Many people access the material from this web site daily. Most just take what they want and run. That's OK provided they are not selling on the material as their own; however if your productivity gains from the material you have used, a donation from the money you have saved would help to ensure the continued availability of the resource. Click the appropriate button above to access PayPal.

Formatting Word fields with switches

Word provides the means to format the contents of fields inserted into the document, by the use of 'switches' or masks. The full range of these switches can be found at the end of this page.

The following tips are intended to cover some of the less obvious issues.

Where field combinations are inserted manually, the field boundaries { } are inserted with CTRL+F9 and not typed directly from the keyboard.

To edit the field switches, toggle between the field display and the result with ALT+F9 or us add a View Field Codes toolbar button to toggle the display.

Select the item from the list and drag it to a toolbar:
The F9 key will update the field content to reflect the changes, or you could use macro code in the document.

Microsoft changed the way office programs are connected for the purposes of mail merging with Office 2002 and this change was maintained in Office 2003/2007/2010. Word is now expected to provide the formatting. It is still possible to revert to the old way of doing things. To this end, see the Excel Data section of "Mail Merge to Labels with Office XP". Number fields (For negative numbers in calculations, see the separate entry.)

Word's formatting switches can handle numbers up to 14 digits after which it rounds off the cause problems when serial numbers contain more than 14 digits and the last digits are important!

Take the example of the number 3456 7123 0987 7652, assembled as four blocks of four digits: this may be stored as a single number 3456712309877652.

Inserting the Mergefield

\{ MERGEFIELD Serial_No \}

will produce:

3456712309877652

Logic, and Word's help on the subject, suggests the addition of a numeric switch thus:

\{ MERGEFIELD Serial_No \# "0000 0000 0000 0000"\}

will produce the required result, but in fact what happens is that the number is rounded to 1
The answer should be to split up the field into two parts with the aid of the SET field to assign parts of the data to bookmarks, then to display the bookmarks with the switches.

```
{ SET A "\{ MERGEFIELD Serial_No \} " } { SET B "\{ =INT(A / 100000000) =A - (E * 100000000) \} " } { E \# "0000 0000" } { C \# "0000 0000" }
```

The result is:

```
3456 7123 0987 7700
```

unless there is a zero as the penultimate number in the sequence, where thanks to a bug in Microsoft’s mathematics, the last number is rounded down to zero also.

```
From the database - 2234567890124506
From the calculation - 2234 5678 9012 4500
```

It requires a little brute force to overcome this by adding the last digit back into the bookmark usually known, this means that you have to add a calculation for each of the 9 possible alternate constructions is never going to look elegant. But it does work.

```
{ SET A "\{ Mergefield Serial_No \} " } { SET B "\{ =INT(A / 100000000) \} " } { IF \{ A \} = "*01" \{ SET C \{ =C+1 \} \} \} { IF \{ A \} = "C+2 \} } " } { IF \{ A \} = "*03" \{ SET C \{ =C+3 \} \} \} { IF \{ A \} = "C+4 \} } " } { IF \{ A \} = "*05" \{ SET C \{ =C+5 \} \} \} { IF \{ A \} = "*07" \{ SET C \{ =C+7 \} \} \} { IF \{ A \} = "*09" \{ SET C \{ =C+9 \} \} \} { B \# "0000 0000" }
```

Corrected - 2234 5678 9012 4506

Thanks to MVP Peter Jamieson for suggesting the above solution.

**Insert field information and associated text conditionally**

Sometimes when merging data you may need to include additional text only when the associated field has content, or you may wish to insert a particular field without leaving a space when there is no content in that field.

The simplest method is to use a conditional field or fields to place the data and associated text.

In the following example using data from an Outlook contacts list, I have prepared a simple merge which includes various fields and associated text depending on whether the fields have content.

With the three records used for the illustration, one of the conditionally included fields and text does not appear. In the first of the three illustrations, note the positions of the quote marks “” which set the limit...
For the sake of clarification, for the second illustration I have switched on the display of formatting marks (CTRL+* or click the ¶ button on the toolbar).

In the third of the pictures is the result of the merge.
Conditionally Insert a graphical image

Where images are included as part of a conditional (IF) field construction, the IncludePicture field does not work as anticipated. The results of the conditional field are hard coded into the merge, which does not provide the opportunity to update, following a merge to a new document. (See also Mail Merge Graphics)

Thus conventional wisdom would suggest that:

{ IF { Mergefield Fieldname } <> "" INCLUDEPICTURE D:\My Documents\Pictures\{ Mergefield Fieldname } " " }

should insert the picture identified from the mergefield 'Fieldname' and nothing when the field is empty. Unfortunately this is not the case.

