

Overview

Templates within Engage: Events are useful for redefining the layout of the module without developer interaction. Anyone with knowledge of CSS and the contents of this document can restyle the Events module in many ways.

Listing Display

Select *List* as the display in the module settings to access the templated listing view. On the Choose Display page you can pick which template to use for the main listing display and also for the detailed view of a single event. The easiest way to create a new template is to find an existing template and modify it to meet your needs. Templates are stored in the *Templates* folder in the module's folder under your website's main *DesktopModules* folder (i.e. */DesktopModules/EngageEvents/Templates*). Each folder under the *Templates* folder can contain a template.

Template Manifest

Template details are defined in an XML manifest file, named *Manifest.xml*. The format of that file is defined in *EngageManifest.xsd*.

In the manifest you can define a template's name*, description*, template file*, thumbnail file, stylesheet file, resource file, and settings (items with a * are required).

The name, description, and thumbnail will be used only when selecting a template. These are intended to give users of the template a good idea of its intentions and requirements.

The stylesheet file is a CSS stylesheet that will be referenced on the page whenever the template is being displayed. It is strongly suggested that the styles in this stylesheet be restricted so that they only affect the template's contents (that is, instead of styling *.Normal* or *ul*, style *.my-template .Normal* and *.my-template ul*, where *.my-template* is a class which surrounds the template's contents).

The resource file is a *.resx* file where localized resources can be stored. These files can be edited through the DotNetNuke language editor in order to provide custom text per language and/or per portal. You can access resources in your template using the *%\$* or *-ResourceKey* syntax, described below.

A template is a file consisting of HTML and custom tags to allow for complete control over the display of the module.

Settings

The settings section defines a list of name/value pairs, where the name is a setting name in the module, and the value is the value of that setting. The specific settings and their possible values differ for each module. Please note that the setting names are case-sensitive.

The available settings for Engage: Events are as follows:

Setting Name	Setting Purpose	Possible Values
<i>DisplayModeOption</i>	Which set of events to show	All, Past, CurrentMonth, Future
<i>FeaturedOnly</i>	Whether to only show events marked as featured	True, False
<i>RecordsPerPage</i>	The number of events to display in a page of events	(Whole number)

Installing Templates

If you feel the need to alter the structure of the module's output, you can create a template of your own design to accomplish that task. You may also find a template created by someone else that you would like to use.

In order to use any templates that do not come with the module, you will need to install them onto your site. Templates are stored in the *Templates* folder in the module's folder under your website's main *DesktopModules* folder (i.e. */DesktopModules/EngageEvents/Templates*). Any new templates that you add into this folder will be immediately available from the module's Choose Display page.

Therefore, if you have a template that you would like to use on your site, you will need to transfer it (as a folder with a valid manifest, template file, and any other related files) to your site, placing it in the folder */DesktopModules/EngageEvents/Templates*. Once there, it will immediately be available from the module's Choose Display page, so long as the manifest is valid.

Engage Entities and Attributes

Engage has developed an HTML-like tag system for designers to include Engage: Events-specific entities within their templates. All Engage tags will start with *Engage:* to differentiate them from HTML. Any tags starting with *Engage:* that are not defined below will not be displayed on the page.

The templating system provides a special syntax to allow access to the properties of specific events, and to text resources in a .resx resource file associated with the template. To access properties, you can surround the property name with *%#* and *%*, or append *-PropertyName* to the attribute name, i.e. *Text="%#Title%"* or *Text-PropertyName="Title"* will set the *Text* value of a tag to the *Title* of an event. Note that event properties can only be accessed from within an *Engage:List* tag.

You can also provide a format, separating it from the property name by a colon (:) in the terse syntax, or by providing an attribute with *-Format* appended to the attribute name, e.g. *Text="%#EventStart:hh:mm%"* or *Text-PropertyName="EventStart" Text-Format="hh:mm"* will format the start date to display hours and minutes. The default format for date/time values is *G*. The default format for numeric values depends on the exact type of value (integer, decimal, etc.). The default format for text values depends on the contents of the value (i.e. the Description is displayed as entered, since it is an HTML value, and the other values are displayed HTML encoded). For more information on formatting, see the appendices at the end of this document.

