

8

Floats
and
footnotes



8 Floats and footnotes

Floats and footnotes are used to render distinctive information that is supplemental to the information found in the flow, making it easy for the reader to find it because of its predetermined location on the page. These constructs are defined “in line” of the flow of information being paginated, but are rendered “out of line” of the flow of information being paginated. Such information is considered auxiliary enough not to disturb the flow itself for the reader, so that the reader can choose to examine a float or footnote at leisure without interrupting the reading of the flow in which it is referenced.

These constructs are dynamically rendered on the page where detected by the formatter in the flow, or perhaps on the immediately following page should it not fit where referenced. They can only be defined in the flow filling the body region of a page, though not in an absolutely positioned area, and are stacked in a different reference area than the main reference area. Floats are moved to either the before, start, or end edges within the body region (not in the perimeter regions). Footnotes are two-part constructs: the footnote citation that is rendered inline in the flow and the footnote body is rendered at the after edge within the body region. An

after float is accomplished using a footnote without a footnote citation, but can only be used when not using footnotes since this kind of after float cannot be distinguished from other footnotes on the same page.

Note that the labeling of footnotes is up to the transformation process producing the XSL-FO instance and cannot be generated by the formatter. A feature that is commonly requested but unavailable in XSL-FO 1.0 is numbering the footnotes on a per-page basis. Since the transformation process is unaware of where the page breaks are, yet is responsible for the labeling of the footnote references, this feature cannot yet be realized.

There are no endnote layout constructs in XSL-FO. An endnote is a two-part footnote-like construct with a citation and a definition, but all endnote definitions are collected at the end of a page sequence (e.g. chapter) rather than at the bottom of pages. To render endnotes, it would be the responsibility of transformation to cite the endnotes inline in the flow of the scope and then collect and render endnotes at the end of the scope. You could then choose to render the citations as part of the flow on the body, or use empty citations to take advantage of the XSL-FO footnote construct to sink the collection of endnotes to the bottom of the last page.

Judicious use of floats can keep the main flow of information continuous without unseemly breaks at the bottom of pages. The formatter moves the floating object to the perimeter and flows subsequent information in the main flow without interruption, as if the flow had not been present. If the construct is too large to fit on the page, it can be rendered on the next page and the subsequent information continues on the previous page in a seamless fashion.

There are many candidate uses of floating constructs. You can use `float` to float images to the side of a page, e.g. to present sidebar portions of content, or to format lists where the item bodies are indented relative to the corresponding item label's formatted length. It can also be used to implement a multi-line drop initial cap in a paragraph. You can use `footnote` to sink content to the bottom of a page, e.g. for traditional footnotes, acronym expansions, glossary definitions, or images floating to the bottom of a page (using an empty inline construct).

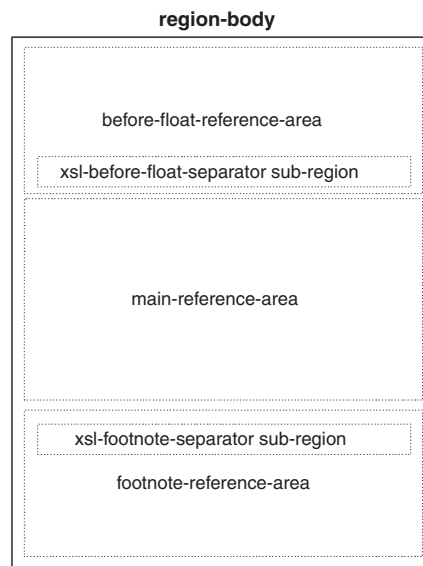
The name of the construct shouldn't prejudice how the construct is used. Consider the need to format a disclaimer at the bottom of the last page of a document. You can flow the disclaimer in the body of a footnote with

an empty citation in an empty block at the end of the document. The disclaimer isn't a footnote, yet you'll get the desired effect of the out-of-line placement.

Every page's body region has two sub-regions that are rendered only if necessary. Before floats and footnotes are stacked in the body region with other block level constructs, but the reader needs some separation rendered to distinguish content belonging in a float or footnote from the content belonging in the body. The body region is separated into the before-float reference area, main reference area, and footnote reference area portions, as shown in Figure 8–1.

The act of defining these visual separators does not affect their rendering, as they are only rendered on a page if the floated information is being rendered on the page. Static content defines the rendering of a separator. When needed, the static content associated with `xsl-before-float-separator` is rendered inside and at the end of the before-float reference area. When needed, the static content associated with `xsl-footnote-separator` is rendered inside and at the start of the footnote reference area.

