#### altChunk (Anchor for Imported External Content)

This element specifies a location within a document for the insertion of the contents of a specified file containing external content to be imported into the main WordprocessingML document. The specified file's contents should appear at the specified location within the document, and may henceforth be emitted as regular WordprocessingML without distinction to its origin. The location of the external content to be imported shall be specified by the relationship whose Id attribute matches the [id](id.docx) attribute on this element.

If the relationship [type](type.docx) of the relationship specified by this element is not http://schemas.openxmlformats.org/officeDocument/2006/afChunk, is not present, or does not have a TargetMode attribute value of Internal, then the document shall be considered non-conformant. If an application cannot process external content of the content [type](type.docx) specified by the targeted part, then it should ignore the specified alternate content but continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

[Example: Consider a WordprocessingML document consisting of contents which shall be imported from the following HTML document:

<html … >
 <[body](body.docx) style="margin-left:200px;margin-top:50px">
 <[p](p.docx)>Paragraph one.</[p](p.docx)>
 <blockquote style="border:5px solid #00FFFF">Paragraph in a blockquote.</blockquote>
 <[p](p.docx)>Paragraph two.</[p](p.docx)>
 </[body](body.docx)>
</html>

The resulting WordprocessingML host document would consist of its own WordprocessingML content as well as an external content import anchor in the appropriate location:

<w:[body](body.docx)>
 <w:altChunk r:id="altChunk1" />
 <w:[p](p.docx)/>
 <w:[sectPr](sectPr.docx)>
 …
 </w:[sectPr](sectPr.docx)>
</w:[body](body.docx)>

The altChunk element specifies that the external content targeted by the relationship with an ID of altChunk1 shall be imported at the beginning of the document. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

<Relationships … >
 …
 <Relationship Id="altChunk1" TargetMode="Internal" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/afChunk" Target="import.htm" />
 …
</Relationships>

The corresponding relationship part item shows that the file to be imported is located [next](next.docx) to the main document and is named import.htm. end example]

|  |
| --- |
| Parent Elements |
| [body](body.docx) (§); [comment](comment.docx) (§); [docPartBody](docPartBody.docx) (§); [endnote](endnote.docx) (§); [footnote](footnote.docx) (§); [ftr](ftr.docx) (§); [hdr](hdr.docx) (§); [tc](tc.docx) (§); [txbxContent](txbxContent.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [altChunkPr](altChunkPr.docx) (External Content Import Properties) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| [id](id.docx) (Relationship to Part)Namespace: .../officeDocument/2006/relationships | Specifies the relationship ID to a specified part containing alternate content for import. If the specified relationship does not match the [type](type.docx) required by the parent element, then this document shall be considered to be invalid.[Example: Consider an [XML](XML.docx) element which has the following [id](id.docx) attribute:<… r:id="rId10" />The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent [XML](XML.docx) element. end example]The possible values for this attribute are defined by the ST\_RelationshipId simple [type](type.docx) (§). |

The following [XML](XML.docx) Schema fragment defines the contents of this element:

<complexType [name](name.docx)="CT\_AltChunk">

 <sequence>

 <element name="[altChunkPr](altChunkPr.docx)" [type](type.docx)="CT\_AltChunkPr" minOccurs="0" maxOccurs="1"/>

 </sequence>

 <attribute ref="r:id" use="optional"/>

</complexType>