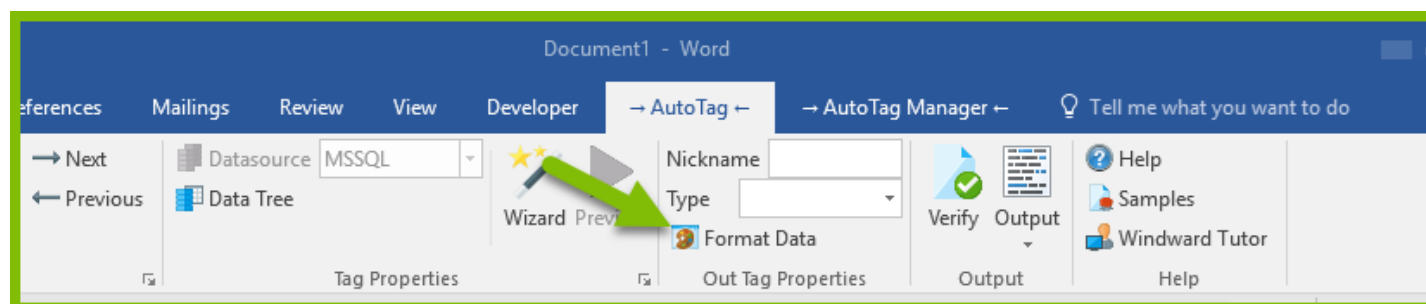


# How Do I Format Dates and Numbers With an Out Tag?

This article will show you how to format data displayed by an Out Tag using the Tag's **format** property. The **format** property allows you to apply special formats to the data such as currency, dates, times, decimal and thousands separators, etc. The formats can be locale specific. There should be enough formatting options to satisfy most needs, but if not, you can also create a custom format.

The Out Tag **format** property was designed to be very similar to Microsoft Excel cell formatting. If you built your template in Microsoft Excel then you should continue to use Excel's native cell formatting. But if you built your template in Microsoft Word or PowerPoint, use the **format** property to achieve the same degree of control you have when using Excel's native cell formatting.

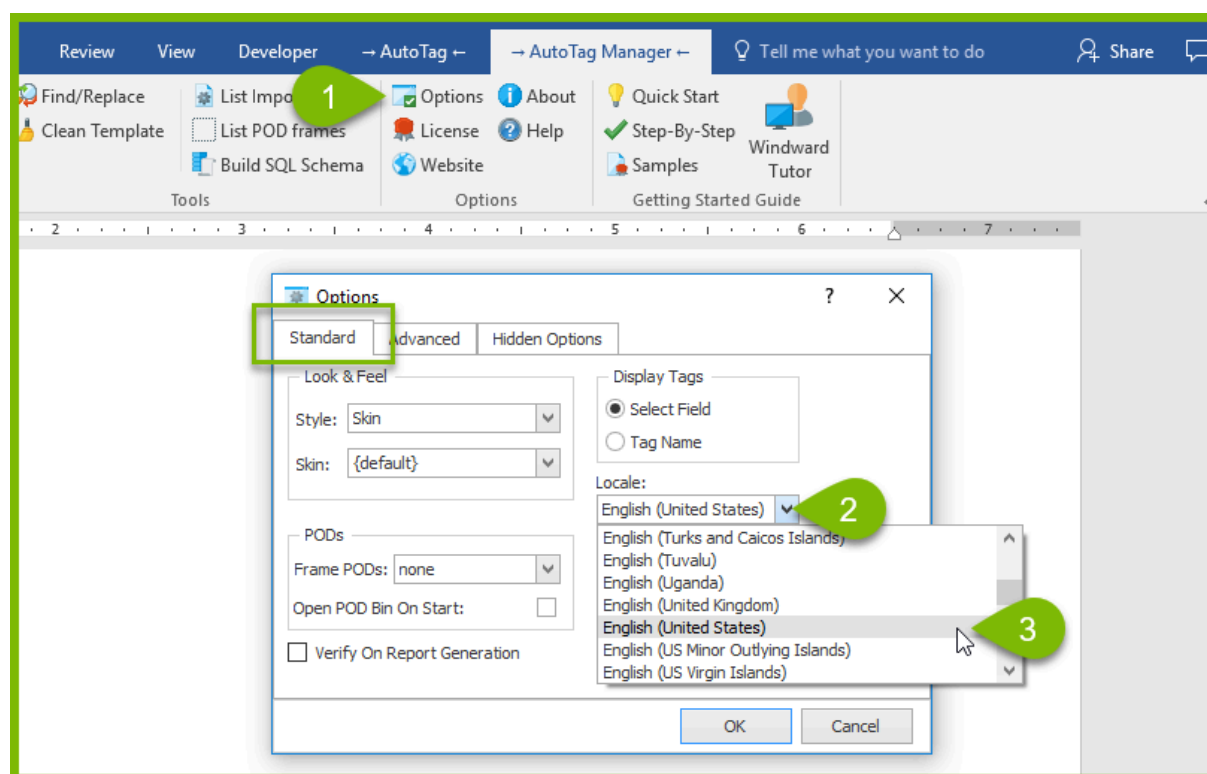
💡 When you select an Out Tag in Word or PowerPoint, the *Format Data* button appears in the AutoTag tab of the ribbon. You can click on this button to access the Out Tag **format** property. It has the same effect as opening the [Tag Editor](#) on an Out Tag and clicking on the **format** property. Selecting an Out Tag is the only time the Format Data button appears.