The following workaround demonstrates the insertion of a blank image where the condition is not met (save e.g. a 1 pixel transparent image as Blank.jpg to give the effect of nothing being inserted - or download such an image by clicking the link). Note that INCLUDEPICTURE is the primary field and the condition is achieved as part of its content rather than the other way around.

{ INCLUDEPICTURE { IF { Mergefield "Fieldname" } <> "" "D:\My Documents\Pictures\{ Mergefield 'Fieldname' } "D:\My Documents\My Pictures\Blank.jpg" } " " }

You can insert the required picture into the result of the conditional field directly and that will work also.
an airmail image to envelopes intended for other countries, you could use the following:

```plaintext
{ IF{ Mergefield Country \* upper } <> "CYP" }
```

which will insert the **airmail image** when the country field is other than Cyprus (where I live).

## Conditionally insert a Styleref Field in a header/footer

Styleref fields are used to display the content of a nominated paragraph or character style—frequently found in page header or footer.

**Condition**

```
{ STYLEREF "Heading 1" }
```

**Result**

`This is Heading 1`

If the nominated style does not exist, then the following error message is displayed:

```
Header
{ STYLEREF "Heading 1" }< >" Error!*" { STYLEREF }
```

`This isn't!`

This is not very helpful when building a template that may not have this particular style. It is conditionally search for the error message in the field result and only insert the Styleref field when the error message is not present. You can use a wildcard in the search pattern as shown.

```
Header
{ IF{ STYLEREF "Heading 1" } <> "Error!*" { STYLEREF } }
```

`This is Heading 1`

This produces the required content when the style is present......
but nothing when it isn't

Insert quotation marks in a conditional merge field result

Peter Jamieson also came up with the following approach to the problem of inserting quote result of a conditional mail merge.

In a recent newsgroup question the questioner wanted to insert Exhibit “A” when a certain field was empty.

The logical code for this would appear to be

```
{ IF{ Mergefield Test } = "" "Exhibit 'A':" }
```

but this chops off the text at the first quote mark:

```
Exhibit
```

You may then consider 'escaping' the quote mark with a \ e.g.:

```
{ IF{ Mergefield Test } = "" "Exhibit \"A\":" }
```

but unfortunately Word inserts the quotes but also the slash

```
Exhibit \"A\":
```

One solution is to set the quote and the slash as a bookmark using a SET field and then in using REF fields (the REF part is optional) thus:

```
{ SET q "\"" IF{ Mergefield Test } = "" "Exhibit { q }A{ q
```

An alternative approach is to use a pair of Quote fields to place the quotes characters directly e.g.

```
{ IF{ Mergefield Test } = "A" "Exhibit { Quote 34 }A{ Quote 3
```

either of which produces the desired result:
Testing for a value in one of several fields.

Sometimes when preparing a mail merge document, you may wish to insert text based upon whether a value has been inserted in one of a number of fields.

The following example tests for whether Field1 contains "A", or Field2 contains "B", or Field3 contains "C". If any or all of those fields meet the condition, "True" is entered in the merge document. If not, "False" is entered. In a real merge situation True or False can be replaced by whatever you wish.

You could test for any content using this method, which uses a collection of conditional fields that insert a value of 1 or 0 into a calculation according to whether the value required is contained in the field. The results are added together. If the result of that calculation is 1 or more, then one of more of those fields contains the required data.

Thus:

```
{ IF{ (=({ IF{ MERGEFIELD Field1 } = "A" 1 0 } + { IF{ MERGEFIELD Field2 } = "B" 1 0 } + { IF{ MERGEFIELD Field3 } = "C" 1 0 } }) >= 1 } "True" "False" }
```

The following shows the content of the three fields and the result of the merge calculation.

```
Field1 = A  Field2 = B  Field3 = C  Result = True
Field1 = B  Field2 = V  Field3 = j  Result = False
Field1 = V  Field2 = B  Field3 = T  Result = True
Field1 = Q  Field2 = L  Field3 = C  Result = True
Field1 = K  Field2 = H  Field3 = P  Result = False
```

Convert upper case data to lower case with the first word capitalized.

