

DIG/Evergreen DocBook Style Guide

DIG/Evergreen DocBook Style Guide

Copyright © 2010 Evergreen Community



This document was updated 2012-05-16.

Table of Contents

| | |
|-----------------------------------------|-----------|
| <u>1. DIG DocBook reference</u> | <u>4</u> |
| <u>Elements</u> | <u>4</u> |
| <u>Attributes</u> | <u>8</u> |
| <u>2. Image Formatting</u> | <u>10</u> |
| <u>3. Creating Links</u> | <u>12</u> |
| <u>4. Processing instructions</u> | <u>13</u> |
| <u>5. Miscellaneous Tips</u> | <u>14</u> |

Chapter 1. DIG DocBook reference

Elements

Below is a non-exhaustive list of DocBook elements used in Evergreen documentation. Authors may use other elements but are encouraged to draw from this list first. Each term links to the authoritative DocBook 5 reference pages at <http://www.docbook.org/tdg5/en/html/part2.html>.

[alt](#) Metadata describing an element, usually for accessibility. For example, each mediaobject may contain an alt element with a generic description of the image content.

[application](#) The name of a software program or application such as Evergreen. Currently not formatted, but consistent use would allow for future batch formatting (e.g. italicize content of all application tags).

[book](#) The top level element for Evergreen documentation, appears only in the file *root.xml*

[chapter](#) A subdivision of a part, further divided into sections. In Evergreen documentation each chapter is a separate xml file referenced by *root.xml*. Each chapter should have an xml:id. The xml:id is required for cross-referencing and stable links to published html files. If there is no xml:id the generated html files names may change as content is added or re-located.

[code](#) An inline code fragment.

[command](#) The name of an executable program or other software command. Use this instead of emphasis where applicable.

[computeroutput](#) Computer generated output. Used with screen to differentiate output from input.

[date](#) Wrapper for date info in any context. Format is YYYY-MM-DD

[emphasis](#) Use to *italicize* regular text. Use with the ROLE attribute for **bold** text

```
<para>For <emphasis>italic</emphasis> and <emphasis role="bold">bold</emphasis> output
```

displays as

For *italic* and **bold** output.

[errortext](#) An error message.

[example](#) An example with a title.

[figure](#) Wrapper for a mediaobject that has a title and appears in list of figures (in the book's front matter). Content in figure and informalfigure is not split across PDF page breaks.

| | |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <u>filename</u> | Specify a name of a file or directory. See docbook entry for class options. |
| <u>formalpara</u> | <p>A titled paragraph. The following</p> <pre><formalpara> <title>Paragraph title</title> <para>Paragraph body...</para> </formalpara></pre> <p>displays as</p> <p>Paragraph title. Paragraph body...</p> |
| <u>function</u> | The name of a function or subroutine, as in a programming language. |
| <u>glossary</u> | Wrapper for a glossary. Must contain title element (to hide the title in published output use an empty <title/> tag). |
| <u>glossentry</u> | <p>Wrapper for each glossary entry. Should always have an xml:id for cross referencing. Entries are sorted alphabetically in published output, regardless of order in xml source document (this is optional and is set with stylesheet parameters). Full glossentry markup including child elements is shown below:</p> <pre><glossentry xml:id="guisubmenu"> <glossterm>guisubmenu</glossterm> <glossdef> <para>A staff client submenu. Used primarily when a submenu is part of a <markup>menuchoice</markup> sequence. Not italicized.</para> <glossseealso otherterm="guimenu"/> </glossdef> </glossentry></pre> |
| <u>glosssee</u> | A "see" link that points from one glossentry to another. Always has an otherterm attribute that corresponds to the xml:id of the referenced term. |
| <u>glossseealso</u> | A "see also" link that points from one glossentry to another. Always has an otherterm attribute corresponds to the xml:id of the referenced term. |
| <u>glossterm</u> | Wrapper for the term in a glossentry. |
| <u>guibutton</u> | <p>A button in the staff client. Currently formatted with italics, though this can be changed in the stylesheets.</p> <pre><para>Click <guibutton>Modify Copies</guibutton> to save your changes</para></pre> <p>displays as</p> <p>Click <i>Modify Copies</i> to save your changes</p> |
| <u>guilabel</u> | Use for labels and titles in the staff client. Where applicable guimenu, guimenuitem, and guibutton are preferred. HTML is formatted in italics, PDF is not formatted. |