The Out Tag **input** property is used to tell the Designer about the format of the data stored in your data source, in case the Designer doesn't recognize it. The **input** property is often used with the **format** property to give you complete control over your data's format when input to and output from an Out Tag. We'll provide more details about advanced formatting using the **input** property below in [Advanced Formatting Using the Input Property](#).

## European-formatted Numbers

To use European-formatted numbers it is easiest to change the Designer locale settings in the Designer Options Interface. In the Standard tab, use the "Locale:" drop-down menu to set the locale for this particular template:



**i** The Locale setting in the Designer Options is in effect until you change it; including new templates.

In the Java Reporting Engine, this option is set in the [WindwardReports.properties](#) file; in the .NET Reporting Engine it is set in the [app.exe.config](#) file; and in the Report Engine for RESTful it is set in the **[web.config file]**.

## Using the Format Data Interface

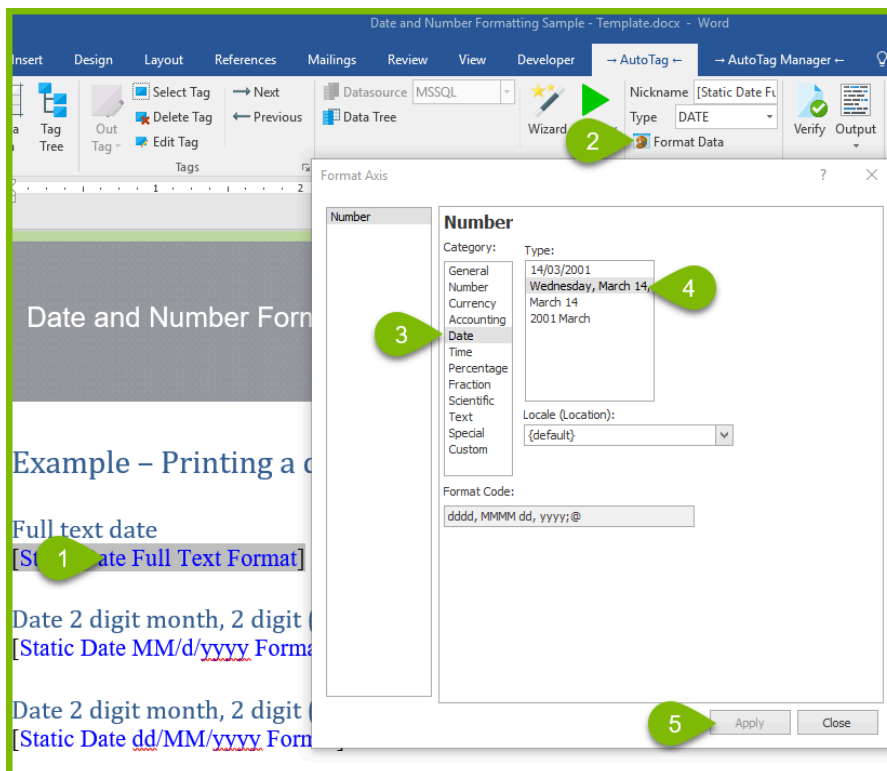
We'll use this example template, which uses our public mssql.windward.net data source, to demonstrate using the Format Data interface.