The \FirstCap switch is normally used to format fields to lower case with the first word capitalized as in a normal sentence. However with some upper case data sources, the switch doesn't appear to do anything. The solution here is to convert the field to lower case first, by using a \Lower switch, thus:

```
{ Mergefield Fieldname \*Lower \*FirstCap }
```

The \Lower switch is also useful when used in conjunction with data - e.g. user forms where entered in upper or lower case, but you wish to test for the entered content. Let's take the example of a bookmark from a user form which requires YES to be entered in the document when the content of the bookmark is 'yes' or 'YES' or 'Yes'. In this case add the \Lower switch to the REF field and test for y followed by any other characters.

```
{ IF{ REF bookmarkname \*lower } = 'y\*n"YES" "NO" }
```

Format cash amounts to 2 decimal places

Sometimes when merging data, comprising cash amounts, especially when importing from displays up to 14 decimal places. This is caused by the way Excel handles numeric data in tamed by the addition of a switch to limit the data to 2 decimal places. e.g.
are produced by the following 2 fields respectively

\[
\{ \text{MERCEFIELD Amount} \} \quad \{ \text{MERCEFIELD Amount} \# "\$,\,0.00 \} \#
\]

In the second field, the switch \# "$,0.00;($,0.00)" includes a comma, which provides for the thousands and millions e.g.

and an optional currency symbol - here a dollar sign. The numeric mask can also be expressed ($#,00)" where the '#' will suppress the 0 for amounts less than a dollar.

One variation I used to use a lot is for Cyprus Pound currency amounts (Cyprus has since adopted the Euro currency):

\[
\{ \text{MERCEFIELD Amount} \# "\cy\,0.00;\cy\,0.00 \} \#
\]

The section after the semi colon dictates what to do with negative amounts - here they are coloured red. The semi-colon and following section are optional.

**Format cash amounts to 0 decimal places**

Sometimes with decimal amounts, when the amount to the right of the decimal is zero, you whole number, without the decimal, whilst retaining the amount to the right of the decimal > greater than zero e.g. $12.00, should display as $12, but $12.95 should retain its decimal achieved with a simple switch and therefore you need to insert a conditional field to display according to what comes after the decimal. In the following example I have suppressed the amounts (see section below).

\[
\{ \text{IF } \text{MOD}(\text{MERGEFIELD Amount};1) = 0 \text{\"\text{MERGEFIELD Amount} \# "$,0.00 \""} \}
\]

**Formatting cash amounts in words**

Word provides special switches - *cardtext* and *dollartext* (detailed in the table at the enc are somewhat limiting if you want to express amounts in words. It is, however, possible to words using conditional, and formula fields with the *cardtext* switch.

The following example takes the decimal output from a form field, using the default form fie **Text1**. The fields can be easily adapted to use mergefields and any decimal currency.

\[
\{ \text{SET x} \{ \text{Ref Text1} \} \}
\{ \text{SET y} \{ \text{INT}(x) \} \}
\{ \text{SET z} \{ (x) - (y) * 100 \} \}
\{ \text{IF} \{ y \} < 0 \{ y \"\text{Cardtext} \text{dollartext} \}, \{ \text{IF} \{ y \} > 1 \"\#\" \}, \{ \text{IF} \{ y \} < 0 \"\text{Cardtext} \text{cent} \}, \{ \text{IF} \{ z \} > 0 \"\text{Cardtext} \text{dollartext} \}, \{ \text{IF} \{ z \} > 0 \"\text{Cardtext} \text{cent} \} \}
\]

Where **Text1** contains the amount **1.00** - the fields produce **one dollar**

Where **Text1** contains the whole number **2.00** (or greater) the fields produce **two (or more**

Where **Text1** contains **.01** the fields produce **one cent**

Where **Text1** contains **.50** the fields produce **fifty cents**

Where Text1 contains **5.75** the fields produce **five dollars and seventy-five cents** etc
The method has problems with numbers over 1 million, so as a workaround, you could use lines of

```plaintext
{ SET r { =MOD({ x },{ r })/1000000 ) }
{ SET m { =INT(({ x }-{ r }))/1000000 ) }
{ SET c { =MOD(({ x }),{ -INT(({ x } )}) }) }
{ IF m } = 0 "=INT({ r })"="Cardtext } Dollars"{ =INT({ m })}"Currency"
   =INT({ r })"="Cardtext } Dollars"
{ IF c } <> 0 "and { =c } * 100 "="Cardtext } Cent" { IF { =c } "
```

### Percentages

Frequently, percentages will display as decimals - e.g. 41% from the data may display as 0.41354836739 when merged into Word. To display the decimal as a percentage, you should create a calculated field:

```
Suppress field display for numbers <=0
```

You may wish to suppress the field result altogether, when the numeric content is 0 or a negative number can be achieved with a conditional field, or more simply with a variation on the above numeric switch e.g.:

```
#"$,0.00;negative;zero"
```

Any positive number will display in a dollar format (the dollar sign being optional again). Zero will display nothing.