Engage: Events allows access to the following properties for events:

- *Id* – the internal ID used to reference the event
- *Title* – the title of the event
- *Overview* – the overview text for the event
- *Description* – the long description of the event
- *Location* – the event's location
- *Event Start* – the date and time on which the event starts
- *Event End* – the date and time on which the event ends
- *Category-Name* – the name of the event's category
- *Category-Color* – the color of the event's category

To access text resources, you can surround the resource key with %\$ and %, or append *-ResourceKey* to the attribute name, i.e. *Text="%\$Header.Text%"* or *Text-ResourceKey="Header.Text"* will set the *Text* value of a tag to the value of the resource with the key *Header.Text* in the template's resource file. If the key provided does not contain a period (.), then *.Text* will be appended to the key.

Tag Definitions

Tag: Engage:Literal

Purpose: Displays literal text.

Attributes/Values:

- Text
 - The text to display

Example: `<Engage:Literal Text="%#Title%"/>`

Tag: Engage:Label

Purpose: Displays literal text within a *span* HTML element.

Attributes/Values:

- Text
 - Text to display
- CssClass
 - The CSS class to apply to the element.
- ToolTip
 - The text displayed when the user's mouse hovers over the label

Example: `<Engage:Label Text="%#Title%" CssClass="event-title"/>`

Tag: Engage:Image

Purpose: Displays an image.

Attributes/Values:

- ImageUrl
 - The URL to the image. If this is not provided, then the image is not rendered on the page.
- AlternateText
 - Text to be displayed when the image cannot be loaded. Also displays as a tooltip in Internet Explorer if a Tooltip is not specified.
- CssClass
 - The CSS class to apply to the image.
- DescriptionUrl
 - Sets the description URL attribute of the image, specifying a URL whose contents is a long description of the image's contents
- Tooltip
 - The text to display when the user's mouse hovers over the image.

Example: `<Engage:Image ImageUrl="%#ImageUrl%" AlternateText="%#Title%"/>`

Tag: Engage:Link

Purpose: Displays a link.

Attribute/Value:

- NavigateUrl
 - The URL to which the user is directed after clicking this link
- Text
 - Text to display. If this tag has nested (i.e. children) tags, those are displayed instead of this property.
- CssClass
 - The CSS class to apply to the hyperlink.
- Target
 - The target window or frame in which to display the linked content

Examples: `<Engage:Link Text="%$View Details%" NavigateUrl="%#LinkUrl%" />`

`<Engage:Link><Engage:Literal Text="%#Title%"/></Engage:Link>`

Tag: Engage:List

Purpose: Repeats its contents for each slide in the module.

Example: `<Engage:List><Engage:Literal Text="%#Title%"/></Engage:List>`

Tag: Engage:ReadMore

Purpose: Displays a link to the Single Item template of its event, if it has one. If it is used outside of an Engage:List tag (i.e., not in reference to a specific event), it displays a link to the detail page for the module.

Attribute/Value:

- Text
 - The text to display.
- CssClass
 - The CSS class to apply to the element.

Examples: `<Engage:ReadMore Text="%$ReadMore%"/>`

`<Engage:ReadMore></Engage:ReadMore>`

Tag: Engage:EventSort

Purpose: Displays a radio button list to allow the user to choose whether the list should be sorted by Title or Start Date.

Example: `<Engage:EventSort/>`

Tag: Engage:StatusFilter

Purpose: Displays a radio button list to allow the user to choose whether the list should display cancelled events or not.

Example: `<Engage:StatusFilter/>`

Tag: Engage:CategoryFilter

Purpose: Displays a drop down list to allow the user to choose whether the list should be filtered to a single category.

Example: `<Engage:CategoryFilter/>`

Tag: Engage:RecurrenceSummary

Purpose: Displays summary text for this event's recurrence rule, for example "Occurs the first day of January" or "Occurs the last weekend day of December."

