

## Appendix K

# Predefined Components

---

This appendix lists and describes the predefined components that install with SitePublisher. Examples of each of these components can be seen in the component-guide site that is installed in the iw-foundation branch, under default/main.

- Using Components
- Analytics Integration Components
- Developer Tool Components
- Digital Asset Components
- Marketing Components
- Page Element Components
- Component Equivalencies

## Using Components

The predefined components are available for use on any site or product as is, or they can be customized as desired. Modifying the original files is not supported. Instead, make copies outside of the Interwoven specific directories and customize those copies. This applies to the following files:

- .component files in iwadmin /main/livesite/component/Interwoven
- .template files in iwadmin /main/livesite/template/Interwoven
- .js, .css, .gif, .jpg, .xsl and other files under /iw/resources
- .java files in any child package of com.interwoven

This allows future upgrades and patches to TeamSite SitePublisher to be installed without overwriting any existing customizations.

When copying components to other branches or sites, make sure the related resources are also copied into the new branch or site or errors may result. Component resources are located under /iw/resources. Copy this entire directory tree into the new location.

Component instances can be resized and have their position and background color changed on the Layout tab of the component's properties. Other style changes (for example, borders, padding, margins, and font selection) can be specified for most

components with web-standard Cascading Style Sheets (CSS). The Appearance tab will accept one or more CSS class names that can be defined in a .css file associated with your site. In cases where the Content tab includes replicants that require individual styling, the class name fields will be found within each replicant section.

## Analytics Integration Components

The following components are found in the /Interwoven/Analytics Integration folder.

### General Analytics

Enables tracking and analysis using third party analytics tools.

Requires:

- LiveSite: Yes
- Targeting: No

/Interwoven/Analytics Integration/General Analytics.component

The General Analytics component provides a base set of functionality for adding web analytics to a site or page. It can be extended to support multiple analytics products, such as Google Analytics.

### Google Analytics

Enables tracking and analysis using Google Analytics.

Requires:

- LiveSite: Yes
- Targeting: No

/Interwoven/Analytics Integration/Google Analytics.component

The Google Analytics component adds web analytics to a site or page. It is based on the General Analytics component, and custom tailored to work with Google's product.

The necessary JavaScript is added to the end of the page using an external class to embed the contents of a given HTML snippet. Including it inline via XSL at the component's position or order on the page would not ensure proper placement necessary for Google's code to properly execute on the page.

Use of this component requires an account from Google.

## HBX Analytics

Enables tracking and analysis using HBX.

Requires:

- LiveSite: Yes
- Targeting: No

```
/Interwoven/Analytics Integration/HBX Analytics.component
```

The HBX Analytics component adds web analytics to a site or page. It is based on the General Analytics component, and custom tailored to work with HBX.

The necessary JavaScript is added to the end of the page using an external class to embed the contents of a given HTML snippet. Including it inline via XSL at the component's position or order on the page would not ensure proper placement necessary for HBX's code to properly execute on the page.

Use of this component requires an account from Visual Sciences or Omniture.

The vendor is phasing out support (end of life) for HBX Analytics.

## SiteCatalyst Analytics

Enables tracking and analysis using Omniture SiteCatalyst.

Requires:

- LiveSite: Yes
- Targeting: No

```
/Interwoven/Analytics Integration/SiteCatalyst Analytics.component
```

The SiteCatalyst Analytics component adds web analytics to a site or page. It is based on the General Analytics component, and custom tailored to work with Omniture's product.

The necessary JavaScript is added to the end of the page using an external class to embed the contents of a given HTML snippet. Including it inline via XSL at the component's position or order on the page would not ensure proper placement necessary for SiteCatalyst's code to properly execute on the page.

Use of this component requires an account from Omniture.

## WebTrends Analytics

Enables tracking and analysis using WebTrends.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Analytics Integration/WebTrends Analytics.component`

The WebTrends Analytics component adds web analytics to a site or page. It is based on the General Analytics component, and custom tailored to work with WebTrends.

The necessary JavaScript is added to the end of the page using an external class to embed the contents of a given HTML snippet. Including it inline via XSL at the component's position or order on the page would not ensure proper placement necessary for WebTrends' code to properly execute on the page.

Use of this component requires an account from WebTrends.

## Developer Tool Components

The following components are found in the `/Interwoven/Developer Tool/Testing` folder.

### Display HTTP Header

Displays the HTTP header for a URL.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Developer Tool/Testing/Display HTTP Header.component`

The Display HTTP Header component examines the HTTP headers for a given web address and displays them.

```
Connection=close
Set-Cookie=JSESSIONID=57DDAE6F5F141BA0B135D2860E75A7FA; Path=/
Date=Mon, 02 Jun 2008 20:09:57 GMT
Server=Apache/2.0.52 (Red Hat)
Content-Type=text/html; charset=UTF-8
Transfer-Encoding=chunked
```

## Dump Context

Displays RequestContext details.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Developer Tool/Testing.component`

The Dump Context component displays a tree of the RequestContext object details for the page.