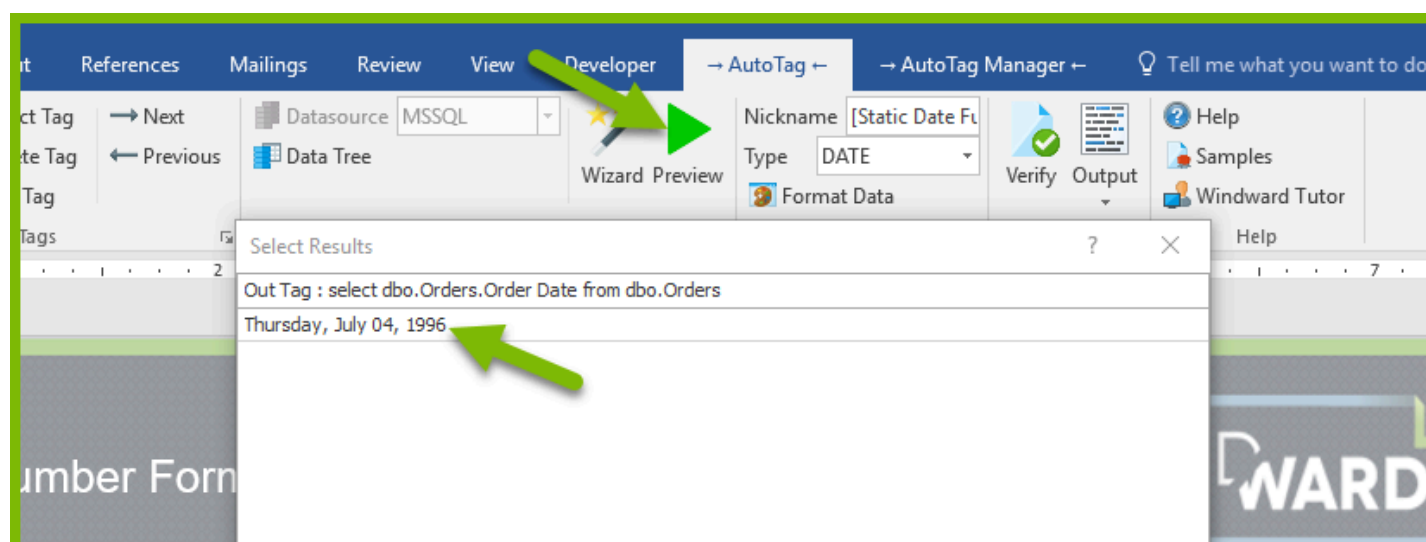
[Date\\_and\\_Number\\_Formatting\\_Sample\\_-\\_Template.docx](#)

## Formatting Dates

1. Open the sample template attached above and select the first date example Out Tag **[Static Date Full Text Format]**
2. In the AutoTag tab on the ribbon select the "Format Data" button to open the Format Data interface.
3. In the "Category:" list select "Date."
4. In the "Type:" section select your desired date format (this is how the data will appear in your output).
5. Click on "Apply", then "Close".