Example: `<Engage:RecurrenceRule/>`

Tag: Engage:EventWrapper

Purpose: Renders a *div* HTML element with specific CSS classes depending on the state of the event.

Attribute/Value:

- **RecurringEventCssClass**
 - The CSS class to apply to this element if the event is recurring.
- **FeaturedEventCssClass**
 - The CSS class to apply to this element if the event is featured.
- **AlternatingCssClass**
 - The CSS class to apply to this element if the event is even numbered in its current list (i.e. this CSS class is applied to every other event in the list).
- **CssClass**
 - The CSS class to always apply to this element.

Example: `<Engage:EventWrapper CssClass="EventItem" RecurringEventCssClass="RecurringEventItem" FeaturedEventCssClass="FeaturedEventItem" AlternatingCssClass="AlternatingEventItem"> ... </Engage:EventWrapper>`

Tag: Engage:Duration

Purpose: Displays the range of time that an event takes place in a “smart” format. For instance, if the event takes place for a couple of hours during just one day, it will (by default) say something like *Wednesday, February 4, 2009, 01:00PM - 02:30PM*, but if it takes place over multiple days, it will include both days in the text (*Sunday, December 27, 2009, 07:00PM - Monday, December 28, 04:00AM*), etc. These formats can also be changed by editing the four “Timespan...” resource keys in the module’s SharedResources.resx resource file.

Example: `<Engage:Duration/>`

Tag: Engage>EditEventButton

Purpose: Displays a button that directs to the edit page for its event when clicked. This is only displayed for users with edit rights to the module.

Example: `<Engage>EditEventButton/>`

Tag: Engage:ViewResponsesButton

Purpose: Displays a button that directs you to the responses page for its Event. This is only displayed for users with edit rights to the module.

Example: `<Engage:ViewResponsesButton/>`

Tag: Engage:RegisterButton

Purpose: Displays a button that directs you to the registration page for its Event. If the user is not logged in, they are directed a page telling them to login or register. Only registered users can register for an event. This button is not displayed if the event has been cancelled, has already ended, or is not set to allow registrations.

Example: `<Engage:RegisterButton/>`

Tag: Engage:AddToCalendarButton

Purpose: Displays a button that causes the user to download an iCalendar file to import into their calendar. It is not displayed if the event has been cancelled or has already ended.

Example: `<Engage:AddToCalendarButton/>`

Tag: Engage>DeleteButton

Purpose: Displays a button that deletes its event when clicked. This is only displayed for users with edit rights to the module.

Example: `<Engage>DeleteButton/>`

Tag: Engage:CancelButton

Purpose: Displays a button that cancels its event when clicked. This is only displayed for users with edit rights to the module.

Example: `<Engage:CancelButton/>`

Tag: Engage:PreviousPage

Purpose: Displays a link to the previous page in this list, if there is a previous page.

Attribute/Value:

- Text
 - Text to display
- CssClass
 - The CSS class to apply to the element.
- ToolTip
 - The text displayed when the user's mouse hovers over the link

Example: `<Engage:PreviousPage Text="%$Prev%" ToolTip="%$Prev.ToolTip%" CssClass="CommandButton" />`

Tag: Engage:NextPage

Purpose: Displays a link to the next page in this list, if there is a next page.

Attribute/Value:

- Text
 - Text to display
- CssClass
 - The CSS class to apply to the element.
- ToolTip
 - The text displayed when the user's mouse hovers over the link

Example: `<Engage:NextPage Text="%$Next%" ToolTip="%$Next.ToolTip%" CssClass="CommandButton" />`

Tag: Engage:CurrentPage

Purpose: Displays the current page number of the current list, if there are any items to display.

Attribute/Value:

- CssClass
 - The CSS class to apply to the span element created by this tag.

Example: `<Engage:CurrentPage CssClass="Normal" />`

Tag: Engage:PageCount

Purpose: Displays the total number of pages in the current list, if there are any items to display.

Attribute/Value:

- **CssClass**
 - The CSS class to apply to the span element created by this tag.