## Generate Context XML

Displays RequestContext details in XML.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Developer Tool/Testing/Generate Context XML.component`

The Generate Context XML component displays an XML rendition of the RequestContext object details for the page.

See the TeamSite SitePublisher Component Guide for sample XML.

## Digital Asset Components

The following components are found in the `/Interwoven/Digital Asset` folder.

### Advanced Flash

Embeds a Flash SWF file in a page with parameters.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Digital Asset/Advanced Flash.component`

The Advanced Flash component allows the page author to embed an Adobe Flash movie or animation (a `.swf` file) into a page and provide any number of nested parameters to be passed to the Flash plugin via `<param>` tags. Each tag is a replicant consisting of a name and value.

The default behavior provides two common parameters:

- Name: quality  
Value: high
- Name: wmode  
Value: opaque

The “opaque” setting for wmode forces the Flash plugin to treat its playback window as an opaque CSS object, which allows other HTML content to overlap it when properly z-indexed. A common use, as shown in this page, is to allow a CSS or DHTML navigation menu to overlap the SWF file playback region.

## Basic Flash

Embeds a Flash SWF file in a page.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Digital Asset/Flash/Basic Flash.component`

The Basic Flash component allows the page author to embed an Adobe Flash movie or animation (a `.swf` file) into a page. If the Flash plugin is not detected on the site visitor's browser, the user will be prompted to download and install it. Default Flash parameter values are set for playback quality (high) and wmode (to allow masking).

## Basic Image

Displays the specified image.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Digital Asset/Image/Basic Image.component`

The Basic Image component displays the specified GIF, JPG, or PNG image within a page. A CSS class can be specified to size the image, add a border, or align the image within the component's boundaries.

## Basic Image Gallery

Displays a collection of images at their natural size.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Digital Asset/Image/Basic Image Gallery.component`

The Basic Image Gallery component displays a collection of small preview images or thumbnails. Thumbnails can be set to small, medium, or large size, and also be positioned relative to the main image area.

When a thumbnail is selected, the corresponding image is shown at its natural size in the main image area. This main image area can be linked to another page, and optionally passed a selected data record associated with the selected image, for example to display that data record using another component on the destination page.

The JavaScript for this component is located in

`/iw/resources/scripts/foundation/image-gallery.js`. This file must be available on pages that contain this component.

## Fixed Size Image Gallery

Displays a collection of images at a fixed size.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Digital Asset/Image/Fixed Size Image Gallery.component`

The Fixed Size Image Gallery component displays a collection of small preview images or thumbnails. Thumbnails can be set to small, medium, or large size, and also be positioned relative to the main image area.

When a thumbnail is selected, the corresponding image is shown at a specified fixed size in the main image area. Smaller images will be scaled up, and larger images will be scaled down. This main image area can be linked to another page, and optionally passed a selected data record associated with the selected image, for example to display that data record using another component on the destination page.

The JavaScript for this component is located in

`/iw/resources/scripts/foundation/image-gallery.js`. This file must be available on pages that contain this component.

## Image Slideshow with Graphic Buttons

Displays a series of images in a slideshow with image-based button controls.

Requires:

- LiveSite: No
- Targeting: No

/Interwoven/Digital Asset/Image/Image Slideshow with Graphic Buttons.component

The Image Slideshow with Graphic Buttons component cycles through a series of selected images in a slideshow format, showing each image or slide for a specified period of time before automatically advancing to the next slide. Once the final slide is reached, playback loops around and resumes at the first slide.

Each slide can be linked to a different page or URL, and the duration between slide advances is also variable.

Controls to fast forward, reverse, and pause or resume playback of the slides are available. Each control is a image and can be further styled with CSS.

This component requires the

/iw/resources/scripts/foundation/image-slideshow.js and /iw/resources/scripts/foundation/effects.js files be available in any page where it is added.

## Image Slideshow with Text Buttons

Displays a series of images in a slideshow with text-based button controls.

Requires:

- LiveSite: No
- Targeting: No

/Interwoven/Digital Asset/Image/Image Slideshow with Text Buttons.component

The Image Slideshow with Text Buttons component cycles through a series of selected images in a slideshow format, showing each image or slide for a specified period of time before automatically advancing to the next slide. Once the final slide is reached, playback loops around and resumes at the first slide.

Each slide can be linked to a different page or URL, and the duration between slide advances is also variable.

Controls to fast forward, reverse, and pause or resume playback of the slides are available. Each control is a button with an editable text label and can be further styled with CSS.

This component requires the

`/iw/resources/scripts/foundation/image-slideshow.js` and  
`/iw/resources/scripts/foundation/effects.js` files be available in any page where it is added.

