec DTD fragment

```
<!ELEMENT link - o EMPTY>
<!ATTLIST link
  href CDATA -- ATI: an URI or idref -- #IMPLIED
  show (replace|new|embed) -- ATI: traversal method -- #IMPLIED
  actuate (user|auto) -- ATI: link trigger -- #IMPLIED>
```

NOTE: Link does not have an `inherit` attribute. The category must be coded in an `e-i-c` to have any effect. When `link` is specified in an `e-i-c`, any empty characteristics default to the `envdesc`, `docdesc`, or system default setting, as detailed in **Figure 21 Inheritance and defaulting flow chart** on page 48.

The `link` category's only supported characteristic is described in the following table.

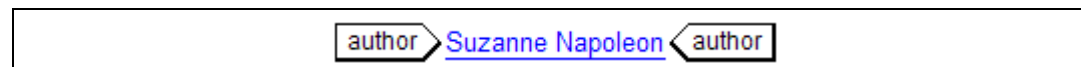
Table 72 Link characteristic

Characteristic	Values	Default	Notes
<code>href</code>	ID or URL	—	Specifies the ID of another element in the same document or a URL.

Link examples

The FOSI fragment in this example shows a `link` category that is hard-coded in the FOSI. Double-clicking on the link in the Edit window or in a PDF viewer opens a browser and displays the designated web page, if it can be found.

Figure 257 Hard-coded link in Edit window



FOSI fragment

```
<e-i-c gi="author">
<charlist inherit="1" charsubsetref="block blue underline">
<link href="http://www.fosiexpert.com/index.html">
...
```

KEY

Monospaced fonts are used for user input and FOSI categories, characteristics, and attributes.

In OutSpec DTD fragments, a single underscore means support only for print/PDF output. A double underscore indicates both Edit window and print/PDF support.

The next figure illustrates a link that is supplied by an attribute value.

Figure 258 Attribute-based link in Edit window



FOSI fragment

```
<e-i-c gi="title">
<charlist inherit="1" charsubsetref="title blue underline"></charlist>
<att>
<fillval attname="link" attloc="title" fillcat="link" fillchar="href">
<charsubset>
<link>
...
```

Also see **Figure 294 Gentext is hard-coded link** on page 542 for another example of the link category.

How to output PDF links, bookmarks, document properties, and viewer settings

Arbortext Editor supports generation of PDF links, bookmarks, document properties, and viewer settings. They are all discussed below.

- ▷ PDF links, page 541
- ▷ PDF bookmarks, page 546
- ▷ PDF document properties, page 553
- ▷ PDF viewer settings, page 554

NOTE: These techniques apply only to the direct PDF method.

PDF links

Automatically generated, FOSI-generated, and manually authored PDF links are discussed below.

- ▷ Automatically generated PDF links, page 541
- ▷ FOSI-generated PDF links, page 541
- ▷ Authored PDF links, page 542
- ▷ PDF link-related ACL, page 545

Automatically generated PDF links

Publishing directly to PDF automatically creates links that go from:

- Each entry in the table of contents to the page on which that entry appears
- A cross reference to the page on which the cross-referenced element appears
- Each index entry to its corresponding page number

FOSI-generated PDF links

The `link` category (page 472) is used to generate PDF links to:

- a named destination
- a URL

`link` can be used in different ways:

- In the resolved charlist of an `e-i-c`.
The `href` characteristic specifies a hard-coded URL or a hard-coded ID that presumably is included in the document.
- In a `fillval` charsubset.
An attribute on an element in the document specifies an ID or URL, which the `fillval` category assigns to the `href` characteristic on the `link` category.
- In `usetext` subchars.
Clicking on the `gentext` output by the `usetext` source activates the `link` category coded in the `usetext` subchars.

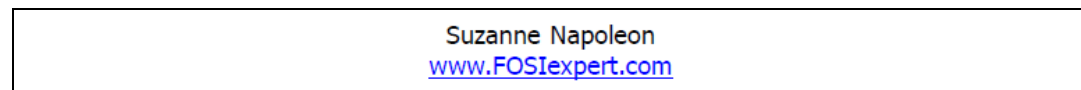
FOSI-generated PDF links examples

See **Figure 257 Hard-coded link in Edit window** on page 472 for an example of link coded in a charlist.

See **Figure 258 Attribute-based link in Edit window** on page 473 for an example of link coded in an attribute rule.

In the following example, the link is hard-coded in the subchars for the generated text. The graphic shows the PDF output.

Figure 294 Gentext is hard-coded link



XML fragment

```
<author>Suzanne Napoleon</author>
```

FOSI fragment

```
<e-i-c gi="author">
<charlist inherit="1" charsubsetref="block center">
<usetext source="\www.FOSIexpert.com\" placemnt="after">
<subchars charsubsetref="block blue underline">
<link href="http://www.fosiexpert.com/index.html">
...

```

Authored PDF links

Three ways of authoring PDF links are discussed in this section.

- ▷ DCF-declared PDF link and target elements, page 542
- ▷ <_link> and <_target> PIs for PDF links, page 545
- ▷ <_link> and <_target> examples, page 545

DCF-declared PDF link and target elements

An element declared as a link in the DCF (doctype configuration file) file can be inserted in a document using Insert→Link.... An element declared as a target in the DCF can be inserted with Insert→Link Target.... A declared link element must have an attribute to specify its target. A declared target element must have an ID attribute to be used by the linking element.

An DCF-declared link can specify:

- Linking within a PDF file to a named destination.
- Linking to a named destination or a specific page in another PDF file.
- Linking to a URL or URL anchor.
- Linking to an element declared in the DCF as a target.

Please see Arbortext Editor documentation for information on declaring and inserting these links.

Links within a PDF file

A link within a PDF file (internal link) is specified by a named destination created by an ID on an element in the source document. These links are also functional in the Edit window.

Clicking on an internal link in a PDF file loads the relevant page in the PDF viewer. Clicking on an internal link in the Edit window moves the cursor to the element with the matching ID.

Links from one PDF file to another

Clicking on a link to another PDF file (external link) loads the target PDF file in the PDF viewer. The opening position of the target file can be specified in two ways:

- As a page number (`#page=`), which displays the specified page in the PDF viewer window.
- As a named destination (`#nameddest=`) created by an ID in the document, which positions the named destination at the top of the PDF viewer window.

Links to a URL

Clicking on a URL link in a PDF viewer opens a browser window that displays the specified page, assuming it is available. If a URL anchor is specified, it is displayed at the top of the browser window.

Authored PDF links examples

These examples use the same DTD and DCF fragments.

In the first figure, the link is to a named destination in the same PDF file. This link is also functional in the Edit window.

Figure 295 Authored internal PDF link

(refer to [GLOSSARY](#)).

DTD fragment

```
<!ELEMENT pdflink ANY>
<!ATTLIST pdflink linkto ID #IMPLIED>
```

DCF fragment

```
<Specials>
<Link element="pdflink" idref="linkto" primary="yes"/>
```

XML fragment

```
<para>...(refer to <pdflink linkto="glossary">Glossary</pdflink>).
...
<appendix id="glossary"><title>Glossary</title>
...
```

FOSI fragment

```
<e-i-c gi="pdflink">
<charlist inherit="1" charsubsetref="blue underline allcaps"></charlist>
...
```

In the next figure, the link is to a named destination in another PDF file.