Example: `<Engage:PageCount CssClass="Normal" />`

Tag: Engage:PageXOfY

Purpose: Displays the current page and total number of pages in a formatted sequence, if there are any items to display.

Attribute/Value:

- **Text**
 - The text to display. In the text, `{0}` will be replaced by the current page number, and `{1}` will be replaced by the total number of pages.
- **CssClass**
 - The CSS class to apply to the span element created by this tag.

Example: `<Engage:PageXOfY Text="{0} of {1}" CssClass="Normal" />`

Appendix A: Text Format Values

Text values are formatted to appear in HTML, by default. If you need a different format, you can use one of the below format value.

Format Specifier	Description	Example
Raw	Plain Text – Default for HTML properties such as <i>Description</i>	Ben & Jerry's Ice Cream
CSS	CSS/Slug attribute value – replaces spaces and other non-alphanumeric characters with dashes (-) and removes diacritics from letters	Ben-Jerry-s-Ice-Cream
HTML	HTML encoded – Default for plain text properties, such as <i>Title</i>	Ben & Jerry's Ice Cream

Appendix B: Date/Time Format Values

When formatting date/time values, there are a number of built-in format options from which to choose. If you require more flexibility, you can also create a custom date/time Format.

Standard Date/Time Formats¹

Format Specifier	Name	American English Example
d	Short date pattern	7/31/2008
D	Long date pattern	Thursday, July 31, 2008
f	Full date/time pattern (short time)	Thursday, July 31, 2008 12:00 AM
F	Full date/time pattern (long time)	Thursday, July 31, 2008 12:00:00 AM
g	General date/time pattern (short time)	7/31/2008 12:00 AM
G	General date/time pattern (long time)	7/31/2008 12:00:00 AM
M or m	Month day pattern	July 31
o	Round-trip date/time pattern	2008-07-31T00:00:00.0000000
R or r	RFC1123 pattern	Thu, 31 Jul 2008 00:00:00 GMT
s	Sortable date/time pattern; conforms to ISO 8601	2008-07-31T00:00:00
t	Short time pattern	12:00 AM
T	Long time pattern	12:00:00 AM
u	Universal sortable date/time pattern	2008-07-31 00:00:00Z
U	Universal sortable date/time pattern	Thursday, July 31, 2008 5:00:00 AM
Y or y	Year month pattern	July, 2008
Any other single character	Unknown format value (uses the General date/time pattern (long time))	7/31/2008 12:00:00 AM

¹ From <http://msdn.microsoft.com/en-us/library/az4se3k1.aspx>

Custom Date/Time Formats²

Format Specifier	Description
d	Represents the day of the month as a number from 1 through 31. A single-digit day is formatted without a leading zero.
dd	Represents the day of the month as a number from 01 through 31. A single-digit day is formatted with a leading zero.
ddd	Represents the abbreviated name of the day of the week.
dddd	Represents the full name of the day of the week.
f	Represents the most significant digit of the seconds fraction.
ff	Represents the two most significant digits of the seconds fraction.
fff	Represents the three most significant digits of the seconds fraction.
ffff	Represents the four most significant digits of the seconds fraction.
fffff	Represents the five most significant digits of the seconds fraction.
ffffff	Represents the six most significant digits of the seconds fraction.
fffffff	Represents the seven most significant digits of the seconds fraction.
F	Represents the most significant digit of the seconds fraction. Nothing is displayed if the digit is zero.
FF	Represents the two most significant digits of the seconds fraction. However, trailing zeros, or two zero digits, are not displayed.
FFF	Represents the three most significant digits of the seconds fraction. However, trailing zeros, or three zero digits, are not displayed.
FFFF	Represents the four most significant digits of the seconds fraction. However, trailing zeros, or four zero digits, are not displayed.
FFFFF	Represents the five most significant digits of the seconds fraction. However, trailing zeros, or five zero digits, are not displayed.
FFFFFF	Represents the six most significant digits of the seconds fraction. However, trailing zeros, or six zero digits, are not displayed.
FFFFFFF	Represents the seven most significant digits of the seconds fraction. However, trailing zeros, or seven zero digits, are not displayed.
g or gg	Represents the period or era (A.D. for example).
h	Represents the hour as a number from 1 through 12, that is, the hour as represented by a 12-hour clock that counts the whole hours since midnight or noon. Consequently, a particular hour after midnight is indistinguishable from the same hour after noon. The hour is not rounded, and a single-digit hour is formatted without a leading zero. For example, given a time of 5:43, this format specifier displays "5".
hh	Represents the hour as a number from 01 through 12, that is, the hour as represented by a 12-hour clock that counts the whole hours since midnight or noon. Consequently, a particular hour after midnight is indistinguishable from the same hour after noon. The hour is not rounded, and a single-digit hour is formatted with a leading zero. For example, given a time of 5:43, this format specifier displays "05".
H	Represents the hour as a number from 0 through 23, that is, the hour as represented by a zero-based 24-hour clock that counts the hours since midnight. A single-digit hour is formatted without a leading zero.