## Large Image Gallery

Displays a collection of images in a fixed area with optional scrollbars.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Digital Asset/Image/Large Image Gallery.component`

The Large Image Gallery component displays a collection of small preview images or thumbnails. Thumbnails can be set to small, medium, or large size, and also be positioned relative to the main image area.

When a thumbnail is selected, the corresponding image is shown at a specified fixed size in the main image area. Images are shown at their native size. If images are larger than the image area, scrollbars will be added automatically. This main image area can be linked to another page, and optionally passed a selected data record associated with the selected image, for example to display that data record using another component on the destination page.

The JavaScript for this component is located in

`/iw/resources/scripts/foundation/image-gallery.js`. This file must be available on pages that contain this component.

## Linked Image

Displays a linked image.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Digital Asset/Image/Linked Image.component`

The Linked Image component displays a specified image, and links it to a specified URL. When the page visitor clicks the image, they are taken to the URL.



## Linked Rollover Image

Displays a linked image that is replaced with an alternate image when moused over.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Digital Asset/Image/Linked Rollover Image.component`

The Linked Rollover Image component displays a specified image, and links it to a specified URL. When the mouse cursor is moved over the image, a replacement image is shown. When the page visitor clicks the image, they are taken to the URL.

## Marketing Components

The following components are found in the `/Interwoven/Marketing` folder.

### Basic Content Numbered Slideshow

Displays a numbered slideshow of rich text and images.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Advanced/Basic Content Numbered Slideshow.component`

The Basic Content Numbered Slideshow component plays through a series of slides, each comprised of a Textarea Datum, which can contain text, hyperlinks, images, or any other arbitrary HTML content. A series of numbered buttons are displayed below the slides to allow the reader to jump directly to any slide. A label field is provided, which should describe the elements of the slideshow.

Specify a default height in pixels using the Slide Height field on the Appearance tab. This will reposition the buttons below the slides properly. Scrollbars can be added for content that exceeds this height using a standard CSS class and referencing it in the Slide CSS Class field.

Once a slide is selected by pressing its button, the slideshow stops playback. Playback will resume the next time the page is opened.

Pages using this component must include the

`/iw/resources/scripts/foundation/content-slideshow.js` and  
`/iw/resources/scripts/foundation/effects.js` files.



## Basic Content Slideshow

Displays a slideshow of rich text and images.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Advanced/Basic Content Slideshow.component`

The Basic Content Slideshow component plays through a series of slides, each comprised of a Textarea Datum, which can contain text, hyperlinks, images, or any other arbitrary HTML content. A series of VCR style buttons are displayed below the slides to allow the reader to navigate through the slides.

Specify a default height in pixels using the **Slide Height** field on the Appearance tab. This will reposition the buttons below the slides properly. Scrollbars can be added for content that exceeds this height using a standard CSS class and referencing it in the **Slide CSS Class** field.

Pages using this component must include the

`/iw/resources/scripts/foundation/content-slideshow.js` and  
`/iw/resources/scripts/foundation/effects.js` files.

## Basic Embed HTML

Displays HTML markup as entered.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Advanced/Basic Embed HTML.component`

The Basic Embed HTML component allows page authors to enter in HTML by hand or by pasting from an existing source. It is then rendered from the markup.

This is an alternative to creating a new component from existing HTML or using the **Edit HTML Source** option within the Rich Text or Direct Content components. The advantages are that it can be done by any page author, unlike creating new components; and formatting and whitespace within the HTML are preserved better than in the other methods. The tradeoff is that some complex HTML may not render precisely across different browsers.

Since paragraph tags are stripped to remove rich text editor formatting, use the `sticky` attribute in your `p` tags to ensure they are correctly kept, like this: `<p sticky>your paragraph</p sticky>`.

Break tags are also removed, but only if written b-r-space-slash like this `<br />`. Just omit the space to ensure your breaks are kept like `<br/>` this. You can also use the sticky attribute on your br like this `<br sticky />` if desired.

Lastly, anything resembling a URL will be transformed by the rich text editor into a link, even though when pasting in HTML it is likely a quoted href for an anchor tag, not a plain string you want to be auto-linked. If your HTML contains URLs within anchor tag hrefs, simply click them and choose the unlink control button in the rich text editor to undo the auto-link.

## Basic IFrame

Displays content in an inline, scrollable frame.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Advanced/Basic IFrame.component`

The Basic IFrame component displays the contents of a given URL inline within another page. Scrollbars are optionally available if the inline content is larger than the space available in the containing page.

## Embed HTML with Ajax

Loads HTML inline within a page asynchronously.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Advanced/Embed HTML with Ajax.component`

The Embed HTML with Ajax component is an alternative method for including inline content. It uses JavaScript to pull the inline content as a second server request and embeds it within a DIV in the containing page. Content embedded with this component should not consist of complete web pages; rather, they should consist of simple fragments of HTML or plain strings that will be added dynamically to the existing page, for example to add boilerplate text or a search form widget.