The two semi colons at the end determine what happens with negative numbers and zeros. Consider the principle:

```
# "$,0.00;negative;zero"
```

In other words for a number greater than zero the result is formatted as $,0.00

For numbers less than zero - the number is formatted as negative - in this case "negative"

For a result of zero - the number is formatted as zero - and again in this example will be entered as "zero"

Let's say you have a data file with a single field called 'Number', which contains the following records:

-1
0
1

Merging that field - `{Mergefield Number # "$,0.00;negative;zero"}` will result in:

- negative
- zero
- $1.00

### Calculated Form Fields - suppress zero

The currency switches available in a calculated form field are limited, but you can suppress calculated form field, by adding a switch to the calculation part of the field. Toggle the display...
Then add the switch.

```plaintext
Quantity \{ FORMTEXT \} \quad Price \{ FORMTEXT \} \quad Sub Total
\quad =Quantity \ast Price \# "\£0.00;\£0.00;"
```

Alternatively abandon the calculated form field and use instead a Word formula field thus:

```plaintext
Quantity \{ FORMTEXT \} \quad Price \{ FORMTEXT \} \quad Sub Total
\quad \ast \{ Price \\# "\£0.00;\£/0.00;"
```

Then toggle back or lock the form to display the results, which would be the same in either

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Price</th>
<th>Sub Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>£30.00</td>
<td>£60.00</td>
</tr>
<tr>
<td>3</td>
<td>£30.00</td>
<td>£90.00</td>
</tr>
<tr>
<td>0</td>
<td>£30.00</td>
<td>0</td>
</tr>
</tbody>
</table>

**US Zip Codes**

Five digit zip codes are reported as dropping the leading zero when merging from Excel/Access files. A simple numeric switch should fix that one

```plaintext
\{ MERGEFIELD Zip\# "00000" \}
```

Formatting problem: the zip codes in my data source are 5-digit or 9-digit. What I'd like to do is apply a number format that will convert all ZIP codes to ZIP + four format, i.e. 12345 becomes 12345-0000 and 12345-0000 becomes 12345-6789

This requires the use of a conditional field which tests whether the field has five or nine digits and applies appropriate formats.

```plaintext
\{ IF\{ MERGEFIELD Zip \} > 99999 \"\{ MERGEFIELD Zip\# "00000\-MERGEFIELD Zip\# "00000\}-0000\}" \}
```

<table>
<thead>
<tr>
<th>Zip = 12345</th>
<th>Zip = 12345-6789</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345-0000</td>
<td>12345-6789</td>
</tr>
</tbody>
</table>

If you wish to display the 5 digit zips without the `-0000` suffix used in the above example, and show nothing when the field is empty then modify the formatting switch thus:
Calculations cannot be performed on text, so in order for the above to be of use, the data should be entered as a continuous number of 5 or 9 digits. 9 digit numbers can be formatted using Excel’s cell formatting as a 5+4 Zip, which will provide the hyphen in the Excel display whilst retaining the number in the raw data.

Telephone Numbers

Formatting problem: the telephone codes in my data source are formatted as (000)000-0000 display as 0000000000. This requires a simple numeric/text format switch:

{ MERCEFIELD Phone \\# "(000) 000-0000" }

Note the single quotes around the hyphen.

Mail Merge and Date Fields

When creating a merge document that will contain a DATE field that is to be merged to a document as opposed to a new document (or documents) as opposed to printed material, ensuring the date in the merged document which the merge took place can be problematical. DATE fields are preserved across a merge, thus when the document is opened at a later date, it will display the system date from the PC rather than the date the merge took place.

If you insert a CREATEDATE field then that will always display the date the document was created, and is not preserved during the merge, but is converted to text. While this is fine if you merge on the date the document with the field was created, you may wish to re-use the merge document, and display the new merge date. CREATEDATE cannot do that (unless you use SAVEAS to save the merge document in a different file).

There are several ways to work around this problem e.g.

{ DATE @ "d MMMM yyyy" }
{ IF TRUE { DATE @ "d MMMM yyyy" } }
{ QUOTE { DATE @ "d MMMM yyyy" } }
{ SET D { DATE @ "d MMMM yyyy" } { REF D } }

when merged these result as:
Testing for events that occur before or after a certain date

When running a mail merge, you may want to test for events that happen before or after a certain date. Mail merge does not convert dates to numbers, so if you cannot automatically derive a date number from the date in the data file, as you could in (say) Excel, a different plan is called for.