Figure 296 Authored external PDF links

when today's numbers are compared with
[yesterday's](#).
 Refer to [yesterday's End Notes](#) for sources.

XML fragment — Current document

```
<para>...when today's numbers are compared with <pdflink
linkto="yesterday.pdf#page=2">yesterday's</pdflink>.</para>
<para>Refer to <pdflink linkto="yesterday.pdf#nameddest=endnotes">
yesterday's End Notes</pdflink> for sources.</para>
```

XML fragment — External document

```
<appendix id="endnotes">
<title>End Notes</title>
...
```

FOSI fragment

```
<e-i-c gi="pdflink">
<charlist inherit="1" charsubsetref="blue underline"></charlist>
...
```

The last example shows a link from a PDF file to a URL.

Figure 297 Authored link to a URL



XML fragment

```
<item>FOSI resource: <pdfink linkto="http://www.FOSIexpert.com">
FOSIexpert.com</pdfink></item>
```

FOSI fragment

```
<e-i-c gi="pdfink">
<charlist inherit="1" charsubsetref="gray underline"></charlist>
</e-i-c>
```

<_link> and <_target> PIs for PDF links

<_link> and <_target> hyperlinks can be inserted in a document by selecting Tools→Create Hyperlink..., as described in **Link and target processing instructions** on page 680.

NOTE: <_link> and <_target> are being deprecated in Arbortext Editor.

<_link> and <_target> examples

See **Link and target examples** on page 682 for examples.

PDF link-related ACL

The following environmental variables affect PDF linking:

- APTNOPDFLINKS controls whether links are created in PDF files. The default is no. Yes disables links, although bookmarks for a table of contents and a list of tables are still created.
NOTE: This environment variable has no effect on HTML output.
- APTNAMEDDESTTOPAGE set to any value converts named destinations in PDF links to page numbers.

PDF bookmarks

Bookmarks can be automatically generated in two ways that are not mutually exclusive.

- Based on declarations in the DCF file
- Based on <atidmd> namespace markup coded in the FOSI

These are covered in the sections below.

- ▷ DCF-based PDF bookmarks, page 546
- ▷ FOSI-based PDF bookmarks, page 551

DCF-based PDF bookmarks

When a PDF file is created, bookmarks are automatically generated according to `ElementOption` settings in the doctype's DCF (doctype configuration file). Specifically, every element that is declared as the title of an element that is declared as division is bookmarked in the PDF file.

Nesting of PDF bookmarks follows the hierarchy of the document structure. Table titles, however, are listed after the hierarchical bookmarks.

The first block of text in the title is bookmarked for PDF, which could be generated text rather than authored text. To bookmark authored text instead, code the gentext on the division `e-i-c` instead of on the title `e-i-c`, as shown in **Figure 300**.

If a title is suppressed and there is no title gentext, the first available text is output as the bookmark.

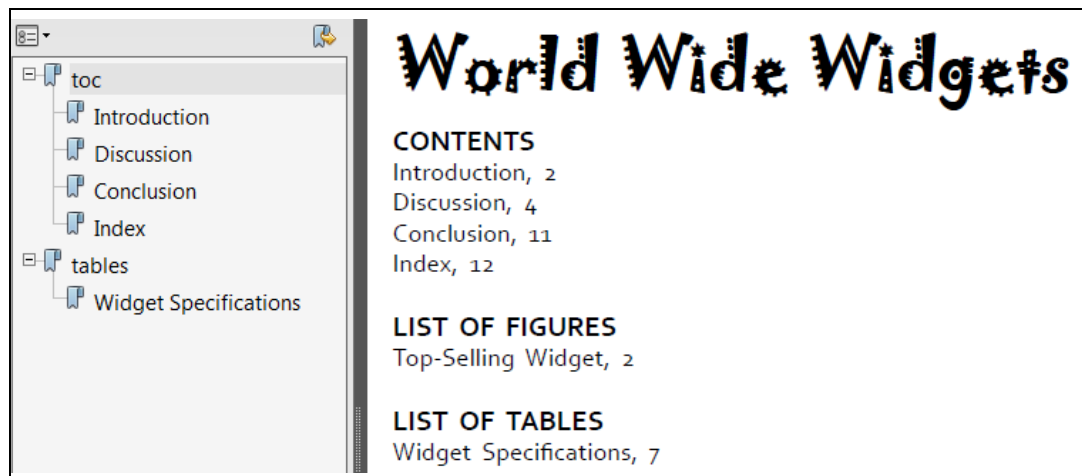
DCF-based bookmarks examples

In the first example of DCF-based bookmarks, <index> is a singleton that is declared in the DCF as a division and <title> is declared as a title. The `e-i-c` for <index> outputs a <title> (as a pseudo-element), which creates a PDF bookmark for it.

The figures show the PDF bookmark and the document table of contents side by side for purposes of comparison. Notice the automatically created Tables bookmark. However, there is no bookmark for the List of Figures shown in the

TOC. If `<figure>` is declared in the DCF as a division and its title or caption is declared as a title, then a bookmark is created for each figure as it occurs in the document structure.

Figure 298 `<Title>` elements and pseudo-element generates PDF bookmarks



DTD fragment

```
<!ELEMENT document (title,front,body,index)
<!ELEMENT body (chapter+)
<!ELEMENT chapter (title,...)
<!ELEMENT front (... ,toc,...)
<!ELEMENT index EMPTY
<!ELEMENT toc EMPTY>...
```

DCF fragment

```
<ElementOptions>
<ElementOption category="division" element="document"/>
<ElementOption category="division" element="index"/>
<ElementOption category="division" element="chapter"/>
<ElementOption category="title" element="title"/>
</ElementOptions>
```

XML fragment

```
<document><title>World Wide Widgets</title>
<front>...<toc/>...</front>
<body>
<chapter><title>Introduction</title>
...
<figure>
<graphic .../>
<caption>Top-Selling Widget</caption>
```

```

</figure>
...
</chapter>
<chapter><title>Discussion</title>
...
<table><title>Widget Specifications</title>...</table>
...
</chapter>
<chapter><title>Conclusion</title>...</chapter>
</body>
<index/>
</document>

```

FOSI fragment

```

<stringdecl textid="caption.txt" hotlink="1">
<stringdecl textid="document-title.txt" hotlink="1">
<stringdecl textid="index.app">
<stringdecl textid="lof.app" status="1" hotlink="1">
<stringdecl textid="lot.app" status="1" hotlink="1"><
<stringdecl textid="chapter-title.txt" hotlink="1">
<stringdecl textid="table-title.txt" hotlink="1">
<stringdecl textid="toc.app" status="1" hotlink="1">
...
<e-i-c gi="caption" context="figure">
<charlist inherit="1" charsubsetref="caption">
<savetext textid="caption.txt" conrule="#CONTENT">
<savetext textid="lof.app" append="1"
conrule="<toc.fmt>,caption.txt,\, \,folioct.txt[BO],</toc.fmt>">
...
<e-i-c gi="index">
<charlist inherit="1">
<usetext source="<title>,\Index\,</title>">
<subchars charsubsetref="title next-page"></subchars>
</usetext>
<usetext source="index.app" userule="2"></usetext>
...
<e-i-c gi="title" context="document">
<charlist inherit="1" charsubsetref="document-title">
<usetext source="\Contents\" placemnt="after">
<subchars charsubsetref="title allcaps"></subchars>
</usetext>
<usetext source="toc.app" placemnt="after"></usetext>
<usetext source="\List of Figures\" placemnt="after">
<subchars charsubsetref="title allcaps"></subchars>
</usetext>
<usetext source="lof.app" placemnt="after"></usetext>
<usetext source="\List of Tables\" placemnt="after">
<subchars charsubsetref="title allcaps"></subchars>
</usetext>
<usetext source="lot.app" placemnt="after">
...