Figure 8–1 Conditional areas and sub-regions



Static content for these sub-regions should always be defined as a contingency if floats and footnotes are being used, for the chance that a given page may have such a construct.

Remember from Figure 4–2 the incursion of the perimeter regions into the body: all reference areas shown above are within the body region’s margins and not part of the perimeters.

Included in this chapter. This chapter includes discussion of the following XSL-FO objects:

- `float` (6.10.2):
 - is the content to be rendered towards either the before, start, or end edges of the body region regardless of where in the region the content is defined,
- `footnote` (6.10.3):
 - is the content to be rendered partly in the flow and partly towards the after edge of the body region regardless of where in the region the content is defined,
- `footnote-body` (6.10.4):
 - is the portion of footnote content rendered towards the after edge of the body region.

8.1 Floats

8.1.1 Float definition

Using the `float` object, you can position out-of-line information at the before, start, or end sides of a page.

- The before floats are flowed in the before-float reference area (see Figure 8–1).
- A float creates a dimensionless anchor area in the area tree.
- The anchor is tied to the information that precedes the float in the flow.
- Out-if-line information is placed relative to the anchor —
 - typically at the top or sides of the page in which the anchor is flowed.
- The content of the float is always a set of block-level constructs.

Information can be defined at the block level or at the inline level of the flow.

- The anchor is treated as a block if defined at the block level.

- The anchor is typically treated as an inline construct if defined at the inline level.
 - It is treated as a block object if the line area created by the inline constructs consists only of anchors.
- It must be defined in the body region of the page.
- It must be defined as a descendant of a relatively positioned block.
 - Absolutely positioned areas cannot have float descendants.
- It cannot have any `float`, `footnote`, or `marker` descendants.

The blocks of the float definition are positioned in the area tree accordingly.

- The before floats are stacked in the before-float reference area.
- The side floats are stacked in a side-float reference area.
 - This is a child of the specifying block's ancestral span reference area.
- Float content that does not float is flowed as normal content.
 - This may be the result of a faulty definition or placement.
 - A float is an area that is not normally flowed and cannot be defined within any other area that is not normally flowed (e.g. another float, a footnote, a perimeter region, an absolutely positioned block container, etc.).
- Float lengths cannot be preset and are always derived from the content of the float.

The block areas that float have no border or padding.

- Padding, border, and content rectangles are coincident.

Side floats do not overlap and are typically placed beside each other.

- They make a further incursion into the inline-progression dimension of the block.
- They can be forced to not be placed beside each other by using the `clear` property.
 - Different values control which floats are clear of other floats.

8.1.2 The float object

Purpose

- This is the content to be rendered towards either the before, start, or end edges of the body region regardless of where in the region the content is defined.

Content

- (6.10.2) (%block;)+,
- child object:
 - %block; (6.2; 69).

Optional properties

- clear (7.18.1; 368),
- float (7.18.2; 376).

8.1.3 The interaction of blocks and floats

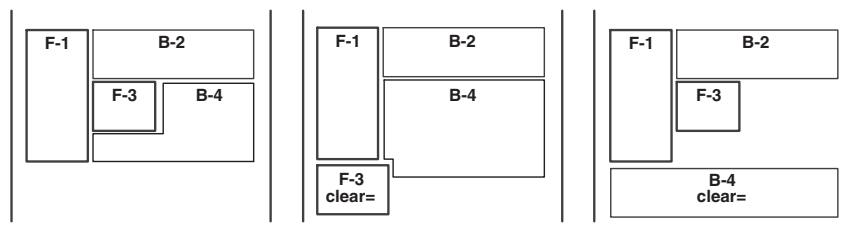
The `clear` property is used to prevent two side floats from being beside each other or a block-level object from being beside a float.

- It is not documented as a property within the individual block-level object descriptions.
 - It applies to all such constructs as documented in the property definition.
- The default is for side floats to continue intruding into the main reference area.
 - Use `clear` to start a side float back at the edge, clear of any previous float.
- The default is for the lines of a block to contour around the accumulation of side floats.
 - Use `clear` to start a block back at the edge, clear of any previous float.
- Those constructs with the `clear` value clear the float they would otherwise be beside.

Consider three situations in Figure 8–2 that illustrate the use of `clear` on each of floats and blocks;

- the numbers reflect the order in which the constructs are flowed;

Figure 8–2 The clearing of slide floats by new constructs



- the thick-lined blocks with “F-” prefixes are floats;
- the thin-lined blocks with “B-” prefixes are blocks.

