### fldChar (Complex Field Character)

This element specifies the presence of a complex field character at the current location in the parent run. A complex field character is a special character which delimits the [start](start.docx) and end of a complex field or separates its field codes from its current field result.

A complex field is defined via the use of the two required complex field characters: a [start](start.docx) character, which specifies the beginning of a complex field within the document content; and an end character, which specifies the end of a complex field. This syntax allows multiple fields to be embedded (or "nested") within each other in a document.

As well, because a complex field may specify both its field codes and its current [result](result.docx) within the document, these two items are separated by the optional [separator](separator.docx) character, which defines the end of the field codes and the beginning of the field contents. The omission of this character shall be used to specify that the contents of the field are entirely field codes (i.e. the field has no result).

[Example: Consider the following complex field definition within a WordprocessingML document:

<w:fldChar w:[type](type.docx)="[start](start.docx)" />  
<w:[r](r.docx)>  
 <w:[instrText](instrText.docx)>AUTHOR</w:[instrText](instrText.docx)>  
</w:[r](r.docx)>  
<w:fldChar w:[type](type.docx)="separate" />  
<w:[r](r.docx)>  
 <w:[t](t.docx)>Rex Jaeschke</w:[t](t.docx)>  
</w:[r](r.docx)>  
<w:fldChar w:[type](type.docx)="end" />

The three fldChar elements specify:

* The beginning of the field, using the [type](type.docx) attribute value of [start](start.docx)
* The [separator](separator.docx) [between](between.docx) the field codes and the current field results, using the [type](type.docx) attribute value of separate
* The end of the field, using the [type](type.docx) attribute value of end

end example]

If a complex field character is located in an inappropriate location in a WordprocessingML document, then its presence shall be ignored and no field shall be present in the resulting document when displayed. Also, if a complex field is not closed before the end of a document story, then no field shall be generated and each individual run shall be processed as if the field characters did not exist (i.e. the contents of all field code run content shall not be displayed, and the field results shall be displayed as literal text).

[Example: Consider the following WordprocessingML document:

<w:[body](body.docx)>  
 <w:[p](p.docx)>  
 <w:fldChar w:[type](type.docx)="[start](start.docx)" />  
 <w:[r](r.docx)>  
 <w:[instrText](instrText.docx)>AUTHOR</w:[instrText](instrText.docx)>  
 </w:[r](r.docx)>  
 <w:fldChar w:[type](type.docx)="separate" />  
 <w:[r](r.docx)>  
 <w:[t](t.docx)>Rex Jaeschke</w:[t](t.docx)>  
 </w:[r](r.docx)>  
 </w:[p](p.docx)>  
</w:[body](body.docx)>

The complex field is technically incorrect since no end character exists in the main document story. The resulting content shall be interpreted as though no field characters exist, resulting in only the literal text Rex Jaeschke being displayed in the document. end example]

|  |
| --- |
| Parent Elements |
| [r](r.docx) (§); [r](r.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [ffData](ffData.docx) (Form Field Properties) | § |
| [fldData](fldData.docx) (Custom Field Data) | § |
| [numberingChange](numberingChange.docx) (Previous [Numbering](Numbering.docx) Field Properties) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| [dirty](dirty.docx) (Field Result Invalidated) | Specifies that this field has been flagged by an application to indicate that its current results are invalid (stale) due to other modifications made to the document, and these contents should be updated before they are displayed if this functionality is supported by the [next](next.docx) processing application.  [Rationale: This functionality allows applications with limited subsets of the full functionality of this Office Open [XML](XML.docx) Standard to process Word Open [XML](XML.docx) documents without needing to understand and update all fields based on the semantics for their field codes.  For example, an application can add a new paragraph and flag the [table](table.docx) of contents as [dirty](dirty.docx), without needing to understand anything about how to recalculate that field's content. end rationale]  If this attribute is omitted, then its value shall be assumed to be false. If the [type](type.docx) of the current field character is not [start](start.docx), then his setting may be ignored.  [Example: Consider the following WordprocessingML for a complex field:  <w:fldChar w:[type](type.docx)="[start](start.docx)" w:[dirty](dirty.docx)="true"/> <w:[r](r.docx)>  <w:[instrText](instrText.docx)>TOC /l 1-3</w:[instrText](instrText.docx)> </w:[r](r.docx)> <w:fldChar w:[type](type.docx)="separate"/> …  The [dirty](dirty.docx) attribute value of true specifies that the contents of this field are no longer current based on the contents of the document, and should be recalculated whenever an application with this functionality reads the document. end example]  The possible values for this attribute are defined by the [ST\_OnOff](ST_OnOff.docx) simple [type](type.docx) (§). |
| fldCharType (Field Character Type) | Specifies the [type](type.docx) of the current complex field character in the document.  [Example: Consider the following WordprocessingML for a complex field character:  … <w:fldChar w:[type](type.docx)="separate" /> …  The [type](type.docx) attribute value of separate specifies that this is a complex field [separator](separator.docx) character; therefore it is being used to separate the field codes from the field contents in a complex field. end example]  The possible values for this attribute are defined by the [ST\_FldCharType](ST_FldCharType.docx) simple [type](type.docx) (§). |
| fldLock (Field Should Not Be Recalculated) | Specifies that the parent complex field shall not have its field [result](result.docx) recalculated, even if an application attempts to recalculate the results of all fields in the document or a recalculation is explicitly requested.  If this attribute is omitted, then its value shall be assumed to be false. If the [type](type.docx) of the current field character is not [start](start.docx), then his setting may be ignored.  [Example: Consider the following WordprocessingML for a complex field:  <w:fldChar w:[type](type.docx)="[start](start.docx)" w:fldLock="true"/>  … <w:fldChar w:[type](type.docx)="separate"/> <w:[r](r.docx)>  <w:[t](t.docx)>field result</w:[t](t.docx)> </w:[r](r.docx)> <w:fldChar w:[type](type.docx)="end" />  The fldLock attribute value of true specifies that the contents of this field shall remain field [result](result.docx) regardless of the actual [result](result.docx) of the current field codes. end example]  The possible values for this attribute are defined by the [ST\_OnOff](ST_OnOff.docx) simple [type](type.docx) (§). |

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

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

<choice>

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

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

<element [name](name.docx)="[numberingChange](numberingChange.docx)" [type](type.docx)="CT\_TrackChangeNumbering" minOccurs="0"/>

</choice>

<attribute [name](name.docx)="fldCharType" [type](type.docx)="[ST\_FldCharType](ST_FldCharType.docx)" use="required"/>

<attribute [name](name.docx)="fldLock" [type](type.docx)="[ST\_OnOff](ST_OnOff.docx)"/>

<attribute [name](name.docx)="[dirty](dirty.docx)" [type](type.docx)="[ST\_OnOff](ST_OnOff.docx)"/>

</complexType>