```

```

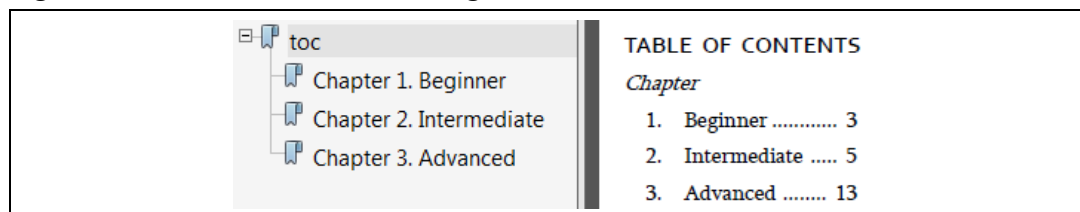
<e-i-c gi="title" context="index">
<charlist inherit="1" charsubsetref="title">
<savetext textid="toc.app" append="1"
conrule="<toc.fmt>,\Index, \, folioct.txt [BO], </toc.fmt>">
...
<e-i-c gi="title" context="chapter">
<charlist inherit="1" charsubsetref="title">
<savetext textid="chapter-title.txt" conrule="#CONTENT">
<savetext textid="toc.app" append="1"
conrule="<toc.fmt>,chapter-title.txt, \, \, folioct.txt [BO], </toc.fmt>">
<savetext textid="index.app" append="1"
conrule="<ixpt>,chapter-title.txt, </ixpt>, <ixpgstr>, folioct.txt [BO],
</ixpgstr>, <ixpgkey>, folioct.txt [BO], </ixpgkey>">
</charlist>
<att>
<fillval attname="id" attloc="chapter"
fillcat="savetext" fillchar="textid">
<charsubset>
<savetext conrule="chapter-title.txt" xrefidtag="chapter">
...
<e-i-c gi="title" context="table">
<charlist inherit="1" charsubsetref="title">
<savetext textid="table-title.txt" conrule="#CONTENT">
<savetext textid="lot.app" append="1"
conrule="<toc.fmt>,table-title.txt, \, \, folioct.txt [BO], </toc.fmt>">
...

```

The next three figures use the same DTD, DCF, and XML fragments. Also, the counter and stringdecl categories in the FOSI fragment in **Figure 299** apply to the other FOSI fragments.

The first graphic shows gentext and title content as bookmarks.

Figure 299 PDF bookmark with gentext and content



DTD fragment

```
<!ELEMENT chapter (title,...)>
```

DCF fragment

```

<ElementOptions>
<ElementOption category="division" element="chapter"/>

```

```
<ElementOption category="title" element="title"/>
</ElementOptions>
```

XML fragment

```
<chapter><title>Beginner</title>...</chapter>
<chapter><title>Intermediate</title>...</chapter>
<chapter><title>Advanced</title>...</chapter>
```

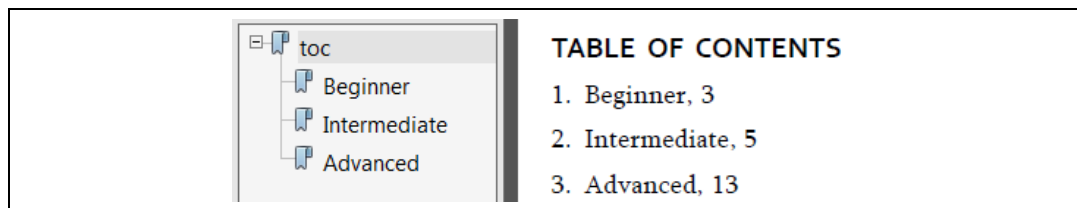
FOSI fragment

```
<counter initial="0" style="arabic" enumid="chapterct">
<stringdecl textid="chapterct.txt" hotlink="1">
<stringdecl textid="chapter-title.txt" hotlink="1">
<stringdecl textid="toc.app" status="1" hotlink="1">
...
<e-i-c gi="chapter">
<charlist inherit="1" charsubsetref="block">
<textbrk startpg="recto">
<enumerat increm="1" enumid="chapterct">
<savetext textid="chapterct.txt" conrule="chapterct">
...
<e-i-c gi="title" context="chapter">
<charlist inherit="1" charsubsetref="title endline">
<usetext source="\Chapter \,chapterct.txt,\. \"></usetext>
<savetext textid="chapter-title.txt"
conrule="chapterct.txt,\. \,#CONTENT">
<savetext textid="toc.app" append="1"
conrule="<toc.fmt>,chapter-title.txt,\, \,foliocr.txt[BO],</toc.fmt>">
...

```

In the following example, the gentext is output from the division element (<chapter>), so it does not appear in the bookmarks.

Figure 300 PDF bookmark without gentext

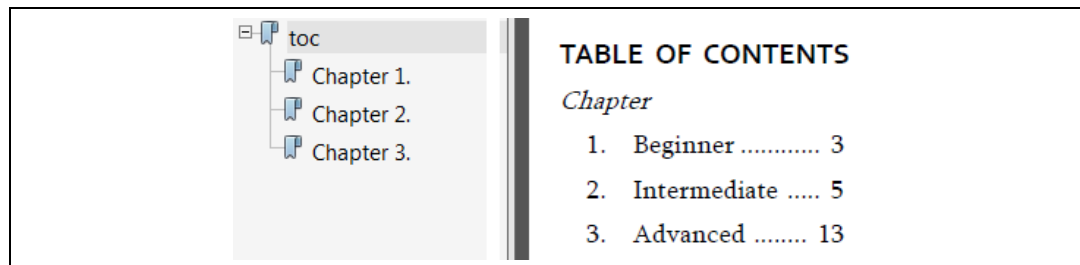


```
<e-i-c gi="chapter">
<charlist inherit="1" charsubsetref="block">
<enumerat increm="1" enumid="sectionct">
<savetext textid="chapterct.txt" conrule="chapterct">
<usetext source="chapterct.txt,\. \"></usetext>
...
<e-i-c gi="title" context="chapter">
<charlist inherit="1" charsubsetref="title endline">
```

```
<savetext textid="chapter-title.txt" conrule="chapterct.txt,\. \, #CONTENT">
<savetext textid="toc.app" append="1"
conrule="<toc.fmt>,chapter-title.txt,\, \, folioct.txt[BO],</toc.fmt>">
...
```

In the last example, the <title> is suppressed and doesn't appear in the bookmarks. However, the gentext is used.