Let's say the date comes into Word in the format d/MM/yyyy or 1/10/2002 (1st October 2002) from a MERGEFIELD called Start_Date. In this example, we are looking to identify records with Start_Date before 1st October 2002.

{ IF{ MERGEFIELD Start_Date } < 1/10/2002 "True" "False"

The above would appear the logical check, but the check treats the date as a number and as 1, which is the first part of the number before the slash '/'. All dates other than the first of the month will be greater than 1, so all will produce the result "False".

We therefore need to display the date in numbers that represent the date in a unique way \yyyyMMdd which displays the date as a series of digits for year month and finally day, with a number that the conditional field will view as a whole.

{ IF{ MERGEFIELD Start_Date }@ "yyyyMMdd" } < 20021001 "True"

Thus any date before 1/10/2002 would produce "True" and any other date would produce "False".

Testing for fields that may contain numbers/date or text

While it would be better to ensure that the data file has separate fields for numbers and text to work with what you get. In a Word forum question a user wanted to test a field that contained dates and reproduce only those fields that contained dates. The solution relies on the fact that if you perform a calculation on a field containing text, an error message is produced e.g.:

!!Syntax Error, TEXT

This can be trapped in a conditional field structure by comparing the result of the field with thus:

{ MERGEFIELD Result } { IF{ =({ MERGEFIELD Result } "The field contains a date" "The field contains text" )

produces

13/03/2006 The field contains a date
if the field contains a date, or

Text  The field contains text

If the field contains text.

There is no need to test against the complete error message, as the condition will accept search for the leading exclamation mark and the wildcard character ‘*’ as shown.

**Date fields with ordinals**

UK dates are often displayed using a superscripted ordinal such as 23rd November 2002. To produce if the date is typed from the keyboard, but is difficult to produce automatically with fields (though there is a possible solution suggested in fellow MVP Paul Edstein's impressive study of date field download from this site in DateCalc.zip) as shown below:

```
{QUOTE{DATE[@ 'ddddd the d']} {IF{=(mod{DATE[@ d],10)<4)*([d],10)<>0)*{DATE[@ d]<><11)*{DATE[@ d]<>12)*{DATE[@ d] =mod{DATE[@ d],10}-2 (# 'rd nd th')}*} DATE[@ " of MMMM, yyyy
```

which produces:

Monday the 11th of April, 2005

Without such complexity, Word can readily manage 23rd November 2002 i.e. without the superscript is usually applied by Word's autoformat function, which does not affect insertions.

The required format can be achieved by converting the field to text and then applying the autoformat:

For a document template start with a combination of fields:

```
{CREATEDATE[@ "d" * Ordinal } {CREATEDATE[@ "MMM!
```

Select both fields and save to a bookmark - call it **Date**

The following macro will run automatically when a new document is opened, which will convert and autoformat to add the superscript.

```vba
Sub AutoNew()
Dim oRng As Range
Set oRng = ActiveDocument.Bookmarks("Date").Range
With oRng
.Fields.Unlink
.End = .Words.First.End - 1
.Start = .End - 2
.Font.Superscript = True
End With
End Sub
```
Sub InsertFormattedDate()
Dim oRng As Range
Dim oFld As Field
Set oRng = Selection.Range
NormalTemplate.AutoTextEntries("Date").Insert Where:=oRng
With oRng
.Fields.Unlink
.End = .Words.First.End - 1
.Start = .End - 2
.Font.Superscript = True
End With
End Sub

Changing minutes to hours and minutes

In a database used for mail merge, you may have a field called (e.g) MINUTES that contains e.g. 165 minutes, that you want to display in hours and minutes. Normal date/time switches will not help here, so you need to resort to mathematics to produce the required result.

\[
\text{hours} = \left\lfloor \frac{\text{minutes}}{60} \right\rfloor \\
\text{minutes} = \text{minutes} \mod 60
\]

This works as follows:

- \( \left\lfloor \frac{\text{minutes}}{60} \right\rfloor \) gives the number of whole hours.
- \( \text{minutes} \mod 60 \) gives the remaining minutes.

Thus the number of hours and minutes can be calculated as follows:

\[
\text{hours} = \left\lfloor \frac{165}{60} \right\rfloor = 2 \\
\text{minutes} = 165 \mod 60 = 45
\]

Either of the calculations will produce the same result:

In both examples the use of the formatting switches indicated below will remove any text following the number, thus if the field MINUTES actually contains 165 minutes, the content of the field is treated as 165.