| | |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <u>guimenu</u> | Use for the names of staff client menus; formatted in italics. |
| <u>guimenuitem</u> | An option in a staff client menu or submenu; formatted in italics. |
| <u>guisubmenu</u> | A staff client submenu. Used when a submenu is part of a menuchoice sequence. Not italicized. |
| <u>imagedata</u> | Contained by imageobject, its attributes determine sizing, scaling, alignment, and source file. See <i>Chapter 2, Image Formatting</i> for more on Sitka image formatting. |
| <u>imageobject</u> | Contained by mediaobject and a wrapper for imagedata. See <i>Chapter 2, Image Formatting</i> for more on image formatting in <i>The Book of Evergreen</i> . |
| <u>info</u> | A wrapper for metadata about an element, including title. |
| <u>informalexample</u> | An example with no title. |
| <u>informalfigure</u> | A wrapper for a mediaobject and optional associated text. Stylesheet customizations ensure the contents of an informfigure are kept together during PDF pagination. The step in the following example will not split across a page break: |

```

<step>
  <informalfigure>
    <para>STEP TEXT GOES HERE</para>
    <mediaobject>
      <alt>OPTIONAL DESCRIPTION OF IMAGE</alt>
      <imageobject>
        <imagedata scalefit="0" fileref="media/FILENAME_HERE.png"/>
      </imageobject>
    </mediaobject>
  </informalfigure>
</step>

```

| | |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <u>itemizedlist</u> | A bulleted list with an optional title. |
| <u>keycap</u> | The label on a keyboard button; formatted in bold (eg Press F5). |
| <u>keycap</u> | For a combination of two or more keystrokes, wrap keycap elements in a keycombo. Format is bold with a + between keystrokes. |

```

<para>Open a new tab with
  <keycombo>
    <keycap>Ctrl</keycap>
    <keycap>T</keycap>
  </keycombo>
</para>

```

displays as

Open a new tab with **Ctrl+T**

| | |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <u>link</u> | For external links. Must have an xlink:href attribute with the destination URL. In PDF output the full URL is printed in brackets after the link text. |
| <u>listitem</u> | Wrapper for individual entries on an itemizedlist or orderedlist. |

mediaobject A wrapper for imageobjects. May be wrapped in an informalfigure to keep image and text together across page breaks. See [Chapter 2, Image Formatting](#) for more on image formatting.

menuchoice Use to describe a menu selection. Usually contains guimenu and guimenuitem elements (may also contain guisubmenu). Output displays an arrow between each child element.

```
<para>To clear the staff client cache go to
  <menuchoice>
    <guimenu>Admin</guimenu>
    <guisubmenu>For developers</guisubmenu>
    <guisubmenuitem>Clear Cache</guisubmenuitem>
  </menuchoice>
</para>
```

displays as

To clear the staff client cache go to Admin → For developers → Clear Cache

option Specify an option for a software command. Use this instead of emphasis where applicable.

orderedlist A numbered list with an optional title.

para A paragraph; a required wrapper for text in most contexts.

part A subdivision of a book. Used only in *root.xml*; each part should have a title and xml:id.

procedure A multi-step process consisting of two or more steps. Applies to most converted Sitka docs.
See Also step.

programlisting When used with CDATA, text in a programlisting is published exactly as it appears in the xml file (including line breaks and white space). Formatted as a block with a shaded background, used only for coding examples (see */style/style_glossary.xml* for syntax example).

prompt A character or string indicating the start of an input field in a computer display. Used with screen to indicate a prompt for user input.

screen Text that a user sees or might see on a computer screen

section A titled subdivision of a chapter, only required if a chapter has two or more sections. Sections appear in TOC and have their own pages in HTML output (use simplesect to exclude from TOC). The xml:id attribute is optional and can be applied ad hoc when linking to a section.

The section element can be nested to create subsections. Here are some examples used in *The Book of Evergreen*:

- *section* -> *section* (for a section and subsection, both appearing in TOC)

- *section* -> *simplesect* (for a section and subsection, w/ only the top level in the TOC)
- *section* -> *section* -> *simplesect* (for a section, subsection, and sub-subsection, with the top 2 in the TOC))

simplesect

A subdivision of a section that does not appear in the TOC. A simplesect is terminal, i.e. cannot be further subdivided with section or simplesect elements.

step

Component part of a procedure. Steps are auto-numbered and may contain text, figures, examples, etc. The xml:id attribute is optional and only required if linking to a particular step. To keep step content together during pagination (eg text w/ associated image), use informalfigure.

systemitem

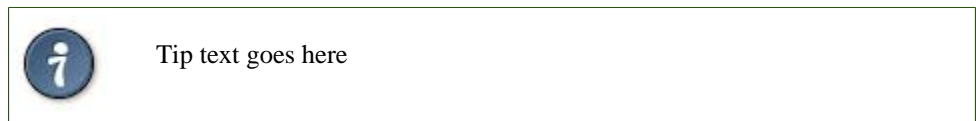
Indicates various system related resources and items. See the Docbook entry for classification possibilities. Use this in place of emphasis where applicable.