Figure 301 PDF bookmark with only gentext



```
<e-i-c gi="title" context="chapter">
<charlist inherit="1" charsubsetref="title endline SUPPRESS">
<savetext textid="chapter-title.txt" conrule="#CONTENT">
<savetext textid="toc.app" append="1"
conrule="<toc.fmt>,1em,chapterct.txt,\.\,1em,chapter-title.txt,
dotfill,folioct.txt[BO],</toc.fmt>">
<usetext source="\Chapter \,chapterct.txt,\. \"></usetext>
...
<e-i-c gi="chapter">
<charlist inherit="1" charsubsetref="block">
<enumerat increm="1" enumid="chapterct">
<savetext textid="chapterct.txt" conrule="chapterct">
...
```

FOSI-based PDF bookmarks

A FOSI can be coded to output <atidmd> namespace markup configured to generate PDF bookmarks. <Atidmd> namespace markup is detailed in **PDF customization namespace** on page 739.

NOTE: When gentext tags are displayed in the Edit window, as shown in **Figure 302**, the <_gtlink> processing instructions that support gentext hyperlinks and PDF links are visible. More information about <_gtlink> is available at <_gtlink> **PI** on page 685.

TIP

Set `gentexttagdisplay=full` displays <atidmd> tags in the Edit window.

FOSI-based bookmark examples

Closed and open bookmarks are shown in **PDF customization example** on page 741.

Figure 302 illustrates how <atidmd> markup can be formatted for Edit window display. Notice the <_gtlink> PIs, which support gentext and PDF linking.

Figure 302 Formatted <atidmd> markup in Edit window

The screenshot shows a sequence of nested boxes representing the document structure. The top-level box is <atidmd:DocumentMetaData>. Below it are <atidmd:DocView fit="fitPage" mode="fullscreen"> and <atidmd:Outline>. The main content is a series of <atidmd:Bookmark> elements, each containing <atidmd:Title font="bold"> followed by a title (e.g., "Fruits and Vegetables", "Chapter 1.", "Apricots", "Description", "Cultivation", "Chapter 2.", "Kumquats", "Chapter 3.", "Mangos", "Chapter 4.", "Broccoli", "Chapter 5.", "Brussels sprouts", "Chapter 6.", "Lettuce", "Chapter 7.", "Spinach", "Cultivation") and <_gtlink> elements. The <_gtlink> elements are shown as small boxes with arrows pointing right. The bottom-most box is <atidmd:DocumentMetaData> with <document> below it.

FOSI fragment

```
<e-i-c gi="atidmd:Bookmark">
<charlist inherit="1" charsubsetref="startline"></charlist>
</e-i-c>
<<e-i-c gi="atidmd:DocumentMetaData">
<charlist inherit="1" charsubsetref="gray">
...
```

PDF document properties

The DocumentMetaData namespace tag `<atidmd:DocInfo>` is used to specify PDF document properties. It can be coded in a `usetext` for output anywhere in a document, either with other child elements of `<atidmd:DocumentMetaData>` or separately, as shown in **Figure 303**.

PDF document properties example

In this example, `<atidmd>` markup specifies the document name, author, and date properties of the PDF file that is created. Notice `placemnt="after"` on the `<figure> e-i-c`, which is needed so the string variables have content.

Figure 303 `<Atidmd:DocInfo>` specifies PDF document properties

DTD fragment

```
<!ELEMENT document (front,body)>
<!ELEMENT front (title,author,date)>
```

FOSI fragment

```
<stringdecl textid="author.txt" hotlink="1">
<stringdecl textid="date.txt" hotlink="1">
<stringdecl textid="document-title.txt" hotlink="1">
...
<e-i-c gi="author" context="front">
<charlist inherit="1">
<savetext textid="author.txt" conrule="#CONTENT">
...
<e-i-c gi="date" context="front">
<charlist inherit="1">
<savetext textid="date.txt" conrule="#CONTENT">
...
<e-i-c gi="front">
<charlist inherit="1">
<usetext placemnt="after" source="!<atidmd:DocumentMetaData
bookmarks="stylesheet" source="atend"><atidmd:DocInfo><atidmd:Entry>
<atidmd:Key>Title</atidmd:Key><atidmd:Value>!,document-title.txt,
!</atidmd:Value></atidmd:Entry><atidmd:Entry><atidmd:Key>Author
</atidmd:Key><atidmd:Value>!,author.txt,!</atidmd:Value></atidmd:Entry>
<atidmd:Entry><atidmd:Key>Date</atidmd:Key><atidmd:Value>!,date.txt,
!</atidmd:Value></atidmd:Entry></atidmd:DocInfo></atidmd:DocumentMetaData>
...
<e-i-c gi="title" context="document">
<charlist inherit="1" charsubsetref="title center">
<savetext textid="document-title.txt" conrule="#CONTENT">
...
```


PDF viewer settings

The DocumentMetaData namespace tag `<atidmd:DocView>` is used to specify PDF window customizations. It can be coded in a `usetext` for output anywhere in a document, separately, as shown in **Figure 303**, or with other child elements of `<atidmd:DocumentMetaData>`.

PDF window customization example

In this example, `<atidmd:DocView>` specifies that the document should be scaled to fit within the width of the PDF viewer window and that the content with the named destination of “openpage” should appear at the top of the window.

Figure 304 `<Atidmd:DocView>` specifies PDF window properties

XML fragment

```
<chapter id="openingpage"><title>Latest Update</title>...
```

FOSI fragment

```
<usetext source='!<atidmd:DocumentMetaData bookmarks="stylesheet"
source="atend"><atidmd:DocView fit="fitWidth" destination="openpage"/>
</atidmd:DocumentMetaData>!'>
```

XML fragment

```
<title>The <?Pub _font TypeSize="20pt" FamName="Castellar"?>Loch Ness
Monster<?Pub /_font?>: <?Pub _touchup TextbrkStartln="yes"?>Fact or
Fiction?<?Pub /_touchup?></title><para>Sitings of the <?Pub _font
FamName="Arial"?>Loch Ness Monster<?Pub /_font?> are few and far between...
```

FOSI fragment

```
<e-i-c gi="touchup" gitype="pi">
<charlist inherit="1"></charlist>
</e-i-c>
<e-i-c gi="font" gitype="pi">
<charlist inherit="1">
<font famname="Chiller" size="20pt">
...
```

Link and target processing instructions

Users can create hyperlinks by inserting `<_link>` and `<_target>` processing instructions in documents. As discussed below, some formatting is supported for these PIs.

Also see `<_link>` and `<_target>` PIs for PDF links on page 545.

A `<_link>` is inserted by highlighting the relevant content in the Edit window and selecting Tools→Create Hyperlink.... A dialog then directs the user to click in the document location where the `<_target>` PI should be inserted. Refer to Arbortext Editor documentation for details.

NOTE: `<_link>` and `<_target>` are being deprecated in Arbortext Editor but are still supported.

