#### General formatting

general-formatting-switch:
\\* [ " ] switch-argument [ " ]

A general-formatting-switch specifies a variety of formats for a numeric or text result. If the [result](result.docx) [type](type.docx) of a field does not correspond to the [format](format.docx) specified, this switch has no effect.

Quotation marks are required around switch-argument if it contains white space; otherwise, they are optional.

A switch-argument is made up of a series of [picture](picture.docx) items.

|  |
| --- |
| General Formatting Switch Arguments |
| Switch Argument | Description |
| AIUEO | Formats a numeric [result](result.docx) using hiragana characters in the traditional a-i-u-e-o order. [Example: 1 \\* AIUEO results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of aiueo. |
| ALPHABETIC | Formats a numeric [result](result.docx) as one or more occurrences of an uppercase alphabetic Latin character. Value 1 results in the letter A, value 2 results in the letter B, and so on up to value 26, which results in the letter Z. For values greater than 26, 26 is repeatedly subtracted from the value until the [result](result.docx) is 26 or less. The [result](result.docx) value determines which letter to use, and the same letter is repeated for each time 26 was subtracted from the original value. [Example: =54 \\* ALPHABETIC results in "BBB" as subtracting 26 from 54 two times, results in the value 2, which is represented by the letter B. end example] |
| alphabetic | Formats a numeric [result](result.docx) as one or more occurrences of an lowercase alphabetic Latin character. Value 1 results in the letter a, value 2 results in the letter b, and so on up to value 26, which results in the letter z. For values greater than 26, 26 is repeatedly subtracted from the value until the [result](result.docx) is 26 or less. The [result](result.docx) value determines which letter to use, and the same letter is repeated for each time 26 was subtracted from the original value. [Example: =52 \\* alphabetic results in "zz" as subtracting 26 from 52 one time, results in the value 26, which is represented by the letter z.. end example] |
| Arabic | Formats a numeric [result](result.docx) using Arabic cardinal numerals. [Example: For page 123, PAGE \\* Arabic results in "123". end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of decimal. |
| ARABICABJAD | Formats a numeric [result](result.docx) using ascending Abjad numerals. [Example: 12 \\* ARABICABJAD results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of arabicAbjad. |
| ARABICALPHA | Formats a numeric [result](result.docx) using characters in the Arabic alphabet. [Example: 12 \\* ARABICABJAD results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of arabicAlpha. |
| ArabicDash | Formats a numeric [result](result.docx) using Arabic cardinal numerals, with a prefix of "- " and a suffix of " -". [Example: For page 123, PAGE \\* ArabicDash results in "- 123 -". end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of numberInDash. |
| BAHTTEXT | Formats a numeric [result](result.docx) using the given Thai style. [Example: 1 \\* BAHTTEXT results in . end example] |
| Caps | Capitalizes the first letter of each word. [Example: [USERNAME](USERNAME.docx) "mary smith" \\* Caps results in "Mary Smith", whereas [USERNAME](USERNAME.docx) "marysmith" \\* Caps results in "Marysmith". end example] |
| CardText | Formats a numeric [result](result.docx) as lowercase cardinal text. [Example: For page 123, PAGE \\* CardText results in "one hundred twenty-three". end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of cardinalText. |
| CHARFORMAT | See the discussion following this table. |
| CHINESENUM1 | Formats a numeric [result](result.docx) using ascending numbers from the Chinese counting system. [Example: 10 \\* CHINESENUM1 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of chineseCounting. |
| CHINESENUM2 | Formats a numeric [result](result.docx) using sequential numbers from the Chinese simplified legal format. [Example: 123 \\* CHINESENUM2 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of chineseLegalSimplified. |
| CHINESENUM3 | Formats a numeric [result](result.docx) using sequential numbers from the Chinese counting thousand system. [Example: 10 \\* CHINESENUM3 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of chineseCountingThousand. |