The following information extracted and edited from Word's own help files completes the picture relating to the use of formatting switches.
## Format (`\`) fieldswitch

The following is a list of switches and items that they capitalize:

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\* Caps</code></td>
<td>The first letter of each word. For example, <code>{ FILLIN &quot;Type your name:&quot; \* Caps }</code> displays &quot;Graham Mayor&quot; even if name is typed in lowercase letters - graham mayor.</td>
</tr>
<tr>
<td><code>\* FirstCap</code></td>
<td>The first letter of the first word. For example, <code>{ COMMENTS \* FirstCap }</code> might display &quot;Weekly report on sales&quot;.</td>
</tr>
<tr>
<td><code>\* Upper</code></td>
<td>All letters are upper case. For example, <code>{ QUOTE &quot;word&quot; \* Upper }</code> displays &quot;WORD&quot;.</td>
</tr>
<tr>
<td><code>\* Lower</code></td>
<td>All letters are lowercase. For example, <code>{ FILENAME \* Lower }</code> displays &quot;weekly sales report.doc&quot;.</td>
</tr>
</tbody>
</table>

### Note

Where the text to be formatted is in uppercase, the `\* Caps` switch may do nothing. In this case first convert the content to lowercase by using the `\* Lower` switch thus:

```plaintext```
{ MERGEFIELD Name \*Lower \*Caps }
```

## Number formats

The following is a list of number switches and their results:

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\*alphabetic</code></td>
<td>Displays results as alphabetic characters. The result has the same case as the switch. For example, <code>{ SEQ appendix \* ALPHABETIC }</code> displays &quot;B&quot; (instead of &quot;2&quot;), and <code>{ SEQ appendix \* alphabetic }</code> displays &quot;b&quot;.</td>
</tr>
<tr>
<td><code>\*Arabic</code></td>
<td>Displays results as Arabic cardinal numerals. For example, <code>{ PAGE \* Arabic }</code> displays &quot;31&quot;.</td>
</tr>
<tr>
<td><code>\*CardText</code></td>
<td>Displays results as cardinal text. For example, <code>{ = SUM(A1:B2) \* CardText }</code> displays &quot;seven hundred ninety&quot;, and <code>{ = SUM(A1:B2) \* CardText \* Caps }</code> displays &quot;Seven Hundred Ninety&quot;.</td>
</tr>
</tbody>
</table>

### Note

If the Number format setting in the Page Number Format dialog box (Page Number command, Insert menu) is not Arabic, this switch overrides the default number format.
**\DollarText**
Displays results as cardinal text. Microsoft Word inserts "and" at the first two decimals (rounded) as Arabic numerators over 100. The result is formatted in lowercase letters unless you add a format switch to specify a different capitalization.

For example, \= 9.20 + 5.35 \DollarText \Upper \ displays "FOUR"

**\Hex**
Displays results as hexadecimal numbers. For example, \QUOTE "4

**\OrdText**
Displays results as ordinal text. The result is formatted in lowercase letters unless you add a switch to specify a different capitalization.

For example, \DATE \@ \"d\" \OrdText \ displays "twenty-first", and \FirstCap \ displays "Twenty-first".

**\Ordinal**
Displays results as ordinal Arabic numerals. For example, \DATE \@ \ORDINAL \ Note: See the section above relating to the insertion of superscripted ordinals.

**\Roman**
Displays results as Roman numerals. The result has the same case as the word "roman" in code. For example, \SEQ CHAPTER \Roman \ displays "xii", and \ roman \ displays "XI".

Character formats and protecting previously applied formats

The following are character formatting switches and their results:

**\CHARFORMAT**
Applies the formatting of the first letter of the field to the entire result. For example, \REF chapter2_title \CHARFORMAT \ displays "Whales of the Pacific".

You can of course use this to ensure that the formatting retains the formatting of the line that the field is inserted into. Thus

\REF chapter2_title \CHARFORMAT \ would display "Whales of the Pacific"

**\MERGEFORMAT**
Applies the formatting of the previous result to the new result. For example, displayed by the field \AUTHOR \MERGEFORMAT \ and apply bold formatting when the field is updated when the author name changes.

Note: For some odd reason known only to Microsoft, when you insert fields by using the Field dialog box (Insert menu, Field command), the \MERGEFORMAT switch is included by default. You can turn this option off on an individual basis by clearing the Preserve formatting during updates check box in the Field dialog box.

**Numeric Picture (\#) field switch**

Specifies the display of a numeric result.

For example, the switch \# \$,0.00 in \= SUM(ABOVE) \# \$,0.00 \ displays a result such as "$4,455.70." If the result of a field is not a number, this switch has no effect.