tip

Enclose tip text and images in this element for consistent formatting. Display options are set with CSS and XSL stylesheets and include custom tip text, icons, formatting, etc.

```
<tip>
  <para>Tip text goes here</para>
</tip>
```

displays as



title

Sets the title for the element which contains it. Required in chapters and sections.

userinput

Data entered by the user. Used with screen to differentiate input from output.

varname

The name of a variable. Use this instead of emphasis where applicable.

xref

A cross reference to another part of the document (use link for external links). The destination is set with the linkend attribute, which matches the xml:id of the destination element. Link text is auto-generated.

oXygen returns a validation error if the link destination is in another xml file; this is resolved when the files are assembled for publication and can be ignored.

Attributes

DocBook elements can be modified by one or more attributes. The attributes below are commonly used in *The Book of Evergreen*. More info on DocBook 5 attributes is available at <http://www.docbook.org/tdg5/en/html/ref-elements.html#common.attributes>

Attributes

linkend

Attribute that sets the link destination. Any DocBook 5 inline element can become a link by adding a linkend attribute that corresponds to the xml:id of the destination. See [Chapter 3, *Creating Links*](#) for more detailed info.

otherterm

Applied exclusively to glossee and glosseealso elements, it points to the referenced glossary term.

```
<glosseealso otherterm="date"/>
```

displays as

See also [date](#).

role

Para. Applied to para tags to track changes to draft docs for approval by other editors. Can have value of *added*, *deleted*, or *changed*; in html output will be highlighted in green, red, and yellow respectively.

xml:id

Optional xml identifier for an element, required for cross references and linking. Must be unique within the entire documentation set. Good practice is to make sure that, at minimum, all book, part, chapter, and glossentry elements have an xml:id. In html output, the xml:id becomes the name the html file and the URL for that segment of the documentation (if there is no xml:id, the html file is auto-numbered).

Chapter 2. Image Formatting

- .png is the preferred image format, though .jpg from legacy conversions are acceptable
- Images are stored in *media* directories for each part.
- Images up to 900px wide can be displayed without resizing. Images larger than 900px are scaled down with the *scalefit="1"* attribute as shown below.
- When scaling DocBook uses both the size (pixels) and resolution (dpi or pixels/in) of image files. Source images should be at 96dpi, a default chosen to match legacy images. 96dpi seems to be standard in FullShot; if using Gimp change the default for new images from 72 pixels/in to 96 pixels/in under File → Preferences → Default Image

Images 900px wide or less

If the source image is 900px wide or less use the following code

```
<mediaobject>
  <alt>OPTIONAL DESCRIPTION OF IMAGE</alt>
  <imageobject>
    <imagedata scalefit="0" fileref="media/FILENAME.png"/>
  </imageobject>
</mediaobject>
```

In HTML output, the image will display at actual size; in PDF it will be scaled down to 75% (scale percentage set in *~/stylesheets/sitka_fo_graphics.xsl*).

Images wider than 900px

If the source image is wider than 900px use the following code

```
<mediaobject>
  <alt>OPTIONAL DESCRIPTION OF IMAGE</alt>
  <imageobject>
    <imagedata width="100%" scalefit="1" fileref="media/FILENAME.png"/>
  </imageobject>
</mediaobject>
```

In HTML the image will scale down to 900px width (*default.image.width* setting in *~/stylesheets/sitka_xhtml.xsl*). In PDF it will scale down to page width.



Wrap the *mediaobject* tag in a *para* to add a small space between the image and preceding content.

Keeping image and text together

Content wrapped in any formal or informal object (eg *informalfigure*, *formalfigure*, *informalexample*, *formalexample*) is kept together on the same page in pdf output. Use *informalfigure* to keep an image with associated text, as in this example of a *step*:

```
<step>
  <informalfigure>
    <para>STEP TEXT HERE</para>
    <para>
      <mediaobject>
        <alt>OPTIONAL DESCRIPTION OF IMAGE</alt>
        <imageobject>
          <imagedata width="100%" scalefit="1" fileref="media/FILENAME.png"/>
        </imageobject>
      </mediaobject>
    </para>
  </informalfigure>
</step>
```

Chapter 3. Creating Links

There are many way to link w/ docbook but the options in the table below cover the most common documentation scenarios.