The `<_link>` PI has several attributes, as detailed in the following tables:

Table 99 <_link> Link Control attributes

Attribute	Values	Default	Notes
Command	CDATA	—	The command is executed when the NOTE: When a command is specified for <link>, any command specified for <_target> is ignored. The <_target> command is executed only when no command is specified for <link> .
TargetTag	CDATA	—	Clicking on the <link> text in a PDF viewer causes the <_target> text to be positioned at the top of the viewer window.
TargetDoc	CDATA	—	This does not work in PDF. Clicking on <link> in the Edit window changes the display to raw PDF code.
Print	<u>yes</u> no	no	When set to <code>no</code> , the contents are suppressed from output. The default defined in <code>atitag.cf</code> is <code>no</code> , which can be seen by entering <code>it _link</code> at the command line and opening its dialog to see the attribute defaults. However, when <link> is inserted with Tools→Create Hyperlink..., this attribute is set to <code>yes</code> .
Underline	<u>yes</u> no	yes	Controls underlining in the Edit window but does not affect formatted output.

Table 100 <_link> Font attributes

Attribute	Values	Default	Notes
FontStyle	†	inherit	—
TypeSize	†	inherit	—
Posture	†	inherit	—
Weight	†	inherit	—
FamName	†	—	—
FontColor	COLOR	inherit	The default defined in <code>atitag.cf</code> is <code>inherit</code> , which can be seen by entering <code>it _link</code> at the command line and opening its dialog to see the attribute defaults. However, when <link> is inserted with Tools→Create Hyperlink..., the FontColor is set to <code>green</code> .
BackColor	COLOR	inherit	—
†The same as the values for the equivalent <code>font</code> characteristic.			

In addition, ACL can be used to set several other attributes, as described in **Link and target-related ACL** on page 682.

KEY

Monospaced fonts are used for user input and FOSI categories, characteristics, and attributes.

In OutSpec DTD fragments, a single underscore means support only for print/PDF output. A double underscore indicates both Edit window and print/PDF support.

NOTE: The Edit window shows `<_link>` PI formatting attributes such as the default green font, but formatted output show FOSI coding for the PI.

The `<_link>` PI can be formatted by one or more `e-i-cs`. However, while attspecs are recognized, repeating categories are ignored.

NOTE: An `e-i-c` for these PIs should be coded with `gitype="element"`.

The `<_target>` PI has one attribute, which is described in the following table..

Table 101 `<_target>` attribute

Attribute	Values	Default	Notes
Command	CDATA	—	The command is executed if the corresponding <code><_link></code> does not specify a command to execute.

Link and target-related ACL

In addition to those available in the Modify Attributes dialog box, several other attributes can be set using ACL:

- `MenuLabel`, which controls what appears in the Link menu item on the Edit window popup menu. This attribute does not display in Arbortext Editor when the `<_link>` PI is displayed.
- `Func`, which names an ACL function to perform the linking action. If `Func` is set, all other link-governing attributes are ignored.
- `Attr1`, `attr2`, and `attr3`, which are provided for use by custom applications.

`sSet hyperlinkmenus=on` adds Create Hyperlink, Delete Hyperlink, and Show Hyperlinks to the Tools menu. The default is `off` because these menu items have been replaced with Insert→Link and Insert→Link Target.

KEY

Monospaced fonts are used for user input and FOSI categories, characteristics, and attributes.

In OutSpec DTD fragments, a single underscore means support only for print/PDF output. A double underscore indicates both Edit window and print/PDF support.

Link and target examples

The formatting specified in the `<_link>` attributes applies only to the Edit window. The formatting specified in the `e-i-c` applies to the PDF output.

Figure 370

Refer to [End Notes](#).

XML fragment

```
<para>...Refer to <?Pub __link TargetTag="__target_2" Print="yes"
Weight="bold" FontColor="green"?>End Notes.<?Pub /__link?></para>
...
<appendix><title><?Pub __target_2?>End Notes</title>...</appendix>
```

FOSI fragment

```
<e-i-c gi="__link">
<charlist inherit="1" charsubsetref="blue underline"></charlist>
</e-i-c>
```

User-defined tag processing instructions

A user-defined tag (UDT) is an alias for any of the following:

- an element
- an Arbortext Editor processing instruction
- another UDT

The markup for these differs, as shown in the **User-defined tags examples** on page 684

A UDT has the same attributes as the base element or PI, which can be set in the usual way.

A user-defined tag is actually a processing instruction that is created with an ACL command, which is described in the next section.

If the FOSI contains an `e-i-c` for the base tag, that `e-i-c` will also match the UDT. See **Figure 371 User-defined tag formatted with e-i-c** on page 684 for an example.

User-defined tags-related ACL

- `Define_tag [newtagname oldtagname]` creates UDTs. `dft` is a synonym for `define_tag`.
Newtagname may contain any combination of letters, numbers, dashes (-), underscores (_), and periods (.). If an alias map has been applied to the document, oldtagname can be an alias or a real name.
If a tag name is not specified, a Tag Definition dialog is displayed, which provides fields for entering the names. It also includes a checkbox to indicate that the tag before the cursor in the Edit window is the oldtagname.

TIP 

UDTs are not found by Find→Processing Instruction. Instead, enter `show usertags` (described below) at the command line.

When `oldtagname` is an Arbortext Editor PI or another UDT, the checkbox is replaced by three choices:

- ▶ other, which requires that the `oldtagname` be entered in the first field
- ▶ the name of the tag before the cursor
- ▶ the name of the tag before the cursor “with local mods,” meaning that any settings on `oldtagname` are also set on `newtagname`

In an SGML/XML file, UDT declarations appear before the first document element. For instance, `<?Pub UDT tip note?>` indicates the `<tip>` UDT is based on the `<note>` tag.

- `Show usertags [output=filename]` displays user-defined tags. If an alias map has been applied to the document, `show usertags` displays aliases for tags that have been assigned aliases in the Original Tag column. If `output` is specified, a list of user-defined tags is written to `filename`, which can be a complete path name. A right angle bracket (`>`) preceding `filename` causes the list of aliases to be appended to the end of the file.
- `User_tag_names(arr[, doc])` is a function that discards any previous elements in `arr`, fills it with a list of user-defined tags for the current document type, and returns the number of tags. The first tag is stored at index `[1]`. `Doc` specifies the identifier of the document tree to query. If omitted or `0`, the current document is used.
- `Undefine_tag tagname` removes the previously defined user tag `tagname`. `udft` is a synonym for `undefine_tag`.

NOTE: `Show usertags` and `user_tag_names` show FOSI pseudo-elements, notations, text entities, file entities, bookmarks, and Touchup tags.

User-defined tags examples

In the first example, the UDT is defined and formatted with an `e-i-c` in the FOSI. Notice the markup in the `.xml` file.