Quotation marks are not required around simple numeric pictures that do not include text, for example, \MarchSales \# \$,0.00 \. For more complex numeric pictures and those that include text...
spaces, enclose the numeric picture in quotation marks, as shown in the following Microsoft Word adds quotation marks to numeric picture switches if you insert a field command (Insert menu) or the Formula command (Table menu).

Combine the following picture items to build a numeric picture switch.

<table>
<thead>
<tr>
<th>Numeric Picture Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (zero)</td>
<td>Specifies the requisite numeric places to display in the result. If the that place, Word displays a 0 (zero). For example, { = 4 + 5 # 00.00 } displays “49”.</td>
</tr>
<tr>
<td>#</td>
<td>Specifies the requisite numeric places to display in the result. If that place, Word displays a space. For example, { = 9 + 6 # $### } displays “&quot;95&quot;.</td>
</tr>
<tr>
<td>x</td>
<td>Drops digits to the left of the &quot;x&quot; placeholder. If the placeholder is to Word rounds the result to that place. For example: { = 111053 + 111439 # x## } displays “492”. { = 1/8 # 0.00x } displays “0.125”. { = 3/4 # .x } displays “0.8”.</td>
</tr>
<tr>
<td>. (decimal point)</td>
<td>Determines the decimal point position. For example, { = SUM(ABOVE) # &quot;$,000.00&quot; } displays “$495.47”. Note: Use the decimal symbol specified as part of the regional Windows Control Panel.</td>
</tr>
<tr>
<td>, (digit grouping symbol)</td>
<td>Separates a series of three digits. For example, { = NetProfit # $#,###,### } displays “$2,456,800”. This can be written more simply as { = NetProfit # $,0 } Note: Use the digit grouping symbol specified as part of the regional Windows Control Panel.</td>
</tr>
<tr>
<td>- (minus sign)</td>
<td>Adds a minus sign to a negative result, or adds a space if the result example, { = 10 - 90 # -0 } displays “-80” and { = 10 + 90 # -0 } displays “190”.</td>
</tr>
<tr>
<td>+ (plus sign)</td>
<td>Adds a plus sign to a positive result, a minus sign to a negative result (zero). For example, { = 100 - 90 # +0 } displays “+10” and { = 90 - 100 # +0 } displays “-10”.</td>
</tr>
<tr>
<td>%, $, *, and so on</td>
<td>Includes the specified character in the result. For example, { = netprofit # &quot;##%&quot; } displays “33%”.</td>
</tr>
<tr>
<td>&quot;positive; negative&quot;</td>
<td>Specifies different number formats for positive and negative results. Sales95 is a positive value, the field { Sales95 # &quot;$0.00;-$0.00&quot; } displays the value with regular formatting &quot;$1,245.65&quot;. A negative value is displayed with bold formatting and a minus sign.</td>
</tr>
<tr>
<td>&quot;positive; negative; zero&quot;</td>
<td>Specifies different number formats for a positive result, a negative result, depending on the value of the Sales95 bookmark, { Sales95 # &quot;$0.00;-$0.00;0&quot; } displays positive, negative, and 0 (zero) values as follows: $1,245.65, ($345.56), $0.</td>
</tr>
</tbody>
</table>

Formatting fields with switches [http://www.gmayor.com/formatting_word_fields.htm](http://www.gmayor.com/formatting_word_fields.htm)
**Date-Time Picture (@) field switch**

Specifies the display of a date or time.

For example, the switch @ "dddd, MMMM d, yyyy" in the field \{ DATE @ "dddd, MMMM d, yyyy" \} displays "Friday, November 24, 2002." Combine the following date and time instructions—day (d), month (M), year (y); hours (h) and minutes (m)—to build a date-time picture. You can also include text, punctuation, and spaces.

### Date instructions

<table>
<thead>
<tr>
<th>Month (M)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Displays the month as a number without a leading 0 (zero). For example, February is &quot;2&quot;.</td>
</tr>
<tr>
<td>MM</td>
<td>Displays the month as a number with a leading 0 (zero). For example, February is &quot;02&quot;.</td>
</tr>
<tr>
<td>MMM</td>
<td>Displays the month as a three-letter abbreviation. For example, February is &quot;Feb&quot;.</td>
</tr>
<tr>
<td>MMMM</td>
<td>Displays the month as its full name. For example, February.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Day (d)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>Displays the day of the week or month as a number without leading 0 (zero). For example, the sixth day of the month is displayed as &quot;6&quot;.</td>
</tr>
<tr>
<td>dd</td>
<td>Displays the day of the week or month as a number with leading 0 (zero). For example, the sixth day of the month is displayed as &quot;06&quot;.</td>
</tr>
<tr>
<td>ddd</td>
<td>Displays the day of the week or month as a three-letter abbreviation. For example, Tuesday is displayed as &quot;Tue&quot;.</td>
</tr>
<tr>
<td>dddd</td>
<td>Displays the day of the week as its full name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Year (y)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yy</td>
<td>Displays the year as two digits with a leading 0 (zero). For example, 1999 is displayed as &quot;99&quot;, and 2006 is displayed as &quot;06&quot;.</td>
</tr>
<tr>
<td>yyyy</td>
<td>Displays the year as four digits.</td>
</tr>
</tbody>
</table>
## Time instructions