Because of browser security restrictions, content loaded in this manner should come from the same server as the wrapping page; loading content from a different host or domain will result in warnings or errors for the site user in most cases.

## Advanced Stacked Content

Displays a column of basic content entries and images in any order; images can exceed the component width.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Content/Advanced Stacked Content.component`

The Advanced Stacked Content component displays a column of replicant data. Each replicant can consist of either rich text, an image, or both.

The text wraps when it reaches the edge of the component. Images can be any width, regardless of the component width. If an image is wider than the component, it overflows onto the rest of the page, and can even flow on top of other components on the page.

The overflow capability is only visible in the runtime site or in preview, not in the Page Editor.

## Basic Content

Displays rich text and images.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Content/Basic Content.component`

The Basic Content component displays rich formatted text and images. It can be styled using the Textarea Datum controls, or by selecting a CSS class, or by entering CSS style information directly.

## Content with Style Selector

Displays basic content with styles selected by the site visitor.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Content/Content with Style Selector.component`

The Content with Style Selector component is similar to the Basic Content component, but in addition it presents a style toolbar to the site visitor, allowing them to choose a CSS style to apply to the content. This is commonly used to allow users to change the size of fonts in a component for easier reading or printing.

## FormsPublisher Content

Displays rich text and images from a TeamSite FormsPublisher Form.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Content/FormsPublisher Content.component`

The FormsPublisher Content component displays data stored within a data record. It also provides styling controls to specify a CSS class, id, or additional inline style elements for the output. When no data record is specified, a text message is displayed prompting the page author to select one.

This component can be used as a model for converting most other components that use directly entered data into data record driven versions.

This component is designed to be a baseline starting point or example for other implementations. As each DCT has different fields, and the desired output from those fields also varies, a customized version of this component will need to be created for each form type.

## Stacked Content

Displays a column of basic content entries and images in any order.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Content/Stacked Content.component`

The Stacked Content component displays a column of replicant data. Each replicant can consist of either rich text, an image, or both. Using CSS the images and text can be styled and aligned uniquely for each instance of the component.

## Email

Generates and sends an email based on the fields from a posted form.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Marketing/Form Builder/Email.component`

The Email component generates an email, commonly with form results, and sends it through the specified SMTP server. Handling of the email is performed through an external method in the EmailService class.

Four pages are required to use the Email component. The first page contains a form that is submitted to the second page, which contains an instance of the Email component. After that page is processed, the user is automatically redirected to one of the final two pages. One page is reached in the case of successfully completed operations, and the other in the case where an error was encountered.

This component does not have a preview mode and email sending operations are only supported in the runtime environment.

## Datum Configuration

**Subject Field Name** - The name of the field in the posted form containing the subject value.

**Sender Field Name** - The name of the field in the posted form containing the email address of the person submitting the form.

**Default Mail Recipient** - The target email address of user or alias that will receive all emails submitted through this component. Note that some servers do not allow anonymous SMTP. Therefore the most reliable way to test is either with a local SMTP server such as Postfix or James, or using a corporate SMTP server from within the firewall.

**SMTP Server Address** - The address of the SMTP server used to transport the email. Examples are "127.0.0.1", "legalist", or "www.domain.com". Note that some servers do not allow anonymous SMTP. Therefore the most reliable way to test is either with a local SMTP server such as Postfix or James, or using a corporate SMTP server from within the firewall.

**SMTP Port** - The port on the SMTP server. Note that some ISPs block port 25. To make sure that the server running the LiveSite instance has access to both the server and the port, try telneting to the desired server from a command prompt on the LiveSite machine.

**Debug Logging** - This field is used to toggle the Java Mail API's debug mode. When set to "Yes", log file output will have statements prefixed with the word "DEBUG". Note that the log output will contain debug statements that may not necessarily be a problem. If the need arises to examine log files, pay attention first to any "ERROR" statements.

**MimeType** - The MIME type used for the email. The default is text.

**Email Body Transform XSL (Optional)** - The path to the XSL file that can be used to transform the fields of the submitted form page into a specialized email body. If an XSL is used to generate an HTML body, be sure to set the MimeType accordingly. The default format of the email in cases where the XSL is not provided is the format of "field name: field value". Click here to see an example of the internal XML format.

**Success URL** - The page or URL that the user should be forwarded to when the email is successfully processed.

**Error URL** - The page or URL that the user should be forwarded to when there was an error processing the form or connecting to the SMTP server.

## Feedback Form Example

An example form built with the Form Builder components.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Marketing/Form Builder/*.component`

This page contains example implementations of the Form Builder components. It provides users the ability to submit their comments and suggestions for a site. The form is submitted using the Email component page, and the user is then redirected to a Thank You page.