² From <http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx>

HH	Represents the hour as a number from 00 through 23, that is, the hour as represented by a zero-based 24-hour clock that counts the hours since midnight. A single-digit hour is formatted with a leading zero.
M	Represents the minute as a number from 0 through 59. The minute represents whole minutes passed since the last hour. A single-digit minute is formatted without a leading zero.
Mm	Represents the minute as a number from 00 through 59. The minute represents whole minutes passed since the last hour. A single-digit minute is formatted with a leading zero.
M	Represents the month as a number from 1 through 12. A single-digit month is formatted without a leading zero.
MM	Represents the month as a number from 01 through 12. A single-digit month is formatted with a leading zero.
MMM	Represents the abbreviated name of the month.
MMMM	Represents the full name of the month.
s	Represents the seconds as a number from 0 through 59. The second represents whole seconds passed since the last minute. A single-digit second is formatted without a leading zero.
ss	Represents the seconds as a number from 00 through 59. The second represents whole seconds passed since the last minute. A single-digit second is formatted with a leading zero.
t	Represents the first character of the A.M./P.M. designator. The A.M. designator is used if the hour in the time being formatted is less than 12; otherwise, the P.M. designator is used.
tt	Represents the A.M./P.M. designator. The A.M. designator is used if the hour in the time being formatted is less than 12; otherwise, the P.M. designator is used.
y	Represents the year as a two-digit number.
yy	Represents the year as a two-digit number.
yyy	Represents the year as a three-digit number.
yyyy	Represents the year as a four-digit number.
yyyyy	Represents the year as a five-digit number. If the year has fewer than five digits, the number is padded with leading zeroes to achieve five digits. If there are additional "y" specifiers, the number is padded with as many leading zeroes as necessary to achieve the number of "y" specifiers.
:	The time separator that is used to differentiate hours, minutes, and seconds.
/	The date separator that is used to differentiate years, months, and days.
"	Quoted string (quotation mark). Displays the literal value of any string between two quotation marks ("). Precede each quotation mark with an escape character (\).
'	Quoted string (apostrophe). Displays the literal value of any string between two apostrophe (') characters.
%c	Represents the result associated with a custom format specifier "c", when the custom format value consists solely of that custom format specifier. That is, to use the "d", "f", "F", "h", "m", "s", "t", "y", "z", "H", or "M" custom format specifier by itself, specify "%d", "%f", "%F", "%h", "%m", "%s", "%t", "%y", "%z", "%H", or "%M".
\c	The escape character. Displays the character "c" as a literal when that character is preceded by the escape character (\). To insert the backslash character itself in the result string, use two escape characters ("\\").
Any other character	Any other character is copied to the result string, and does not affect formatting.