| CHOSUNG | Formats a numeric [result](result.docx) using sequential numbers from the Korean Chosung format. [Example: 1 \\* CHOSUNG results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of chosung. |
| CIRCLENUM | Formats a numeric [result](result.docx) using decimal [numbering](numbering.docx) enclosed in a circle, using the enclosed alphanumeric glyph character for numbers in the range 1–20. For non-negative numbers outside this range, formats them as with ARABIC. [Example: 12 \\* CIRCLENUM results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of decimalEnclosedCircle. |
| DBCHAR | Formats a numeric [result](result.docx) using double-byte Arabic numbering. [Example: 123 \\* DBCHAR results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of decimalFullWidth. |
| DBNUM1 | Formats a numeric [result](result.docx) using sequential digital ideographs, using the appropriate character. [Example: 12 \\* DBNUM1 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of ideographDigital. |
| DBNUM2 | Formats a numeric [result](result.docx) using sequential numbers from the Korean counting system. [Example: 12 \\* DBNUM2 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of koreanCounting. |
| DBNUM3 | Formats a numeric [result](result.docx) using sequential numbers from the Japanese legal counting system. [Example: 12 \\* DBNUM3 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of japaneseLegal. |
| DBNUM4 | Formats a numeric [result](result.docx) using sequential numbers from the Japanese digital ten thousand counting system. [Example: 12 \\* DBNUM4 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of japaneseDigitalTenThousand. |
| DollarText | Formats a numeric [result](result.docx) in the following form:integer-part-as-cardinal-text and nn/100The fractional part is rounded to two decimal places, nn, and is formatted using Arabic cardinal numerals. [Example: =1234.567 \\* DollarText results in "one thousand two hundred thirty-four and 57/100". end example] |
| FirstCap | Capitalizes the first letter of the first word. [Example: [USERNAME](USERNAME.docx) "mary smith" \\* FirstCap results in "Mary smith". end example] |
| GANADA | Formats a numeric [result](result.docx) using sequential numbers from the Korean Ganada format. [Example: 12 \\* GANADA results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of ganada. |
| GB1 | Formats a numeric [result](result.docx) using decimal [numbering](numbering.docx) followed by a period, using the enclosed alphanumeric glyph character. [Example: 12 \\* GB1 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of decimalEnclosedFullstop. |
| GB2 | Formats a numeric [result](result.docx) using decimal [numbering](numbering.docx) enclosed in parenthesis, using the enclosed alphanumeric glyph character. [Example: 12 \\* GB2 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of decimalEnclosedParen. |
| GB3 | Formats a numeric [result](result.docx) using decimal [numbering](numbering.docx) enclosed in a circle, using the enclosed alphanumeric glyph character. Once the specified sequence reaches 11, the numbers may be replaced with non-enclosed equivalents. [Example: 12 \\* GB3 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of decimalEnclosedCircleChinese. |
| GB4 | Formats a numeric [result](result.docx) using decimal [numbering](numbering.docx) enclosed in a circle, using the enclosed alphanumeric glyph character. Once the specified sequence reaches 11, the numbers may be replaced with non-enclosed equivalents. [Example: 12 \\* GB4 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of ideographEnclosedCircle. |
| HEBREW1 | Formats a numeric [result](result.docx) using Hebrew numerals. [Example: 123 \\* HEBREW1 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of hebrew1. |
| HEBREW2 | Formats a numeric [result](result.docx) using the Hebrew alphabet. [Example: 123 \\* HEBREW2 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of hebrew2. |
| Hex | Formats the numeric [result](result.docx) using uppercase hexadecimal digits. [Example: For page 355, PAGE \\* Hex results in "FF". end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of hex. |
| HINDIARABIC | Formats a numeric [result](result.docx) using Hindi numbers. [Example: 123 \\* HINDIARABIC results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of hindiNumbers. |
