

# 1. Side floats

Copyright © 2005 Bob Stayton

A *side float* is used to position a block of content to the side of a printed page. The body text will wrap around the side float if it doesn't take up the whole width of the page, or the side float can be positioned in a side indent so it doesn't intrude on the text area.

The DocBook XSL fo stylesheet added support for side floats starting in version 1.68 of the stylesheet. Side floats only work with those XSL-FO processors that support them. Two that are known to work are RenderX's XEP and Antenna House's XSL Formatter.

The DocBook FO stylesheet provides support for three kinds of side floats:

- The `sidebar` element rendered as a side float.
- A `sidebar` element with a processing instruction that identifies it as a margin note.
- Custom side floats.

## 1.1. A sidebar as side float

One common use of side floats is for the `sidebar` element. By default, a `sidebar` prints as a full-width block on the page instead of a side float. This example shows some text in a sidebar to demonstrate how it can appear in a left side float.

### Sidebar example

A `sidebar` can be used to emphasize certain content. It can intrude into the body area, or it can appear in side indents as a margin note.

```
<sidebar>
<title>Sidebar example</title>
<?dbfo sidebar-width="1.5in"?><?dbfo float-type="left"?>
<para>A <sgmltag class="element">sidebar</sgmltag> can be
used to emphasize certain content. It can intrude into
the body area, or it can appear in side indents as
a margin note.</para>
</sidebar>
```

## 1. Side floats

---

The formatting of a sidebar is controlled by these features:

- *sidebar.float.type* parameter
- *sidebar.float.width* parameter
- *sidebar.properties* attribute-set
- *sidebar.title.properties* attribute-set
- Processing instructions for individual instances.

The overall behavior is controlled by the *sidebar.float.type* stylesheet parameter. That parameter contains one of the values of the `float` property as defined in the XSL-FO specification:

<code>none</code>	No float is used. The sidebar appears in sequence with other text blocks. This is the default value.
<code>left</code>	The float appears on the left side of the page.
<code>start</code>	The float appears on the start side of the page. For languages that read left-to-right, that is the left side of the page.
<code>right</code>	The float appears on the right side of the page.
<code>end</code>	The float appears on the end side of the page. For languages that read left-to-right, that is the right side of the page.
<code>inside,</code> <code>outside,</code>	Some XSL-FO processors support an extension that lets you place the float on the inside or outside edge of a double-sided layout. For a right-hand page (usually numbered odd), an inside float is on the left.
<code>before</code>	The float appears at the top of the page, using the full width of the page. This is not a side float, but is included here for completeness.

The *sidebar.float.type* sets the default location for all sidebars, but you can change the location for an individual sidebar element using a `float-type` processing instruction. Here is an example that will create a float on the right side of the page:

```
<sidebar>
<?dbfo float-type="right"?>
<title>My floating sidebar</title>
...
```

The width of the sidebar float is controlled by *sidebar.float.width* parameter, which is set to 1 inch by default. You may need to set a different width for individual sidebar elements.

You can change the width for one sidebar using a `sidebar-width` processing instruction, such as this example:

```
<sidebar>
<?dbfo sidebar-width="3.5in"?>
<title>My floating sidebar</title>
...
```

In addition to the above properties being applied to the float container, the `sidebar.properties` attribute set is applied to the block of content within the container. Use that property set to change the font size, text alignment, background, or border for the content.. Here are the default values:

```
<xsl:attribute-set name="sidebar.properties"
  use-attribute-sets="formal.object.properties">
  <xsl:attribute name="border-style">solid</xsl:attribute>
  <xsl:attribute name="border-width">1pt</xsl:attribute>
  <xsl:attribute name="border-color">black</xsl:attribute>
  <xsl:attribute name="background-color">#DDDDDD</xsl:attribute>
  <xsl:attribute name="padding-left">12pt</xsl:attribute>
  <xsl:attribute name="padding-right">12pt</xsl:attribute>
  <xsl:attribute name="padding-top">6pt</xsl:attribute>
  <xsl:attribute name="padding-bottom">6pt</xsl:attribute>
  <xsl:attribute name="margin-left">0pt</xsl:attribute>
  <xsl:attribute name="margin-right">0pt</xsl:attribute>
  <!--
  <xsl:attribute name="margin-top">6pt</xsl:attribute>
  <xsl:attribute name="margin-bottom">6pt</xsl:attribute>
  -->
</xsl:attribute-set>
```

These properties draw a border around the sidebar content and add a background color. The padding properties add some space between the sidebar text and the border. The margin properties can be used to provide some space between the border and the body text that wraps around the sidebar. To learn how to customize an attribute-set, see ???.