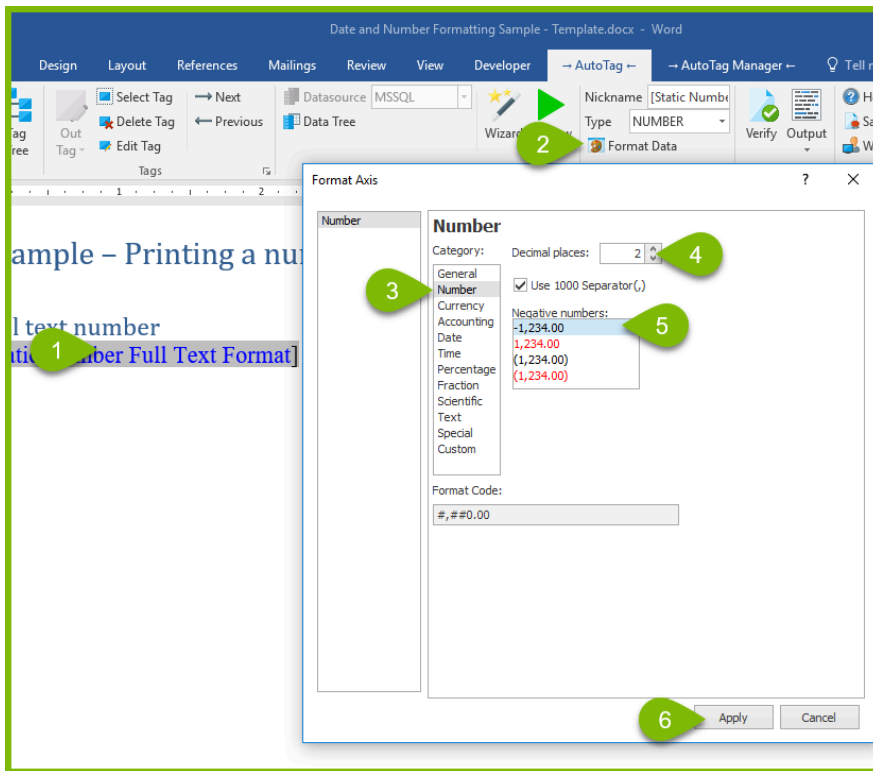
Click on the "Preview" button to preview your newly formatted date.



**i** Using the "Format Data" button will automatically change the data type to DATE by analyzing the result and automatically applying it.