Of note in the diagram:

- the left-most page fragment shows the default situation where side floats intrude and blocks flow around floats;
 - the second float (“F-3”) is to the right of the first float (“F-1”);
 - the second block (“B-4”) wraps around both floats;
- the center page fragment shows the second float being clear of the first float;
 - the second block (“B-4”) starts abutted to the first block (“B-2”);
- the right-most page fragment shows the second block being clear of all floats;
 - the second float (“F-3”) is to the right of the first float (“F-1”);
 - the second block (“B-4”) is clear of both floats.

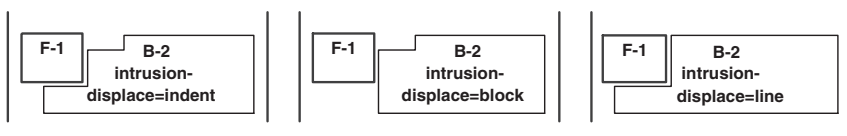
The `intrusion-displace` property indicates the strategy of locating the start (or end) edges and indents of the lines of the block being intruded upon.

- Note that displacing the edge of a block necessarily displaces all lines in that block.

Consider three situations in Figure 8–3 that illustrate the use of `intrusion-displace` on a block with both `text-indent` and `start-indent`.

- “indent” respects `text-indent` and `start-indent`;
- “block” respects `text-indent` but limits all lines by the width of the float;
- “line” ignores `text-indent` if occupied by the float, and respects `start-indent`.

Figure 8–3 The behavior of lines in a block next to a float



8.2 Footnotes

8.2.1 Footnote definition

Using the `footnote` object, you can position out-of-line information at the bottom of a page.

- The anchor area is the last area defined by the `inline` child.
- Out-of-line information is placed relative to the anchor —
 - typically at the bottom of the page in which the anchor is flowed.
- Out-of-line content is the content of the `footnote-body` child.
 - It is always a set of block-level constructs.

Information must be defined at the inline level of the flow.

- The object supplies the content of the inline anchor.

The blocks of the footnote body are positioned in the area tree accordingly.

- The footnote body is flowed to the footnote reference area (see Figure 8–1).
- Footnote body length cannot be preset and is always derived from the content of the float.

The same restrictions as for `float` apply.

- A footnote's body is an area that is not normally flowed and cannot be defined within any other area that is not normally flowed (e.g. another footnote, a float, a perimeter region, an absolutely positioned block container, etc.).
- It must be defined in the body region of the page as a descendant of a relatively positioned block.
- It cannot have any `float`, `footnote`, or `marker` descendants.

There exist no built-in semantics of footnote numbering or citations.

- Numbering and presentation is the responsibility of the stylesheet or application generating the flow objects, including —
 - what is displayed inline and how it is displayed,
 - e.g. superscripting, italics, font size, etc.,
 - how the inline reference is reflected in the out-of-line information for correlation by the reader.
- Numbering cannot be restarted on a per-page basis.

- The page boundaries are determined by the formatter and are not known to the transformation process.
- The XSLT process producing the XSL-FO instance can probably use the same criteria for determining the page sequence boundaries to do footnote numbering either across the page sequence or across the entire publication.

There is no automatic copying of the inline content to the start of the footnote body.

- It is the stylesheet writer's responsibility to define both the footnote citation and the footnote body separately.
- You may wish to use different formatting properties for each use of the footnote citation.

8.2.2 The footnote object

Purpose

- This is the content to be rendered partly in the flow and partly towards the after edge of the body region regardless of where in the region the content is defined.
 - It includes the inline content to render in the flow where the footnote body content is defined.

Content

- (6.10.3) (*inline*, *footnote-body*),
- child objects (listed alphabetically):
 - *footnote-body* (6.10.4; 238),
 - *inline* (6.6.7; 103).

Property sets

- Common accessibility properties (7.4; 326).

No other properties are defined for this formatting object.

The example shown in Figure 8–4 illustrates the use of footnotes.

An excerpt from Figure 8–4 is shown in Example 8–1.

Of note:

- the footnote definitions are embedded in the blocks at the point where the footnote citation is rendered;

- the footnote citation in the block is formatted differently from the footnote citation echoed in the footnote reference area.

8.2.3 The footnote-body object

Purpose

- This is the portion of footnote content rendered towards the after edge of the body region.