In addition, the separate `sidebar.title.properties` attribute-set is applied to the sidebar's title element. Here are its default values:

```
<xsl:attribute-set name="sidebar.title.properties">
  <xsl:attribute name="font-weight">bold</xsl:attribute>
  <xsl:attribute name="hyphenate">>false</xsl:attribute>
  <xsl:attribute name="text-align">start</xsl:attribute>
  <xsl:attribute name="keep-with-next.within-column">always</xsl:attribute>
</xsl:attribute-set>
```

## 1.2. Margin notes

Margin notes are blocks of content that are meant to appear in the empty margin area, rather than having text wrap around them like with a sidebar. By putting them in the margin area, they don't interrupt the flow of paragraphs.

The DocBook DTD does not provide a standard way of specifying margin notes. However, the FO stylesheet supports a special sidebar as a margin note. You can also implement margin notes as a customization, as described in Section 1.3, "Custom margin notes" (page 5).

You can turn a `sidebar` element into a margin note by adding this processing instruction:

```
<sidebar>
<?dbfo float-style="margin.note"?>
...
</sidebar>
```

When a sidebar is designated was a margin note in this way, the stylesheet applies a different set of properties to create the side float. This lets you create one style for regular sidebars, and another for margin notes.

To enable margin notes, you have to give them some open space so they don't intrude on your paragraphs. Increasing the side page margins won't work because the XSL float mechanism puts text within the body region, not outside of it in the margins. So you need to create an additional indent for all body text on the page, and the margin note floats can appear within the indent.

You can create such an indent on the left (or start side) of the page by using the `body.start.indent` parameter. If you want the indent on the right (or end side) of the page, then use the `body.end.indent` parameter. If you don't set the extra body indent, then your margin notes will intrude into the body text area. This example indents the body text an extra 25mm on the left

```
<xsl:param name="body.start.indent">25mm</xsl:param>
```

These parameters are used by the template named `set.flow.properties`, which lets you add properties that apply to an entire `fo:flow` in a page sequence. By default, these parameters are used only for page-sequences using the `body page-master`, and for `preface` and `appendix` pages. You can customize the `set.flow.properties` template to change which flows use these indent parameters.

Chapter titles and section titles are not indented by default. That's because they have their own `start-indent` property that is set to zero in their titlepage specifications. So the use of the `body.start.indent` parameter has the same effect as the `title.margin.left` parameter, which should be set to 0pt if you use `body.start.indent`. If you do want your chapter or section titles to be indented as well, then add your own `start-indent` property and sets its value to `$body.start.indent`. See ??? to learn how to set properties on titles.

After you have set up the indent area, you need to specify the formatting properties for margin notes. The *margin.note.float.type* determines the position of the margin note, using the same selection of values as for *sidebar.float.type*. See Section 1.1, “A sidebar as side float” (page 1). Note that you can't use the `dbfo float-type` processing instruction to change the position of an individual margin note. That's because that PI is already being used to indicate that a sidebar is to be handled like a margin note.

You can set the standard width for all your margin notes by using the *margin.note.width* parameter. To change the width for a single instance, you can use the `dbfo sidebar-width` processing instruction within its *sidebar* element. If you make a margin note wider than the enlarged margin, then the margin note will intrude into the body text, which will wrap around it.

And to give your margin notes a distinctive look, add attributes to the *margin.note.properties* attribute-set. That attribute-set is applied to the block that contains the content of the *sidebar* element. For the *title* element in a margin note sidebar, use the *margin.note.title.properties* attribute-set.

Here is a complete example of setting the parameters and attribute-sets for margin notes:

```
<xsl:param name="body.start.indent">30mm</xsl:param>
<xsl:param name="margin.note.float.type">start</xsl:param>
<xsl:param name="margin.note.width">27mm</xsl:param>
<xsl:attribute-set name="margin.note.properties">
  <xsl:attribute name="font-size">8pt</xsl:attribute>
  <xsl:attribute name="font-family"><xsl:value-of
    select="$title.fontset"/></xsl:attribute>
  <xsl:attribute name="border">0.5pt solid green</xsl:attribute>
  <xsl:attribute name="padding">3pt</xsl:attribute>
</xsl:attribute-set>
<xsl:attribute-set name="margin.note.title.properties">
  <xsl:attribute name="font-size">9pt</xsl:attribute>
</xsl:attribute-set>
```

To learn how to customize an attribute-set, see ???.