Figure 371 User-defined tag formatted with `e-i-c`

DTD fragment

```
<!ELEMENT pn (#PCDATA)>
```

ACL fragment

```
dft partnbr pn
```

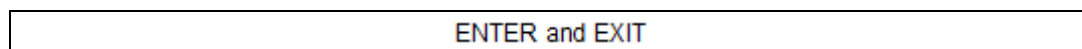
XML fragment

```
...<partnbr><?Pub (partnbr)?>IO-1234560</partnbr>...
```

FOSI fragment

```
<e-i-c gi="pn">
<charlist inherit="1">
<font inherit="1" famname="Lucida Sans">
</charlist>
</e-i-c>
```

In the last example, the UDT `allcaps` has been created based on a Touchup PI with local mods. Notice that the markup in the `.xml` file is not the same as for a UDT of an element.

Figure 372 User-defined tag based on <_touchup>**XML fragment**

```
<paragraph><?Pub _touchup HighlAllcap="on"?>enter<?Pub /_touchup?>
and <?Pub Tag allcaps?>enter<?Pub /allcaps?></paragraph>
```

Internal user-defined tags

Two processing instructions used by the formatter may have default `e-i-cs` in the distribute files and may have `e-i-cs` coded in a FOSI to override the default settings. They are described in this section.

- ▷ `<_gtlink>` PI, page 685
- ▷ `<_display>` PI, page 689

`<_gtlink>` PI

The `<_gtlink>` processing instruction is a tag pair generated by the formatter to support gentext hyperlinks and PDF links. `<_gtlink>` has a read-only *linkto* attribute, which contains the object identifier (oid) for the element whose content was saved.

NOTE: `<_gtlink>` is not saved with the document.
Hyperlinks and their targets cannot be used as DLM resources.

Table 102 shows the `<_gtlink>` attribute:

Table 102 <_glink>

Attribute	Tag	Values	Default	Notes
linkto	pair	OID	—	Surrounds the savetext conrule content.

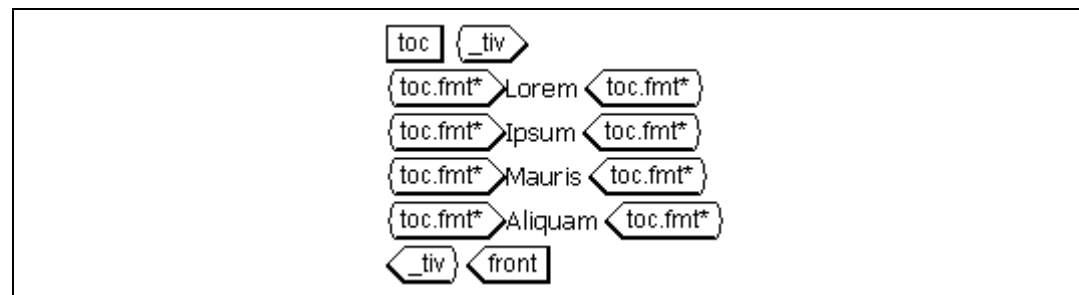
The default *e-i-c* for <_glink> is located in *Arbortext-path\lib_glink-eic.fos* and reproduced here. Notice that it specifies underscoring as the default formatting in the Edit window. <_glink> can have *e-i-cs* in a FOSI with *gitype="pi"* to override the default and/or specify formatting for print/PDF output.

Figure 373 Default formatting for <_glink> PI

```
<!-- $Id: $ FOSI fragment supporting _glink ATI tag-->
<!-- default: underline all glinks on the screen -->
<e-i-c gi="glink" gitype="pi">
<charlist inherit="1"></charlist>
<att>
<specval attname="editor-only" attloc="system-var" attval="#ANY">
<charsubset>
<highlt inherit="1" scoring="1" scorespc="1">
</charsubset>
</att>
</e-i-c>
```

<_glink> example

Figure 375 shows a FOSI *e-i-c* for <_glink>. For purposes of comparison, **Figure 374 TOC gentext in Edit window — no links** shows a generated table of contents in the Edit window with *gentext* tags displayed but no “hot links”. In **Figure 375 TOC gentext with links in Edit window** on page 688, the <_glink> processing instruction is displayed, and an *e-i-c* for it specifies a gray background.

Figure 374 TOC gentext in Edit window — no links

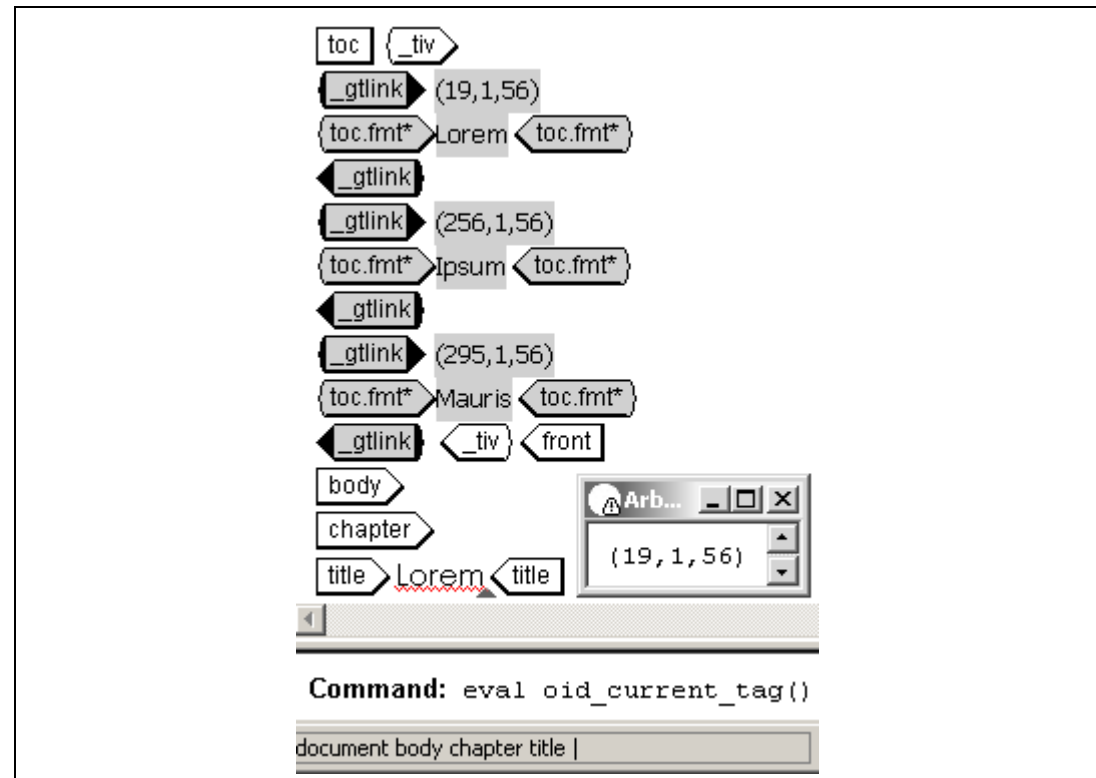
XML fragment

```
<toc/>
...
<chapter><title>Lorem</title>...
<chapter><title>Ipsum</title>...
<chapter><title>Mauris</title>...
<chapter><title>Aliquam</title>...
```

FOSI fragment

```
<stringdecl textid="toc.app" status="1" hotlink="0">
...
<e-i-c gi="title" context="chapter">
<charlist inherit="1" charsubsetref="title">
<savetext textid="toc.app" append="1"
conrule="<toc.fmt>,#CONTENT",</toc.fmt>">
...
<e-i-c gi="toc">
<charlist inherit="1" charsubsetref="startline">
<usetext source="toc.app">
...
<e-i-c gi="toc.fmt">
<charlist inherit="1" charsubsetref="block">
...
```

Figure 375 TOC gentext with links in Edit window

**ACL command**

```
eval oid_current_tag()
```

FOSI fragment

```
<stringdecl textid="toc.app" status="1" hotlink="1">
...
<e-i-c gi="gtlink" gitype="pi">
<charlist inherit="1" charsubsetref="startline">
<highlt inherit="1" bckclr="gray">
...
```

PDF customization namespace

Document metadata markup, `<atidmd>`, is provided to enable customization of PDF output, including:

- bookmarks
- PDF properties
- PDF viewer window settings

Document metadata markup is in the namespace defined by the URL `http://www.arbortext.com/namespace/DocumentMetaData/`.