To use the Form Builder components, the `form-builder.js` and `AjaxRequest.class.js` files from `/iw/resources/scripts/foundation/` must be available in the page.

The Email component page must be configured with an available SMTP server for this form submission to generate an email. This field is blank by default.

## Basic Bulleted List

Displays a bulleted list of text items.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/List/Basic Bulleted List.component`

The Basic Bulleted List component displays a list of text entries in the order entered. Each entry is a replicant in the component.

By applying CSS to this component, the default appearance, bullets, can be replaced with other types of HTML lists that are available using the `<UL>` tag.

## Bulleted Content List

Displays a bulleted list of basic content.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/List/Bulleted Content List.component`

The Bulleted Content List component displays a list of rich text and image entries in the order entered. Each entry is a replicant in the component.

By applying CSS to this component, the default appearance, bullets, can be replaced with other types of HTML lists that are available using the `<UL>` tag. In addition, the rich text itself can be styled individually using the toolbar.

## Dynamic Asset List

Displays a list of links to files for a directory on the server.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Marketing/List/Dynamic Asset List.component`

The Dynamic Asset List component displays a list of assets (files) on the server using a given starting directory and a parameter to indicate how many levels of subdirectories to

show. Each found file name is displayed with a corresponding icon based on the file type or extension, and is linked to the file itself, for downloading.

## Expandable File Tree List

Displays a dynamic file and folder tree from a directory on the server.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Marketing/List/Expandable File Tree List.component`

The Expandable File Tree List component shows files on the server from a specified starting directory. Ajax is used to fetch the contents of each nested subdirectory when its folder is opened.

Certain CSS values are stored within the `.component` file because changes to them could impact the functionality of the tree. Care should be used when modifying the styles for this component.

The `/iw/resources/scripts/foundation/AjaxRequest.class.js` must be available on a page that includes this component.

## Linkable Bulleted List

Displays a bulleted list of linked text items.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/List/Linkable Bulleted List.component`

The Linkable Bulleted List component displays a list of text entries in the order entered. Each entry is a replicant in the component and can be linked to a URL.

By applying CSS to this component, the default appearance, bullets, can be replaced with other types of HTML lists that are available using the `<UL>` tag.

## Basic Promotion

Displays a promotional image with a link to the promotion details.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Marketing/Promotion/Basic Promotion.component`

The Basic Promotion component displays a linked image, that when clicked takes the user to a specified page or URL.

## Promotion

Displays a promotion from a TeamSite FormsPublisher form and links to the promotion details.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Marketing/Promotion/Promotion.component`

The Promotion component displays the contents of a TeamSite FormsPublisher form-based promotion. The list of related items is built by selecting data records, and can be sorted by any field in the data record's content (for example, Title or Publish Date). Each item will then be linked to a page and pass the data record as a parameter of the URL, typically for consumption by a page with an instance of the Promotion Details component, which will display the details for that data record.

The image from the data record can be set to appear either to the right or left of the rest of the content.

When re-editing an existing instance of this component, the selection boxes will reset to defaults. Reselect a workarea and category to recover the selected values before saving.

The associated CSS class results in multiple generated class names, one for each area of the promotion. The suffixes are `-headline`, `-image`, `-summary`, and `-details`. For example, entering a CSS class `Foo` would also result in a `Foo-image` class being used in the generated page, which could be defined in a `.css` stylesheet.

## Promotion Details

Displays promotion details from a TeamSite FormsPublisher form.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Marketing/Promotion/Promotion Details.component`

The Promotion Details component displays the details of a promotion, which is pulled from a data record. It works in tandem with the Promotion component by accepting a URL parameter from it to choose a data record to display.

This component shows the details of data records of type Product/SimplePromo. To display other data record types, make a copy of this component, save it with a different name, and edit the Appearance XSL to format the contents of the desired data record type.

## Promotion Slideshow

Displays a slideshow of promotions from TeamSite FormsPublisher forms.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Marketing/Promotion/Promotion Slideshow.component`

The Promotion Slideshow component displays TeamSite FormsPublisher form-based promotions in a slideshow format.

The component's Appearance XSL is currently designed to render data records that are of type SimplePromo. This component can render other data record structures as well. To do so, make a copy of this component and edit the appearance XSL by adding a template with a match statement that recognizes the category and type of the desired data record. Lastly, modify the data record datum in the content XML to allow users to select data records of the new category and type.

This component requires the `/iw/resources/scripts/foundation/effects.js` and `/iw/resources/scripts/foundation/content-slideshow.js` files to work.

---

## Page Element Components

The following components are found in the `/Interwoven/Page Element` folder.