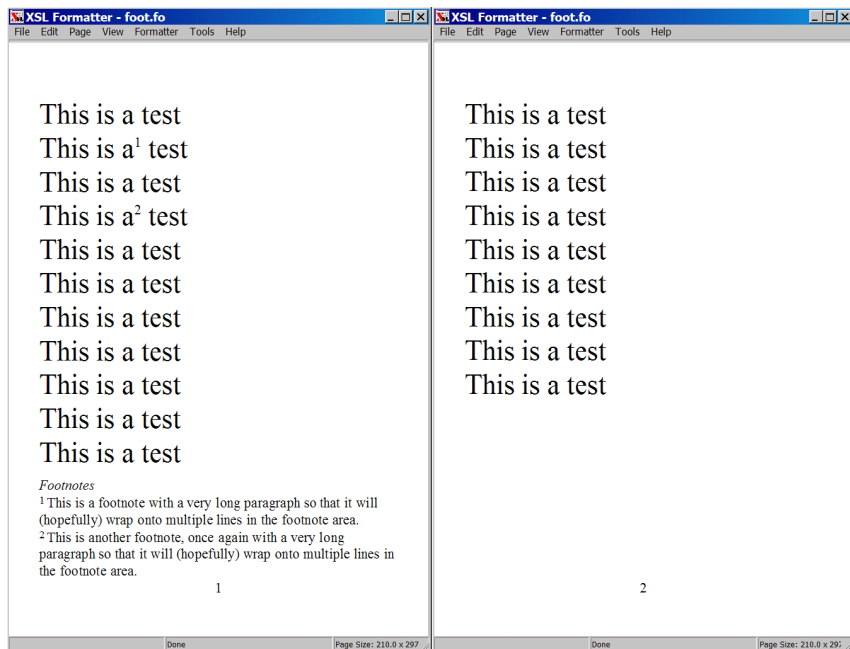
Content

- (6.10.4) (%block;)+,
- child object:
 - %block; (6.2; 69),
- referring object:
 - footnote (6.10.3; 237).

Property sets

- Common accessibility properties (7.4; 326).

Figure 8–4 Example for footnotes



Example 8–1 Footnote constructs in Figure 8–4

```

Line 01 <page-sequence master-reference="frame">
02   <static-content flow-name="xsl-footnote-separator">
03     <block font-style="italic">Footnotes</block>
04   </static-content>
05
06   <flow flow-name="frame-body" font-size="40pt">
07     <block>This is a test</block>
08     <block>This is a<footnote>
09       <inline baseline-shift="15pt" font-size="20pt">1</inline>
10       <footnote-body>
11         <block font-size="20pt">
12           <inline baseline-shift="5pt" font-size="15pt"
13             >1 </inline>This is a footnote with a very
14 long paragraph so that it will (hopefully) wrap onto
15 multiple lines in the footnote area.
16         </block>
17       </footnote-body>
18     </footnote> test</block>
19     <block>This is a test</block>
20     <block>This is a<footnote>
21       <inline baseline-shift="15pt" font-size="20pt">2</inline>
22       <footnote-body>
23         <block font-size="20pt">
24           <inline baseline-shift="5pt" font-size="15pt"
25             >2 </inline>This is another footnote, once
26 again with a very long paragraph so that it will
27 (hopefully) wrap onto multiple lines in the footnote area.
28         </block>
29       </footnote-body>
30     </footnote> test</block>
31     <block>This is a test</block>
32     <block>This is a test</block>
33     ...

```

No other properties are defined for this formatting object.

An excerpt from Figure 8–4 highlighting the `footnote-body` object is shown in Example 8–2.

Example 8-2 Footnote constructs in Figure 8-4

```

Line 01 <page-sequence master-reference="frame">
02   <static-content flow-name="xsl-footnote-separator">
03     <block font-style="italic">Footnotes</block>
04   </static-content>
05
06   <flow flow-name="frame-body" font-size="40pt">
07     <block>This is a test</block>
08     <block>This is a<footnote>
09       <inline baseline-shift="15pt" font-size="20pt">1</inline>
10       <footnote-body>
11         <block font-size="20pt">
12           <inline baseline-shift="5pt" font-size="15pt"
13             >1 </inline>This is a footnote with a very
14 long paragraph so that it will (hopefully) wrap onto
15 multiple lines in the footnote area.
16         </block>
17       </footnote-body>
18     </footnote> test</block>
19     <block>This is a test</block>
20     <block>This is a<footnote>
21       <inline baseline-shift="15pt" font-size="20pt">2 </inline>
22       <footnote-body>
23         <block font-size="20pt">
24           <inline baseline-shift="5pt" font-size="15pt"
25             >2 </inline>This is another footnote, once
26 again with a very long paragraph so that it will
27 (hopefully) wrap onto multiple lines in the footnote area.
28         </block>
29       </footnote-body>
30     </footnote> test</block>
31     <block>This is a test</block>
32     <block>This is a test</block>
33     ...

```