<table>
<thead>
<tr>
<th><strong>Hours (h)</strong></th>
<th>A lowercase &quot;h&quot; bases time on the 12-hour clock. An uppercase &quot;H&quot; bases time on the 24-hour clock; for example, 5 P.M. is displayed as &quot;17&quot;.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>h or H</strong></td>
<td>Displays the hour without a leading 0 (zero) for single-digit hours. For example, 9 A.M. is displayed as &quot;9&quot;.</td>
</tr>
<tr>
<td><strong>hh or HH</strong></td>
<td>Displays the hour with a leading 0 (zero) for single-digit hours. For example, 9 A.M. is displayed as &quot;09&quot;.</td>
</tr>
<tr>
<td><strong>Minutes (m)</strong></td>
<td>The letter &quot;m&quot; must be lowercase to distinguish minutes from months.</td>
</tr>
<tr>
<td><strong>m</strong></td>
<td>Displays minutes without a leading 0 (zero) for single-digit minutes. For example, TIME @ &quot;m&quot; displays &quot;2&quot;.</td>
</tr>
<tr>
<td><strong>mm</strong></td>
<td>Displays minutes with a leading 0 (zero) for single-digit minutes. For example, TIME @ &quot;mm&quot; displays &quot;02&quot;.</td>
</tr>
<tr>
<td><strong>A.M. and P.M. (AM/PM)</strong></td>
<td>Displays A.M. and P.M. To change the A.M. and P.M. symbols for Microsoft Windows, change the settings in Windows Control Panel.</td>
</tr>
<tr>
<td><strong>am/pm or AM/PM</strong></td>
<td>Displays A.M. and P.M. as uppercase. For example, { TIME @ &quot;h AM/PM} displays &quot;9 AM&quot; or &quot;5 PM&quot;.</td>
</tr>
</tbody>
</table>

## Other text and punctuation

| **'text'** | Any specified text in a date or time. Enclose the text in single quotation marks, \{ TIME \@ "HH:mm 'Greenwich mean time' " \} displays "11:45 Greenwich mean time". |
| **character** | Includes the specified character in a date or time, such as a colon (asterisk), or space. For example, \{ DATE \@ "HH:mm MMM-d, 'yy" \} displays "11:15 Nov 6, '99". |
| **`numbereditem`** | Includes the number of the preceding numbered item that you numbered using the Caption command on the Insert menu (Refer to a SEQ field). Enclose the item identifier, such as "Table". Microsoft Word displays the sequential number in arabic numerals. For example, \{ PRINTDATE \@ "Table `table` was printed on' M/dd\} displays "Table 2 was printed on 9/25/02". | **Note:** Quotation marks are not required around simple date-time pictures - for example, \{ DATE \@ "HH:mm MMM-d, 'yy" \} displays "11:15 Nov 6, '99". |
{ DATE \@ MM/yy }.

For more complex date-time pictures and those that enclose the entire date-time picture in quotation marks:

{ DATE \ "dddd MMMM d, yyyy, at' h:mm" }.

Microsoft Word adds quotation marks to date-time a field by using the Date and Time command or the menu.

<table>
<thead>
<tr>
<th>Lock Result (!) field switch</th>
</tr>
</thead>
</table>
| Prevents a field that is included in the result of a BOO!
REF field from being updated unless the field result in changed.

Without this switch, Microsoft Word updates fields incl
the BOOKMARK, INCLUDETEXT, or REF field is updated.

For example, the field

{ INCLUDETEXT C:\Sales\Qtr4 Sales.doc ! }

inserts the contents of the document "Qtr4 Sales.doc," and an EMBED field.

If you update the INCLUDETEXT field, the "!" switch 1
DATE and EMBED fields in the included text unless th
original document ("Qtr4 Sales.doc").

The switch ensures that the text inserted by the INCL
in the original document. To update the DATE and EM
update fields in the original document ("Qtr4 Sales.doc
INCLUDETEXT field.