### Advanced Button

Displays a button with a label, tooltip, click action, and style.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Button/Advanced Button.component`

The Advanced Button component creates a button using the provided label, tooltip text, CSS class and inline style information, and when clicked calls the given JavaScript action command. The label can contain HTML, not just plain text.

### Advanced Button with Rollover

Displays a rollover button with a label, tooltip, click action, and style.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/CSS Rollover Button.component`

The Advanced Button with Rollover component draws a button using a specified style or inline style parameters. When the button is hovered on by the mouse an alternate style is applied. When clicked, a third style is applied, and a specified JavaScript action is initiated. After moving the mouse cursor off the button it returns to the original style.

Specify a label using HTML, a tooltip, inline constant style information, and the default CSS class. The two other styles are referenced using the default name plus suffixes: `-hover` and `-active`. For example if the default class `foo` were specified in the component, the style sheet should define the classes `foo`, `foo-hover`, and `foo-active`.

## Basic Button

Displays a button that loads a specified URL.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Button/Basic Button.component`

The Basic Button component creates a linked button. Clicking the button takes the user to the specified URL or Page. The button can be customized with any label or tooltip, and any CSS class or inline style data.

Labels of the buttons can include HTML markup.

## Horizontal Line

Displays a styled horizontal divider line.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Design/Horizontal Line.component`

The Horizontal Line component draws a solid horizontal line across the full width of the component. Color and line thickness can be adjusted within the Appearance tab.

Similar effects could be achieved by adding CSS to an existing component, inserting an `<hr>` tag, using an image of a line, or adding an empty component with the desired background color and dimensions.

## Page Style Selector

Displays page content with persistent, selectable styles.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Design/Page Style Selector.component`

The Page Style Selector component enables the user to select from multiple style sheets (CSS files) on the server to stylize the site content to their liking. This can be used, for

example, to provide a larger default text size, alternate color schemes, or high-contrast styles for better Section 508 compliance.

The chosen style will apply to the page where the component is added, affecting all components on that page. If a copy of the component is placed on multiple pages or added to a template that is applied to multiple pages, the same selection can apply to all pages if desired.

User selections are persisted in a cookie. The page author can select a name (key) for this cookie. Using the same cookie name on multiple pages will result in those pages sharing the same style selection. Using different names allows different options to be saved. Multiple instances of the component can be added to a single page, each with a different cookie name, if a combination of styles is needed.

Some examples:

- Add the component to one page only. Result: the style preference only applies to that page.
- Add the component to all pages of a site, always using the same cookie name. Result: all pages get the same style preference.
- Add the component to two templates A & B, using a different cookie name in each. Result: pages made with template A inherit one style sheet; pages made from template B inherit the other.
- Add two instances to the same page or template, one for coloring and one for text size, using two different cookie names. Result: two CSS files are applied allowing the user to pick any combination of colors and text sizes.

Two skins are available. The default skin shows a control for selecting styles and applies the chosen style when clicked and on future page views. An alternate skin is available that simply applies any previously selected style sheet preference without showing the control. This can be useful, for example, to free up screen space if the change control is only needed on a user preferences page.

Style sheets used by this component should not be specified as site resources like other CSS files.

## Remove Inline Colors

Strips color attributes from HTML tags for components and pages, including component contents.

Requires:

- LiveSite: No
- Targeting: No

/Interwoven/Page Element/Design/Remove Inline Colors.component

The Remove Inline Colors component uses JavaScript to strip the color attributes from HTML tags that commonly specify a color, thereby allowing color choices specified in CSS to take effect. It does not target inline CSS coloring. Removing the inline colors allows rapid changes to site color schemes without having to revisit each component on a page to update its color.

This component has no requirement to be visible on a page, so defaults to a small size that can be placed out of the way of other components on a page, typically as part of a template. Note that although the component is set to a red background (visible when editing this page), it appears to blend in with the page because the color data is stripped when the page loads.

Selecting transparent for component and page backgrounds often eliminates the need for this component.

## Remove Inline Component Colors

Strips color attributes from HTML tags for components and pages.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Design/Remove Inline Component Colors.component`

The Remove Inline Component Colors component uses JavaScript to strip the color attributes from pages and components, but attempts to not modify other HTML tags that commonly specify a color, thereby allowing color choices specified in CSS to take effect for components and pages, but leaving color information from other sources untouched. Inline CSS style colors are not affected by this component. Removing component colors allows rapid changes to site color schemes without having to revisit each component on a page to update its color.

This component has no requirement to be visible on a page, so defaults to a small size that can be placed out of the way of other components on a page, typically as part of a template. Note that although the component is set to a red background (visible when editing this page), it appears to blend in with the page because the color data is stripped when the page loads.

Selecting transparent for component and page backgrounds often eliminates the need for this component.

## Vertical Line

Displays a styled vertical divider line.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Design/Vertical Line.component`

The Vertical Line component draws a solid vertical line down the full height of the component. Color and line thickness can be adjusted within the Appearance tab.