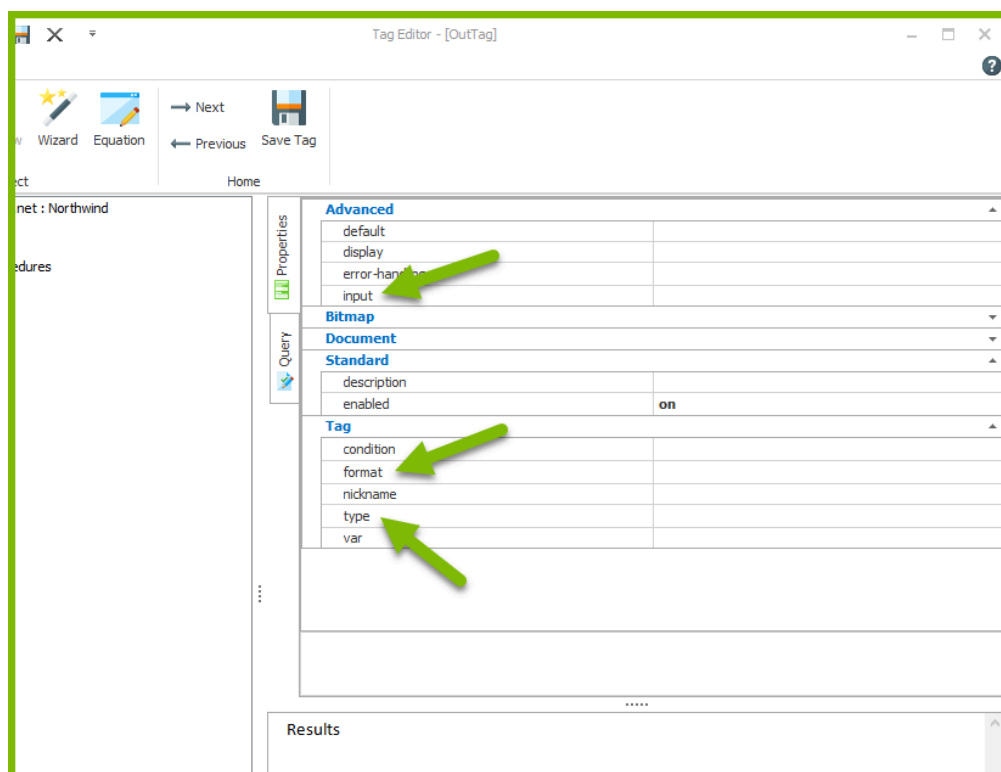
## Formatting Numbers

1. Open the sample template from above and select the first number example Out Tag [**Static Number Full Text Format**].
2. In the AutoTag tab on the ribbon select the "Format Data" button to open the Format Data interface.
3. In the "Category:" list select "Number."
4. Set the desired "Decimal places:."
5. In the "Negative numbers:" section select your desired number format (this is how the data will appear in your output).
6. Click on "Apply", then "Close."



## Advanced Formatting Using the Out Tag Format and Input Properties

Up to this point we've used the Format Data interface to set the **format** property of Out Tags to control the display of dates and numbers. While this is almost always adequate, there may be cases where the format of the data in your data source isn't recognized by the Designer, or your desired output format isn't included in the Format Data interface selections. Then we must use the Out Tag **type** property, the "Custom" category of the **format** property, and the **input** property, for complete control over date and number formatting.



## Advanced Date Formatting

If the dates in your data source aren't formatted using standard patterns such as "MM/dd/yyyy", "yyyy/mm/dd", or the [ISO 8601](#) "MM/dd/yyyy'T'hh:mm:ss" formats, then you must specify how your dates are formatted using the **input** property.

For example, suppose your date is formatted in your data source as "D2018-05-11 T14:16:00 MST", which is intended to mean 2:16 PM on May 5, 2018, Mountain Standard Time. The Designer would not recognize this format as a date, so you must set the Out Tag **type** property to *DATE*, and set the **input** property to the *format pattern* 'D'YYYY-MM-DD 'T'HH:MM:SS 'MST'.

We'll describe this process, using this example, in detail below.

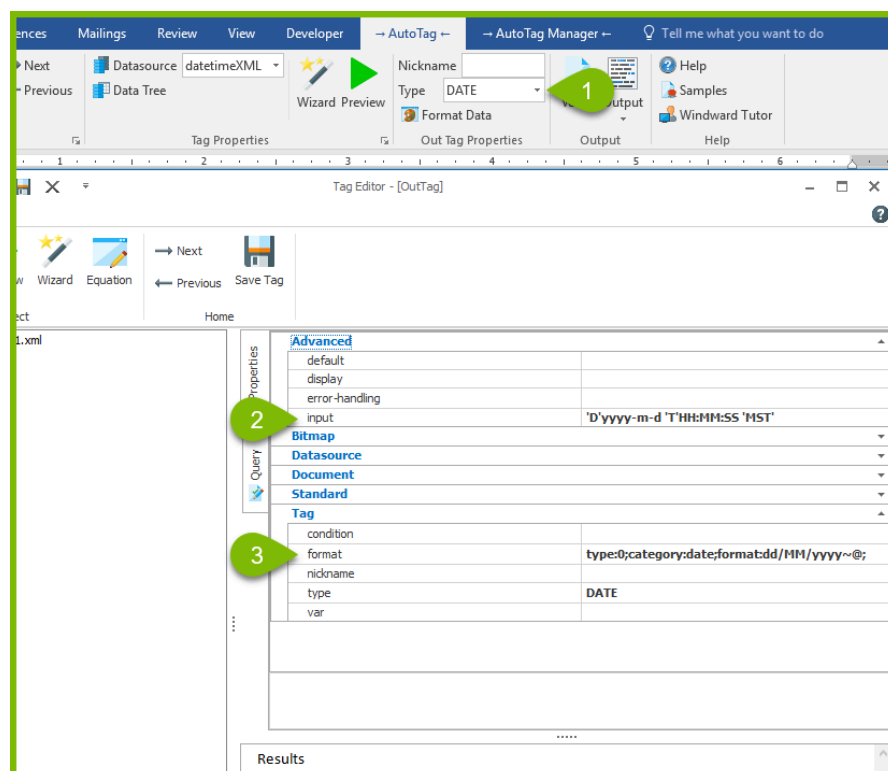
**i** These Java libraries specify the symbols you can use to create format patterns for the **input** property:

- [SimpleDateFormat](#) and [DateFormat](#) are used for dates
- [DecimalFormat](#) and [NumberFormat](#) are used for numbers

To create a custom date format for your Out Tag output when the Designer doesn't recognize the date format in your data source:

1. Set the **type** property to *DATE*.

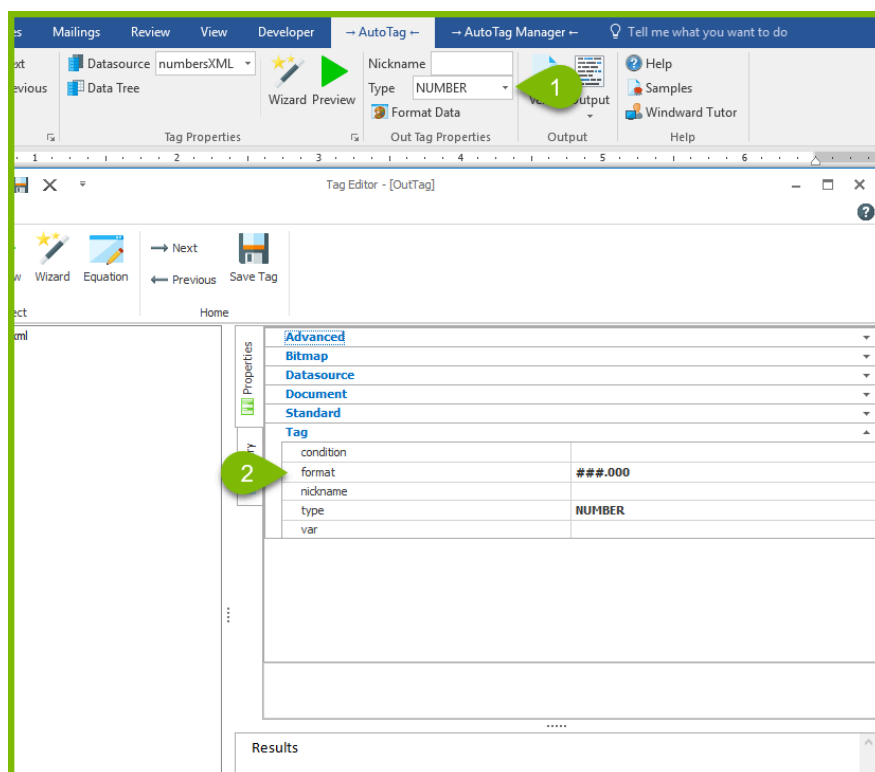
- Using the Java library documentation referenced above, create a format pattern that matches the format of your dates in your data source, then enter it into the **input** property (for example 'D'YYYY-MM-DD 'T'HH:MM:SS 'MST').
- Using the Java library documentation referenced above, create a format pattern that matches the desired format of dates in your output, and enter it into the format property (for example 'dd/MM/yyyy').



## Advanced Number Formatting

Numbers can be formatted using the [DecimalFormat](#) Java Library symbols, which is a string of mostly "#" and "0" characters.

- Set the **type** attribute to *NUMBER* to apply a format pattern to the number returned from your data source.
- Enter a format pattern using the [DecimalFormat](#) Java Library symbols. Entering a pattern of "000.000" will output leading and trailing zeros if there are fewer than three digits before or after the decimal point. This would output the number "32.38" as "032.380". However, this is unlikely to be the desired result. To remove the leading and trailing zeros, instead enter the the format pattern "###.000". The character "#" will display a digit if present, but will *not* display a "0" if a digit isn't present. This pattern will output the number "32.38" as "32.380".



- i** Internationalization (I18N) note: The number format always uses a comma for the thousands separator and a period for the decimal separator. These separators are replaced with the characters for the default locale-specific settings of the computer generating the output. This allows a single format setting to be used by all locales.