| HINDICARDTEXT | Formats a numeric [result](result.docx) using sequential numbers from the Hindi counting system. [Example: 123 \\* HINDICARDTEXT results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of hindiCounting. |
| HINDILETTER1 | Formats a numeric [result](result.docx) using Hindi vowels. [Example: 123 \\* HINDILETTER1 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of hindiVowels. |
| HINDILETTER2 | Formats a numeric [result](result.docx) using Hindi consonants. [Example: 123 \\* HINDILETTER2 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of hindiConsonants. |
| IROHA | Formats a numeric [result](result.docx) using the Japanese iroha. [Example: 12 \\* IROHA results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of iroha. |
| KANJINUM1 | Formats a numeric [result](result.docx) using a Japanese style using sequential digital ideographs, using the appropriate character. [Example: 12 \\* KANJINUM1 results in . end example] |
| KANJINUM2 | Formats a numeric [result](result.docx) using the Japanese counting system. [Example: 12 \\* KANJINUM2 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of japaneseCounting. |
| KANJINUM3 | Formats a numeric [result](result.docx) using the Japanese legal counting system. [Example: 12 \\* KANJINUM3 results in . end example] |
| Lower | All letters are lowercase. [Example: [USERNAME](USERNAME.docx) "Mary Smith" \\* Lower results in "mary smith". end example] |
| MERGEFORMAT | See the discussion following this table. |
| Ordinal | Formats a numeric [result](result.docx) using lowercase ordinal Arabic numerals. [Example: =32 \\* Ordinal results in "32nd". end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of ordinal. |
| OrdText | Formats a numeric [result](result.docx) as lowercase ordinal text. Apart from being used to round off the whole number part, the fractional part is not used. [Example: =1234.567 \\* OrdText results in "one thousand two hundred thirty-fifth". end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of ordinalText. |
| Roman | Formats a numeric [result](result.docx) using uppercase Roman numerals. [Example: For page 123, PAGE \\* Roman results in "CXXIII". end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of upperRoman. |
| roman | Formats a numeric [result](result.docx) using lowercase Roman numerals. [Example: For page 123, PAGE \\* roman results in "cxxiii". end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of lowerRoman. |
| SBCHAR | Formats a numeric [result](result.docx) using single-byte Arabic numbering. [Example: 123 \\* SBCHAR results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of decimalHalfWidth. |
| THAIARABIC | Formats a numeric [result](result.docx) using Thai numbers. [Example: 123 \\* THAIARABIC results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of thaiNumbers. |
| THAICARDTEXT | Formats a numeric [result](result.docx) using sequential numbers from the Thai counting system. [Example: 123 \\* THAICARDTEXT results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of thaiCounting. |
| THAILETTER | Formats a numeric [result](result.docx) using Thai letters. [Example: 123 \\* THAILETTER results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of thaiLetters. |
| Upper | All letters are uppercase. [Example: [USERNAME](USERNAME.docx) "Mary Smith" \\* Upper results in "MARY SMITH". end example] |
| VIETCARDTEXT | Formats a numeric [result](result.docx) using Vietnamese numerals. [Example: 12 \\* VIETCARDTEXT results in [i](i.docx). end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of vietnameseCounting. |
| ZODIAC1 | Formats a numeric [result](result.docx) using sequential numerical traditional ideographs. [Example: 1 \\* ZODIAC1 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of ideographTraditional. |
| ZODIAC2 | Formats a numeric [result](result.docx) using sequential zodiac ideographs. [Example: 1 \\* ZODIAC2 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of ideographZodiac. |
| ZODIAC3 | Formats a numeric [result](result.docx) using sequential traditional zodiac ideographs. [Example: 1 \\* ZODIAC3 results in . end example]Corresponds to an [ST\_NumberFormat](ST_NumberFormat.docx) enumeration value of ideographZodiacTraditional. |