Similar effects could be achieved by adding CSS to an existing component, using an image of a line, or adding an empty component with the desired background color and dimensions.

## Basic Linked Footer

Displays a footer including a copyright and multiple links.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Footer/Basic Linked Footer.component`

The Basic Linked Footer component displays a copyright message and one or more footer links. The text, link destinations, separator characters between the links, and the CSS can all be adjusted.

## Copyright Footer

Displays a copyright message with CSS styling for use in a footer.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Footer/Copyright Footer.component`

The Copyright Footer component simply displays an editable copyright message intended to appear in the page footer, and allows the page author to specify a few CSS classes that apply to the text and its surroundings.

## Linked Footer with Separators

Displays a footer including a copyright and multiple links, each separated by an image.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Footer/Linked Footer with Separators.component`

The **Linked Footer with Separators** displays a copyright message and one or more footer links. The text, link destinations, separator images between the links, and the CSS can all be adjusted.

## Three Row Footer

Displays a footer on three lines.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Footer/Three Row Footer.component`

The **Three Row Footer** component displays a set of links, plus two additional rows of text, one of which is typically used as a copyright. Each element can be customized using CSS by specifying CSS classes in the component and defining those classes in an external style sheet associated with the site or page.

## Two Row Footer

Displays a footer on two lines.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Footer/Two Row Footer.component`

The **Two Row Footer** component displays a set of links and a copyright message, stacked on two lines. All footer elements can be styled using an external CSS file and specified CSS class names in the component properties.

---

## Basic Image Header

Displays a header with multiple images.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Header/Basic Image Header.component`

The Basic Image Header component displays a logo and multiple additional images anywhere within the component's bounds using CSS. The logo is linked to the root of the site from which it is viewed.

## Basic Linked Image Header

Displays a header with multiple linked images.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Header/Basic Linked Image Header.component`

The Basic Linked Image Header displays one or more images within a header component, positioned using CSS. A logo is not automatically placed or linked - the entire header space is available for all images.

## Basic Logo Header

Displays a header with a logo image linked to the root of the site.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Header/Basic Logo Header.component`

The Basic Logo Header component displays the selected logo image using the specified CSS classname. The logo image is coded to link to the URL / of the site.

---

## Linked Rollover Image Header

Displays a header with multiple linked rollover images.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Header/Linked Rollover Image Header.component`

The Linked Rollover Image Header displays one or more linked images in a header with rollover alternate images. The images can be positioned using CSS. Alternate images are shown only if an image is linked.

## One Line Heading

Displays a one line text heading.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Header/One Line Heading.component`

The One Line Heading component simply displays a text heading for a section or page. It can be styled with a CSS class.

## Two Line Heading

Displays a heading and subheading.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Header/Two Line Heading.component`

The Two Line Heading component displays a title and subtitle with the prescribed CSS class. This can be used to title a page, or a section of a page.

## Edit This Page

Adds an **Edit this page** button.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Page Element/Page Edit.component`

The Page Edit component renders a link on the live production page for the SitePublisher Page Editor environment. This feature is not available from the edit and preview areas.

The link can have any text label and any CSS class applied. Optional left and right styles can be applied to areas surrounding the link, for example to draw two endcaps on a button. These styles are referenced using the main style name plus `-left` and `-right`. For example if the CSS class `foo` were entered, styles `foo-left` and `foo-right` become available.

## In Context Edit

Adds an **Edit this page** button for in-context editing of components.Requires

- LiveSite: Yes
- Targeting: No

`/Interwoven/Page Element/In Context Edit.component`

The In Context Edit component renders a link on the live production page for the SitePublisher Page Editor environment to open with the Edit Mode set to **In-Context Edit**. Not all components can be edited in this fashion, and this feature is not available from the edit and preview areas.

The link can have any text label and any CSS class applied. Optional left and right styles can be applied to areas surrounding the link, for example to draw two endcaps on a button. These styles are referenced using the main style name plus `-left` and `-right`. For example if the CSS class `foo` were entered, styles `foo-left` and `foo-right` become available.

## Basic Registration - Edit

Allows users to create and update an account summary.

Requires:

- LiveSite: Yes
- Targeting: No

/Interwoven/Page Element/Registration/Basic Registration.component

The Basic Registration component allows users to register, update, and view their profile on a LiveSite site. Upon registering, a user profile record is created in the LiveSite database, and assigns the user permission to login to the current site. This component is intended to be used on pages that are not access controlled, thereby allowing new users to complete the registration process without the aid of a site administrator.

Two skins are provided. The default skin allows creation and editing. A read-only skin for viewing is also available. It provides an Edit button that takes the user to a designed edit page.

## Basic Registration - View

Allows users to view an account summary.