| Linking scenario | Syntax | Output | Notes |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Internal link with auto-generated link text | <code><xref linkend="style-glossary"/></code> | <u>Chapter 1, DIG DocBook reference</u> | The <i>linkend</i> value (<i>style-glossary</i>) is the <i>xml:id</i> of the target element; link text is auto-generated from the target. The tag is self-closing (no <code></xref></code> closing tag required) |
| Internal link with custom link text | <code><link linkend="style-glossary">The DIG glossary page</link></code> | <u>The DIG glossary page</u> | <i>linkend</i> is the <i>xml:id</i> of the target element; link text is supplied by author |
| External link | <code><link xlink:href="http://evergreen-ils.org/" xlink:title="Link hover text goes here">http://evergreen-ils.org</link></code> | <u>http://evergreen-ils.org/</u> | <i>xlink:href</i> is the target URL. It is good practice to show full URL so users can see it in PDF or copy and paste from pages displayed in the staff client (to open in a separate browser) |

For internal links the hover tip text is generated automatically; for external links please include an `xlink:title`, otherwise the link hover text will be the name of the section or chapter where the link appears.

Chapter 4. Processing instructions

Processing instructions (PIs) are written into the xml when the desired output cannot be achieved with DocBook elements and attributes alone. PIs may also be used to force special formatting in only part of a document. For a more technical introduction see Bob Stayton's [DocBook XSL: The Complete Guide](#). There are also lists of DocBook-specific processing instructions for [html](#) and [pdf](#) output.

PIs should be used sparingly: it is easier to maintain documents that achieve the desired output with DocBook elements and [stylesheet parameters](#). However, there are a few cases where PIs may be necessary:

- **Force line break.** Line breaks should not be used for spacing between elements; this is much better done with CSS and XSL. However, if entering multiple lines that must be displayed in succession but without the separation of a new paragraph, you may use the linebreak PI. The following code

```
<para>First line <?linebreak?>Second line <?linebreak?>Third line</para>
```

Will output

First line Second line Third line



This only works if custom stylesheets include the following code, described in [Bob Stayton's DocBook XSL](#). Sitka's customization layer includes this template as of June 30, 2010:

```
<xsl:template match="processing-instruction('linebreak')"> <fo:block/> </xsl:template>
```

- **Include external content.** To insert external content into the html output use the *dbhtml-include* processing instruction as described in [DocBook XSL](#). This could be used to add a custom search box to the title page, an animated tutorial in the relevant section, or any other content in valid xhtml.

Chapter 5. Miscellaneous Tips

Table column width

If column width is not specified in the XML, tables will look different depending on the output format. In HTML the columns automatically adjust to fit contents, but in PDF the page width is divided evenly by the number of columns. This often wastes space in PDF tables so setting relative column widths for all tables is recommended.

In the example below, the *colwidth* parameter defines relative column width. Columns 1 and 2 are of equal width, each using a quarter of the page. Column 3 is twice the size and covers the remaining half of the page.

```
<table>
  <title>Title of Table Here</title>
  <tgroup cols="3" align="left" colsep="1" rowsep="1">
    <colspec colnum="1" colname="issues" colwidth="1*" />
    <colspec colnum="2" colname="actions" colwidth="1*" />
    <colspec colnum="3" colname="tasks" colwidth="2*" />
  </tgroup>
</table>
```

For details see [DocBook XSL: The Complete Guide](#).

To avoid the following error during processing of PDF files,

```
WARNING: table-layout="fixed" and column-width unspecified => falling back to proportional-column-width(1)
```

it seems necessary to add decimal values in the *colwidth* parameter `<colspec colwidth="1.0*" />`

See <http://osdir.com/ml/fop-users-xmlgraphics.apache.org/2009-04/msg00109.html>

Using CDATA for XML Code

if you find yourself with long `<programlisting>` or `<screen>` sections that contain a lot of XML code, you can wrap the whole thing in `<![CDATA[...]>` and the XML parser will skip it. For example:

```
<programlisting language="xml">
<![CDATA[
<!-- Example of an app-specific setting override -->
<opensrf.persist>
  <app_settings>
    <dbfile>/tmp/persist.db</dbfile>
  </app_settings>
</opensrf.persist>
]]&gt;
</programlisting>
```

It's a lot easier to add the `<![CDATA[...]>` wrapper than change the individual occurrences of `<` and `>`.

Semantic Oriented Markup

Adopting more semantic markup gives us the ability to have more fine-grained control over the stylesheets and resulting output. Finding semantic alternatives to the `<emphasis>` tag is important for allowing more control over display of the content.

Here are some examples of existing DocBook tags which should be used in place of `<emphasis>`:

- `<command>` is used for command names (<http://www.docbook.org/tdg5/en/html/command.html>)
- `<filename>` is used for file paths and file names (<http://www.docbook.org/tdg5/en/html/filename.html>)
- `<option>` is used for command options (<http://www.docbook.org/tdg5/en/html/option.html>)
- `<systemitem class="username">` is used for user names (<http://www.docbook.org/tdg5/en/html/systemitem.html>)