The DTD fragment at `http://www.arbortext.com/namespace/DocumentMetaData/atidmd-dtd.txt` is reproduced here.

Figure 390 Atidmd DTD

```
<!------->
<!--
  Arbortext DocumentMetaData DTD
  Copyright (C) 2003, 2004 Arbortext, Inc.

  DocumentMetaData doctype.
  Represents Document Meta Data
  for inclusion in document as pseudo tags.

  This DTD defines the vocabulary specified by the namespace name:

  http://www.arbortext.com/namespace/DocumentMetaData

  =====>

<!ELEMENT DocumentMetaData ( DocView?, DocInfo?, Outline? ) >
<!-- The source attribute provides a mechanism for placing a
  DocumentMetaData tag at the top of the document which will
  serve to disable automatic bookmarks. The DMD block at the
  top of the document would have source="atend", and then the
  one at the end would have source="local" (the default).
  Alternatively, the DMD block can simply be placed at the
  beginning of the document with source="local".

  The bookmarks attribute provides a mechanism for including
  DocView or DocInfo data while still using the automatic bookmark
  method. bookmarks="auto" will cause the TeX write to output
  the old-style automatic bookmark specials. If the stylesheet
  adds bookmarks as well, the results could be unpredictable. -->
<!ATTLIST DocumentMetaData
```

TIP

`<Atidmd>` markup displays in the Edit window when `set gentexttagdisplay=full`. The tags can be formatted with `e-i-cs`, if desired, as shown in **Figure 302**. This formatting does not affect PDF bookmarks or printed output.

```

        source ( atend | local ) "local"
        bookmarks ( none | auto | stylesheet ) "stylesheet" >

<!--=====-->
<!-- -->
<!-- DocView:  controls how the document appears when opened. -->
<!-- -->
<!-- mode: -->
<!-- none:      No tabs selected -->
<!-- bookmarks: Bookmarks visible -->
<!-- thumbs:   Thumbnails visible -->
<!-- fullScreen: Full screen mode -->
<!-- -->
<!-- fit: -->
<!-- actualSize: Zoom=100% -->
<!-- fitPage:   Entire page within window -->
<!-- fitWidth:  Fit width to window -->
<!-- -->
<!-- destination: FOSI:   Epic will generate -->
<!--               XSL-FO: Use internal-destination -->
<!--               Default: Page 1 -->

<!ELEMENT DocView EMPTY >
<!ATTLIST DocView
    mode ( none | bookmarks | thumbs | fullScreen ) "none"
    fit ( actualSize | fitPage | fitWidth ) "fitWidth"
    destination CDATA #IMPLIED >

<!--=====-->

<!--=====-->
<!-- -->
<!-- DocInfo:  Sets Document Properties parameters. -->
<!-- -->
<!ELEMENT DocInfo ( Entry* ) >

<!ELEMENT Entry ( Key, Value ) >

<!-- key ( e.g., title, author, subject, keywords ) -->
<!-- -->
<!ELEMENT Key ( #PCDATA ) >

<!ELEMENT Value ( #PCDATA ) >

<!--=====-->

<!--=====-->
<!-- -->
<!-- Outline:  Defines bookmark structure -->
<!-- -->
<!ELEMENT Outline ( Bookmark* ) >

```

```

<!-- Bookmark -->
<!-- -->
<!-- destination: FOSI: Epic will generate -->
<!-- XSL-FO: stylesheet must supply -->
<!-- If this starts with '#', it will be -->
<!-- interpreted as an internal destination; -->
<!-- otherwise, it is an external file. -->
<!-- -->
<!-- state: -->
<!-- open: children are shown -->
<!-- closed: children are not shown -->
<!-- hidden: will not appear as a bookmark; -->
<!-- will be open (children show) -->
<!-- -->
<!-- format: font style for title -->
<!-- -->
<!-- color: Epic color name or RGB value in Hex (#hhhhh) -->
<!-- Default: black (#000000) -->
<!-- -->
<!-- No Title element: treat bookmark as hidden -->
<!-- -->
<!ELEMENT Bookmark ( Title?, Bookmark* ) >
<!ATTLIST Bookmark
  destination CDATA #IMPLIED
  state ( open | closed | hidden ) "open" >

<!ELEMENT Title (#PCDATA) >
<!ATTLIST Title
  font ( regular | bold | italic | bolditalic ) "regular"
  color CDATA #IMPLIED >

<!--=====>

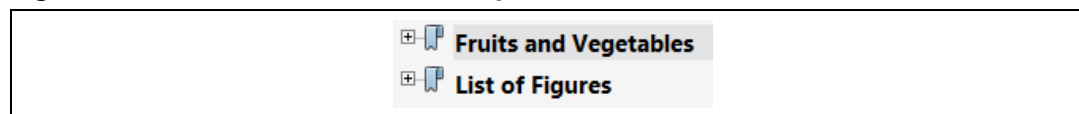
```

The `atend` setting avoids the need for a time-independent variable. Like titles for a table of contents, titles for bookmarks are not available at the beginning of a document during the first formatting pass.

See **FOSI-based bookmark examples** on page 552.

PDF customization example

The graphics in **Figure 391** and **Figure 392** show closed and open bookmarks, respectively. Notice the nested `<atidmd:Bookmark>` elements.

Figure 391 <Atidmd:BookMark> specifies closed PDF bookmarks**XML fragment**

```

<document><front>...</front>
<body>
<chapter><title>Apricots</title>
<section><title>Description</title>...</section>
<section><title>Cultivation</title>...</section>
</chapter>
<chapter><title>Kumquats</title>
<section><title>Description</title>...</section>
<section><title>Cultivation</title>...</section>
</chapter>
<chapter><title>Mangos</title>
<section><title>Description</title>...</section>
<section><title>Cultivation</title>...</section>
</chapter>
<chapter><title>Broccoli</title>
<section><title>Description</title>...</section>
<section><title>Cultivation</title>...</section>
</chapter>
<chapter><title>Brussels sprouts</title>
<section><title>Description</title>...</section>
<section><title>Cultivation</title>...</section>
</chapter>
<chapter><title>Lettuce</title>
<section><title>Description</title>...</section>
<section><title>Cultivation</title>...</section>
</chapter>
<chapter><title>Spinach</title>
<section><title>Description</title>...</section>
<section><title>Cultivation</title>...</section>
</chapter></body></document>

```

FOSI fragment

```

<counter initial="0" style="arabic" enumid="chapterct">
<counter style="arabic" enumid="figurect">
<stringdecl textid="chapter-bookmarks.app" hotlink="1">
<stringdecl textid="chapter-title.txt" hotlink="1">
<stringdecl textid="document-title.txt" hotlink="1">
<stringdecl textid="figure-bookmarks.app" hotlink="1">
<stringdecl textid="figurect.txt" hotlink="1">
<stringdecl textid="figure-title.txt" hotlink="1">
<stringdecl textid="section-bookmarks.app" hotlink="1">
<stringdecl textid="section-title.txt" hotlink="1">
...

