```
<charlist inherit="1">
<textbrk startpg="recto" pageid="front.page" newpgmdl="local">
...
```

FOSI-generated HTML

Using FOSI and an XSLT post-process, it is possible to generate structured HTML with Arbortext Editor.

NOTE: FOSI-generated HTML is not the same as File▸Save as HTML menu selection, which is based on formatting for print/PDF output but does not create structured HTML.

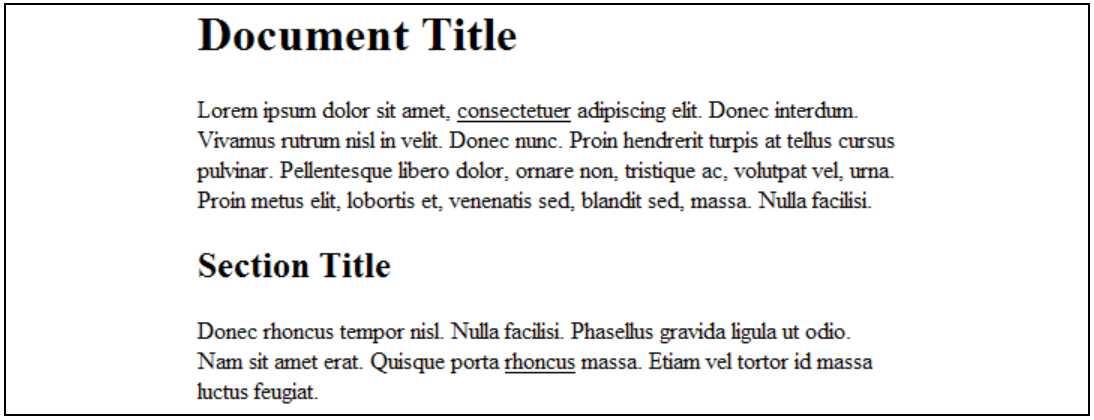
When the document is formatted, the FOSI generates structured HTML for block elements in the source document and writes the resulting document to an external file. The default .exp file extension can be changed with APTEXPORTEXT environment variable (page 163).

In a FOSI, the technique for generating tables from non-table markup described in **Generated tables** on page 158 is used to generate structured HTML. Usetext userule="1" outputs the generated HTML to the external file. However, the Arbortext Editor formatting engine does not transform inline elements, which is why XSLT is needed. The XSLT transformation runs quickly since the document structure is not affected.

FOSI-generated HTML examples

The first example shows how a simple document with two levels of titles, paragraphs, and inline elements is transformed into structured HTML. Usetext userule="1" writes the appended string with the HTML tags and content to an external file. File▸Compose▸XSL... is used with the external file and the XSLT stylesheet to transform inline element names and create the final HTML file.

Figure 107 Generated structured HTML file



XML fragment

```
<document>
<title>Document Title</title>
<para>Lorem ipsum dolor sit amet, <underline>consectetuer</underline>
adipiscing...Nulla facilisi.</para>
<section>
<title>Section Title</title>
<para>Donec ...porta <underline>rhoncus</underline> massa.....</para>
...
```

FOSI fragment

```
<stringdecl textid="outut-html.app" literal="">
...
<e-i-c gi="document">
<charlist inherit="1">
...
<usetext source="<output-html>,</output-html>" placemnt="after">
</usetext>
...
<e-i-c gi="output-html">
<charlist inherit="1">
<usetext source='\<?xml version="1.0" encoding="UTF-8"
standalone="no"?>\,<html>,<head>,</head>,<body>,output-html.app,
</body>,</html>' userule="1"></usetext>
...
<e-i-c gi="para">
<charlist inherit="1" charsubsetref="block prespace">
<savetext textid="output-html.app" conrule="<p>,#CONTENT,</p> append="1">"
...
<e-i-c gi="title" context="document">
<charlist inherit="1" charsubsetref="title-level1">
<savetext textid="output-html.app" conrule="<h1>,#CONTENT,</h1>" append="1">
```

```
...
<e-i-c gi="title" context="section">
<charlist inherit="1" charsubsetref="title-level2">
<savetext textid="output-html.app" conrule="<h2>,#CONTENT,</h2>" append="1">
...
<e-i-c gi="underline">
<charlist inherit="1" charsubsetref="underline "></charlist>
</e-i-c>
```

Exported file fragment

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<html>
<head></head>
<body>
<h1>Document Title</h1>
<p>Lorem ipsum dolor sit amet, <underline>consectetuer</underline>
adipiscing...Nulla facilisi.</p>
<h2>Section Title</h2>
<p>Donec ...porta <underline>rhoncus</underline> massa.....</p>
...
```

XSLT stylesheet

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="@* | node()">
<xsl:copy>
<xsl:apply-templates select="@* | node()"></xsl:apply-templates>
</xsl:copy>
</xsl:template>
<xsl:template match="underline">
<u>
<xsl:apply-templates></xsl:apply-templates>
</u>
</xsl:template>
...
```

In the following example, the FOSI-generated HTML includes boxing with rounded corners.

Figure 108 Rounded corners in FOSI-generated HTML