Appendix C: Numeric Format Values

When formatting numeric values, there are a number of built-in format options from which to choose. If you require more flexibility, you can also create a custom numeric format. The standard format specifiers can have a number appended to specify the precision (i.e. c0 could produce \$10, while c3 could produce \$10.120)

Standard Numeric Formats³

Format Specifier	Name	American English Examples		
		Specifier	Value	Result
C or c	Currency	C	123.456	\$123.46
		C3	123.456	\$123.456
		C	1	\$1.00
D or d	Decimal	D	123	123
		D4	123	0123
		D	-123	-123
E or e	Scientific (exponential)	E	1	1.000000E+000
		e1	1	1.0e+000
		e2	-123.4	-1.23e+002
F or f	Fixed-point	F	1	1.00
		F1	1	1.0
		F1	-1.56	-1.6
G or g	General	G	-123.456	-123.456
		G5	123.456	123.46
		G	.000012	1.2E-05
		G	.0012	0.0012
		G2	1234	1.2E+03
N or n	Number	N	-123.456	-123.46
		N	1.23456	1.23

³ From <http://msdn.microsoft.com/en-us/library/dwhawy9k.aspx>

		N	12345.6	12,345.60
		N1	123	123.0
		N1	123.456	123.5
P or p	Percent	P	.123456	12.35%
		P1	.123456	12.3%
		P	-12.3456	-1,234.56%
		P0	1	100%
R or r	Round-trip	R0	1.01	1.01
		R	1.623e-21	1.623E-21
X or x	Hexadecimal	X	12	C
		X3	17	011
		x	-1	ffffffff
Any other single character	Unknown format value	Displays an error message		

Custom Numeric Formats⁴

Format Specifier	Description
0	If the value being formatted has a digit in the position where the '0' appears in the format string, then that digit is copied to the result string; otherwise, a '0' appears in the result string.
#	If the value being formatted has a digit in the position where the '#' appears in the format string, then that digit is copied to the result string. Otherwise, nothing is stored in that position in the result string.
.	The first '.' character in the format string determines the location of the decimal separator in the formatted value; any additional '.' characters are ignored.
,	If one or more ',' characters is specified between two digit placeholders (0 or #) that format the integral digits of a number, a group separator character is inserted between each number group in the integral part of the output. If one or more ',' characters is specified immediately to the left of the explicit or implicit decimal point, the number to be formatted is divided by 1000 each time a number scaling specifier occurs. For example, if the string "0,," is used to format the number 100 million, the output is "100".
%	The presence of a '%' character in a format string causes a number to be multiplied by 100 before it is formatted. The appropriate symbol is inserted in the number itself at the location where the '%' appears in the format string.
%o	The presence of a '%o' character in a format string causes a number to be multiplied by 1000 before it is formatted. The appropriate symbol is inserted in the number itself at the location where the '%o' appears in the format string.
E0 E+0 E-0 e0 e+0 e-0	If any of the strings "E", "E+", "E-", "e", "e+", or "e-" are present in the format string and are followed immediately by at least one '0' character, then the number is formatted using scientific notation with an 'E' or 'e' inserted between the number and the exponent. The number of '0' characters following the scientific notation indicator determines the minimum number of digits to output for the exponent. The "E+" and "e+" formats indicate that a sign character (plus or minus) should always precede the exponent. The "E", "E-", "e", or "e-" formats indicate that a sign character should only precede negative exponents.
;	The ';' character is used to separate sections for positive, negative, and zero numbers in the format string. If there are two sections in the custom format string, the leftmost section defines the formatting of positive and zero numbers, while the rightmost section defines the formatting of negative numbers. If there are three sections, the leftmost section defines the formatting of positive numbers, the middle section defines the formatting of negative numbers, and the rightmost section defines the formatting of zero numbers.
"	Quoted string (quotation mark). Displays the literal value of any string between two quotation marks ("). Precede each quotation mark with an escape character (\).
'	Quoted string (apostrophe). Displays the literal value of any string between two apostrophe (') characters.
\%	The escape character. Displays the character "%" as a literal when that character is preceded by the escape character (\). To insert the backslash character itself in the result string, use two escape characters ("\\").
Any other character	Any other character is copied to the result string, and does not affect formatting.

⁴ From <http://msdn.microsoft.com/en-us/library/0c899ak8.aspx>