```

```

<e-i-c gi="chapter">
<charlist inherit="1">
<enumerat increm="1" enumid="chapterct">
<savetext textid="chapterct.txt" conrule="chapterct">
<savetext textid="chapter-title.txt" conrule="\">
<savetext textid="section-bookmarks.app" conrule="\">
<savetext textid="chapter-bookmarks.app" placemnt="after" append="1"
conrule='!<atidmd:Bookmark state="closed"><atidmd:Title>!,\Chapter \,
chapterct.txt,\. \,chapter-title.txt,!</atidmd:Title>!'
,section-bookmarks.app,!</atidmd:Bookmark>!'>
...
<e-i-c gi="document">
<charlist inherit="1" charsubsetref="block">
<textbrk startpg="next" pageid="document.page" newpgmdl="global">
<usetext source='!<atidmd:DocumentMetaData bookmarks="stylesheet"
source="atend"
xmlns:atidmd="http://www.arbortext.com/namespace/DocumentMetaData">
</atidmd:DocumentMetaData>!'></usetext>
<usetext placemnt="after" source='!<atidmd:DocumentMetaData
bookmarks="stylesheet"
source="atend"><atidmd:Outline><atidmd:Bookmark state="closed">
<atidmd:Title
font="bold">!,document-title.txt,!</atidmd:Title>!
,chapter-bookmarks.app,!</atidmd:Bookmark><atidmd:Bookmark
state="closed"><atidmd:Title font="bold">List of Figures</atidmd:Title>
<atidmd:Bookmark>!,figure-bookmarks.app,!</atidmd:Bookmark>
</atidmd:Bookmark></atidmd:Outline></atidmd:DocumentMetaData>!'>
...
<e-i-c gi="figure">
<charlist inherit="1">
<enumerat increm="1" enumid="figurect">
<savetext textid="figurect.txt" conrule="figurect">
<savetext textid="figure-bookmarks.app" placemnt="after" append="1"
conrule='!<atidmd:Bookmark state="open"><atidmd:Title>!,\Figure \,
figurect.txt,\. \,figure-title.txt,!</atidmd:Title></atidmd:Bookmark>!'>
...
<e-i-c gi="section">
<charlist inherit="1">
<savetext textid="section-title.txt" conrule="\">
<savetext' placemnt="after" append="1" textid="section-bookmarks.app"
conrule='!<atidmd:Bookmark state="open"><atidmd:Title>!,section-title.txt,
!</atidmd:Title></atidmd:Bookmark>!'>
...
<e-i-c gi="title" context="chapter">
<charlist inherit="1" charsubsetref="title">
<savetext textid="chapter-title.txt" conrule="#CONTENT">
<usetext source="\Chapter \,chapterct.txt,\. \">
...
<e-i-c gi="title" context="document">
<charlist inherit="1" charsubsetref="title center">
<savetext textid="document-title.txt" conrule="#CONTENT">

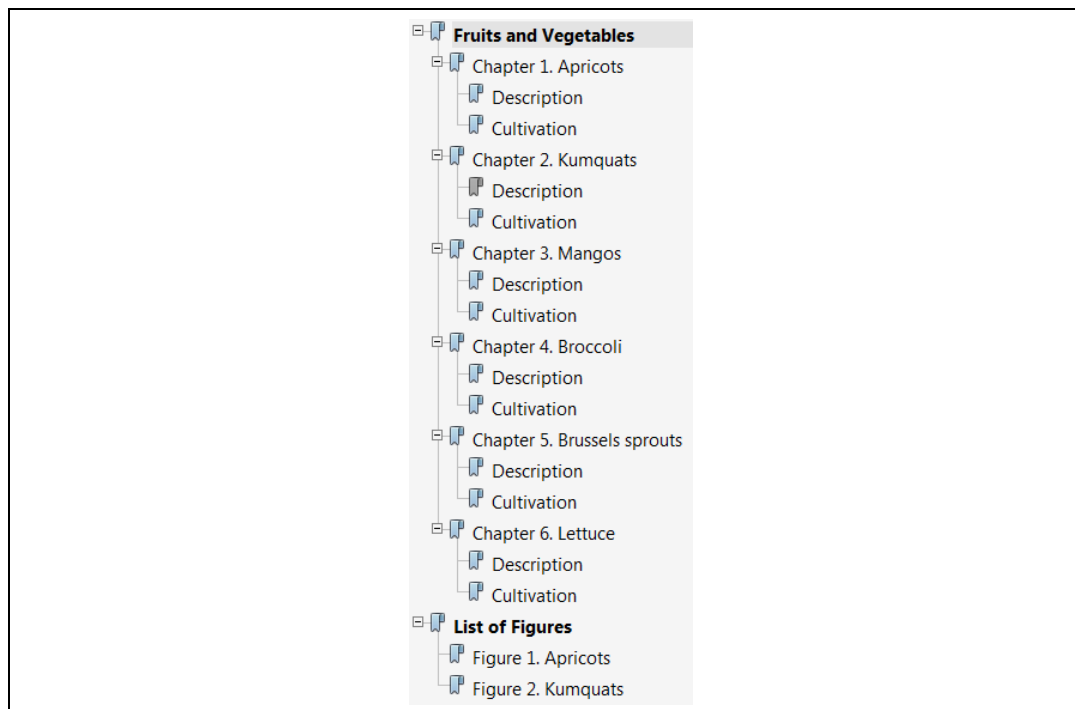
```

```

...
<e-i-c gi="title" context="figure">
<charlist inherit="1" charsubsetref="title">
<savetext textid="figure-title.txt" conrule="#CONTENT">
...
<e-i-c gi="title" context="section">
<charlist inherit="1" charsubsetref="title italic">
<savetext textid="section-title.txt" conrule="#CONTENT">
...

```

Figure 392 <Atidmd:BookMark> specifies open PDF bookmarks



FOSI fragment

```

<e-i-c gi="chapter">
...
<savetext textid="chapter-bookmarks.app" placemnt="after" append="1"
conrule='!<atidmd:Bookmark state="open"><atidmd:Title>!, \Chapter \,
chapterct.txt, \. \, chapter-title.txt, !</atidmd:Title>!,
section-bookmarks.app, !</atidmd:Bookmark>!'>
...

```

Also see **Figure 302 Formatted <atidmd> markup in Edit window** on page 552, which shows formatting for <atidmd> markup.