Advanced Form Template Modification

Customers interested in creating highly customized form templates can leverage the NF Merge field language to create or modify existing form templates.

Inserting New Fields Manually

To insert a new field, use the Insert Field Function.

1. Click the  icon on the Quick Access Toolbar (i.e. the **Insert Field** function)
2. On the Field dialog, change **Field Names** to **MergeField**.

3. Enter in the Name of the field to be inserted (i.e. NFField:Transmittal:subject)
4. Click OK.

Sections & Optional Sections

A **section** defines a set of data that repeats for multiple records (i.e. list of recipients, RFI list, etc.) or data that may not always be displayed. Sections can be data loops and/or program logic driven. A section is defined by a Merge field tagging the start and a Merge field tagging the end of the section using one of the two sets of fields:

|  |  |
| --- | --- |
| NFSectionStart:[Data Node]NFSectionEnd:[Data Node] | Outputs data for a specific section. The layout elements within the section are always shown, even if there is no data to parse out. |
| NFOptionalStart:[Data Node] NFOptionalEnd:[Data Node] | Outputs data only if Data Node or is present and not empty. The elements within the section are only shown when there is data to parse out. |

When there are multiple records in a section, the formatting and static elements between the two tags will be repeated until there are no more records. When placed inside of a Word table, a table row will be added for each record.



In this example, a list of recipients is generated with the **Name**, **Company**, **Email Address** and **Work Phone Number** for each recipient. A new table row is automatically generated for each recipient.

* The **NFSectionStart** and **NFSectionEnd** must reside ***INSIDE*** the first and last cell within the table row respectively to correctly add rows to the Word table for subsequent records.

The above layout will result in a form looks like this when run:



* You must close any **NFSectionStart** or **NFOptionalStart** with the respective **NFSectionEnd** or **NFOptionalEnd** field or the form template will not work properly.
* To avoid a blank line and/or extra space being inserted before each record, be sure the first field of a section is on the same line immediately following the NFSectionStart field and do not place a space inbetween.

Enclosed Fields

When specifying a field inside an **NFSection**, it is not necessary (or often possible) to use an **NFField** field. The data path contained within the **NFSection** is assumed for all fields within the **NFSection**.

In our previous example:



The **name, company, email** and **phoneWork** fields do not require **NFField** as they are enclosed by Section start and end tags.

In most cases, as in the above example, tables will be used to organize the layout of data within sections. The tables themselves do not necessarily need to be visible, but can serve as a powerful tool for laying out how data will flow in a template.

Stand-alone Fields

You can place single fields that come from different data nodes than the current data node or outside of section start and end tags using the **NFField** merge field function.



This field displays the **subject** of a **Transmittal** form. Colon separated notation is used within the field name to specify the nested path to the field being displayed. The name and path of the field is enclosed by guillemets (they look like two greater than or lesser than signs).

When NFField and a corresponding data path is not used in the field name, the field is rendered from the data set of the Section or Optional Section Start and End tags it is enclosed in (covered in the next section, *Sections & Optional Sections).*

* It is not possible to place NFFields from the same data path on the same line, paragraph or table row. Project Center will throw an error. When two or more fields are required on the same line, paragraph or table row – use an appropriate section Start and End tag and Enclosed Field tags.

Using Word Tables

In order for tables to render correctly, it’s important that the table properties be set to disable automatic resizing. This should be set for every table in your form template.

Select the table.

Right click and select **Table Properties…**

On the **Table** tab, click the **Options** button.

Uncheck **Automatically Resize to Fit Contents**.

Formatting Fields

To format the appearance of a field:

Highlight the entire field including the enclosing guillemets.

Apply Font, Styles or any other formatting to the field.

Newforma strongly recommends the use of Styles within Word Form Templates to speed up the creation and editing of forms. For more information on formatting Word Documents, including how to use Styles, please see Microsoft Word’s Help system.

Using Word’s ‘IF’ Construct

Word has an IF field that you can use to display content conditionally. In the example below, if the "street" attribute of the sender node is not empty, a comma is output between the street and city, otherwise an empty string (nothing) is output. For more information, see [Microsoft's Examples of IF fields](http://office.microsoft.com/en-us/word-help/examples-of-if-fields-HP005187676.aspx).



* Word will only display the first 45 characters of a merge field name. To view the entire field, you must use the **Edit Merge Field** dialog box.

Formatting Date and Time Fields

The various Date and Time fields that are made available to Word Form templates can be reformatted in the template. Date formatting only works with fully qualified, enclosed merge field.

For simple formatting, with no spaces or text in the result, the formatting Picture doesn't need quotes. For example:

 { DATE \@ MM/yy }

In the above example, DATE is the field name.

Use **ALT-F9** to Toggle Field view and enable editing of Merge Field formatting.

**Notes & Caveats**

Date formatting is not possible in the ContentDescription and Action nodes. Other nodes support a fully qualified merge field within the Section or Optional delimiters, but ContentDescription and Action do not.

Only one date in an IF construct can be formatted.

Note that month is denoted with a capital M, to distinguish it from minutes (which uses a lower case m).

|  |  |  |
| --- | --- | --- |
| **Day** | **Month**  | **Year** |
|

|  |  |
| --- | --- |
| d | Number without leading zero |
| dd | Number with leading zero |
| ddd | Three letter abbreviation |
| dddd | Full name |

 |

|  |  |
| --- | --- |
| M | Number without leading zero |
| MM | Number with leading zero |
| MMM | Three letter abbreviation |
| MMMM | Full name |

 |

|  |  |
| --- | --- |
| yy | Two digit year with leading zero |
| yyyy | Full four digit year |

 |
| Hours | Minutes | AM & PM |
|

|  |  |
| --- | --- |
| h | 12 hour clock, without leading zero |
| hh | 12 hour clock, with leading zero |
| H | 24 hour clock, without leading zero |
| HH | 24 hour clock, with leading zero |

 |

|  |  |
| --- | --- |
| m | Without leading zero |
| mm | With leading zero |

 |

|  |  |
| --- | --- |
| am/pm | Displays AM or PM |
| AM/PM | Same as above |

 |

* Text within the Date/Time picture must be enclosed in single quotes.

Examples

{ TIME \@ "HH:mm 'Greenwich mean time' " } displays 12:45 Greenwich mean time

{ DATE \@ "dddd, MMMM d" } displays Saturday, November 26

{ DATE \@ "h:mm am/pm, dddd, MMMM d" } displays 10:00 AM, Saturday, November 26

{ DATE \@ \@"MMMM d, yyyy" } displays November 26, 2013

{ DATE \@ "dddd, MMMM d, yyyy" } displays Friday, November 26, 2013