### 1.3. Custom margin notes

You may want to turn other elements in your DocBook content into margin notes. For example, you might want to put a short phrase with a cross reference in the margin, or put a *note* element in the margin. The FO stylesheet uses two templates to create margin notes, the *margin.note* template and the *float* template.

The template named *margin.note* in *fo/block.xsl* is called by the *sidebar* template when the `dbfo float-style="margin.note"` processing instruction is used. The *margin.note* template applies all the margin note properties described in Section 1.2, “Margin notes” (page 4) to the

## 1. Side floats

---

context element that calls the template. So you would use the `margin.note` template when you want your customization to look like other margin notes.

Here are customizations that put a phrase or a note element with a `role="margin.note"` attribute into a margin note:

```
<xsl:template match="phrase[@role='margin.note']">
  <xsl:call-template name="margin.note"/>
</xsl:template>
```

```
<xsl:template match="note[@role='margin.note']">
  <xsl:call-template name="margin.note">
    <xsl:with-param name="content">
      <xsl:apply-imports/>
    </xsl:with-param>
  </xsl:call-template>
</xsl:template>
```

Both of these customizations call the `margin.note` template. They differ in the use of the `content` parameter when the template is called. The `margin.note` template just does an `xsl:apply-templates` to the context element that calls it. That means it processes the element's children, which is appropriate for a `phrase` element. But if you did that with `note`, then it would process the `para` within the note, but not the note element itself. So that template fills in the `content` parameter by doing `apply-imports` on the note element, which formats the note using the admonition templates.

### 1.4. Custom side float

If you need more control than is provided by the `margin.note` template, then you can call the template named `floater`, which is also in `fo/block.xsl`. This is the template that actually creates a side float in the FO output, and it is called by the `sidebar` and `margin.note` templates. The `floater` template takes several parameters to define the float, and does not apply any formatting properties to the content. Thus you can use it to create custom side floats from any element.

Here is an example of using `floater` to put an admonition graphic into the side margin:

**Example 1. Custom side float**

```

<xsl:param name="body.start.indent">1.5in</xsl:param> ❶
<xsl:param name="title.margin.left">0pc</xsl:param> ❷
<xsl:param name="admon.textlabel" select="0"/> ❸

<xsl:template match="note|caution|warning|tip|important"> ❹
  <xsl:call-template name="floater"> ❺
    <xsl:with-param name="position">left</xsl:with-param> ❻
    <xsl:with-param name="width">0.5in</xsl:with-param> ❼
    <xsl:with-param name="start.indent">0.5in</xsl:with-param> ❽
    <xsl:with-param name="content"> ❾
      <fo:block margin-top="1em"
        padding="3pt"
        text-align="center"
        border="1pt solid grey">
        <fo:external-graphic width="auto" height="auto">
          <xsl:attribute name="src">
            <xsl:call-template name="admon.graphic"/>
          </xsl:attribute>
        </fo:external-graphic>
      </fo:block>
    </xsl:with-param>
  </xsl:call-template>
  <xsl:apply-imports/> ❿
</xsl:template>

```

- ❶ Establish a side margin area by indenting the body text 1.5 inches.
- ❷ Set *title.margin.left* to zero so it doesn't conflict with the *body.start.indent*.
- ❸ Turn off the admonition text label such as Note so only the icon is used to identify the admonition.
- ❹ Customized template to match on all admonition elements.
- ❺ Call the template named *floater* to create a side float.
- ❻ Indicate the position of the float. The default value is none, so pick one of the values listed in Section 1.1, "A sidebar as side float" (page 1) to put it in the margin.
- ❼ You can specify the width of the side float container.
- ❽ You can specify an indent for the side float container.
- ❾ Fill the content parameter with whatever you want to appear in the side float. In this case, it is an admonition graphic, selected using the *admon.graphic* template. It is wrapped in an *fo:block*, which is used to format the content.
- ❿ Now actually process the admonition element by using *xsl:apply-imports*. This uses the admonition templates imported from the DocBook stylesheet.

1. Side floats

---