The general formatting switch argument CHARFORMAT is used to set the visual appearance of a field's value by setting the first run in that field's field-type name to the desired state using any of the elements that can be directly nested inside the run property element, [rPr](rPr.docx). [Example: In a US-English context, on January 4, 2006, the field DATE \\* CHARFORMAT results in "1/4/2006". However, if the D in [DATE](DATE.docx) is made bold, the field DATE \\* CHARFORMAT results in "1/4/2006". If that D is made italic, the field DATE \\* CHARFORMAT results in "1/4/2006". If that D is made bold, underlined, and red, the field DATE \\* CHARFORMAT results in "1/4/2006".

The [XML](XML.docx) for the bold, underlined, red case is as follows:

<w:[r](r.docx)>
 <w:[fldChar](fldChar.docx) w:fldCharType="begin"/>
</w:[r](r.docx)>

<w:[r](r.docx)>
 <w:[instrText](instrText.docx) xml:space="preserve"> </w:[instrText](instrText.docx)>
</w:[r](r.docx)>

<w:[r](r.docx) …>
 <w:[rPr](rPr.docx)>
 <w:[b](b.docx)/>
 <w:[color](color.docx) w:val="ED1C24"/>
 <w:[u](u.docx) w:val="single"/>
 </w:[rPr](rPr.docx)>
 <w:[instrText](instrText.docx)>D</w:[instrText](instrText.docx)>
</w:[r](r.docx)>

<w:[r](r.docx)>
 <w:[instrText](instrText.docx) xml:space="preserve">ATE </w:[instrText](instrText.docx)>
</w:[r](r.docx)>

<w:[r](r.docx)>
 <w:[fldChar](fldChar.docx) w:fldCharType="separate"/>
</w:[r](r.docx)>

<w:[r](r.docx) …>
 <w:[rPr](rPr.docx)>
 <w:[b](b.docx)/>
 <w:[color](color.docx) w:val="ED1C24"/>
 <w:[u](u.docx) w:val="single"/>
 </w:[rPr](rPr.docx)>
 <w:[t](t.docx)>1/4/2006</w:[t](t.docx)>
</w:[r](r.docx)>

<w:[r](r.docx)>
 <w:[fldChar](fldChar.docx) w:fldCharType="end"/>
</w:[r](r.docx)>

end example]

If a [format](format.docx) specified directly in the first run of a field's field-type name conflicts with a general formatting switch, the general formatting switch is ignored. [Example: If the first run is set in small [caps](caps.docx) and the switch \\* Lower is also used, that switch is ignored. end example]

The general formatting switch argument MERGEFORMAT is used to apply formatting directly to part of a [result](result.docx) such that when that [result](result.docx) is updated, the formatting is preserved. The formatting is expressed in [XML](XML.docx) using an [rPr](rPr.docx) element on the run that contains the most recently updated field result. [Example: Consider the following field:

[TIME](TIME.docx) \@ "HH:mm:ss" \\* MERGEFORMAT

When it is updated, the [result](result.docx) might be 12:22:27, for example. If the seconds part of the displayed field [result](result.docx) is underlined, as in 12:22:27, when that field is [next](next.docx) updated, the seconds underlining is preserved. If MERGEFORMAT is omitted, the [rPr](rPr.docx) element on the run that contains the most recently updated field [result](result.docx) is ignored.

The [XML](XML.docx) generated for this is:

<w:[r](r.docx)>
 <w:[fldChar](fldChar.docx) w:fldCharType="begin"/>
</w:[r](r.docx)>

<w:[r](r.docx)>
 <w:[instrText](instrText.docx) xml:space="preserve"> [TIME](TIME.docx) \@ "HH:mm:ss" \\* MERGEFORMAT </w:[instrText](instrText.docx)>
</w:[r](r.docx)>

<w:[r](r.docx)>
 <w:[fldChar](fldChar.docx) w:fldCharType="separate"/>
</w:[r](r.docx)>

<w:[r](r.docx) …>
 <w:[t](t.docx)>17:02:</w:[t](t.docx)>
</w:[r](r.docx)>

<w:[r](r.docx) …>
 <w:[rPr](rPr.docx)>
 <w:[u](u.docx) w:val="single"/>
 </w:[rPr](rPr.docx)>
 <w:[t](t.docx)>32</w:[t](t.docx)>
</w:[r](r.docx)>

<w:[r](r.docx)>
 <w:[fldChar](fldChar.docx) w:fldCharType="end"/>
</w:[r](r.docx)>

end example]