Requires:

- LiveSite: Yes
- Targeting: No

/Interwoven/Page Element/Registration/Basic Registration.component

The Basic Registration component allows users to view their profile information in a non-editable mode when the View Only skin is used. When this component is used on a page, it will automatically load the user profile of the person who is currently logged into the site. When the **Edit** button is clicked, the user will be forwarded to a page containing the Basic Registration component that has the default skin selected so that the user can modify their personal information.

## Login

Controls access to a site by requiring a user name and password.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Page Element/Registration/Login.component`

The Login component presents a login form with a field for the user name and password. When submitted, the credentials are checked against the registered site users. If a match is found, the user is logged in; otherwise, a login failure message is presented.

## Registered User Edit

Allows a site administrator to edit the profile of a registered user.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Page Element/Registration/Registered User Edit.component`

The Registered User Edit component allows users to modify and update the account summaries for other users in the LiveSite database.

This component is intended to be used in an administrative capacity in conjunction with the Registered User List component. This component should be used on pages that require the user to be logged in. Since the default LiveSite schema does not have a user table column that differentiates access level, a logical extension of this component in a real-world scenario would require validation of administrative privileges before allowing access to the page.

## Registered User List

Allows a site administrator to display and manage a list of registered users.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Page Element/Registration/Registered User List.component`

The Registered User List component allows a user to view the list of users stored in the LiveSite database.

This component is intended to be used in an administrative capacity in conjunction with the Registered User Edit component, and therefore the pages where it is used should require login, and in a typical case be extended to check the user's role.

Upon selecting a user, the administrator can delete or edit that user's profile. New profiles can also be created by selecting the New button.

## Horizontal Breadcrumb

Dynamically displays horizontal breadcrumb navigation based on the Site Map.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Page Element/Navigation/Horizontal Breadcrumb.component`

Displays the path of the current page, optionally including the top/home page.

## Horizontal Dropdown

Dynamically displays horizontal dropdown navigation based on the Site Map.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Page Element/Navigation/Horizontal Dropdown.component`

The Horizontal Dropdown component is a navigation control that displays the Site Map data as a dynamic nested menu of up to four levels. Node descriptions are shown as tooltips on menu items. The menu can be styled by modifying CSS in the `/iw/resources/styles/foundation/dropdown.css` file.

Two skins are available. The **quirks mode** skin works in most current and older browsers and requires some JavaScript. The CSS-only skin is only compatible with MSIE 7, Firefox 2.0, and other modern browsers but can still function even when scripting is disabled. To use the CSS-only skin, pages must be set to **strict mode**.

MSIE 6 has an issue with both skins when using nested menus within menus - the open menu may close unexpectedly even while the mouse is over the child submenu.

## Horizontal Site Map

Dynamically displays basic horizontal navigation based on the Site Map.

Requires:

- LiveSite: Yes
- Targeting: No

```
/Interwoven/Page Element/Navigation/Horizontal Site Map.component
```

The Horizontal Site Map component displays navigation links for the site based on the Site Map and styled using CSS.

Nodes are presented in single row. Only top level nodes are shown. If a node has a description, it will be used as a tooltip, visible when a user moves their mouse over that node. Parent folder nodes are not linked and represented with a different style. The current node (or its ancestor if it is not itself visible) is also given a separate style.

## Vertical Flyout

Dynamically displays vertical flyout navigation based on the Site Map.

Requires:

- LiveSite: Yes
- Targeting: No

```
/Interwoven/Page Element/Navigation/Vertical Flyout.component
```

The Vertical Flyout component is a navigation control that displays the Site Map data as a dynamic nested menu of up to four levels. Node descriptions are shown as tooltips on menu items. The menu can be styled by modifying CSS in the `/iw/resources/styles/foundation/flyout.css` file.

Two skins are available. The **quirks mode** skin works in most current and older browsers and requires some JavaScript. The CSS-only skin is only compatible with MSIE 7, Firefox 2.0, and other modern browsers but can still function even when scripting is disabled. To use the CSS-only skin, pages must be set to **strict mode**.

MSIE 6 has an issue with both skins when using nested menus within menus - the open menu may close unexpectedly even while the mouse is over the child submenu.

The appearance is similar to the Vertical JavaScript Flyout component used for navigation on this site. While that component does require scripting be enabled, it does not suffer from the MSIE 6 compatibility issue.

## Vertical JavaScript Flyout

Dynamically displays vertical flyout navigation based on the Site Map using JavaScript.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Page Element/Navigation/Vertical JavaScript Flyout.component`

The Vertical JavaScript Flyout component is a navigation control that displays the Site Map data as a dynamic nested menu of multiple levels. Node descriptions are shown as tooltips on menu items. The menu can be styled by modifying CSS in the `/iw/resources/styles/foundation/jsflyout.css` file.

Like the similar but scriptless Vertical Flyout component version, this navigation menu works in MSIE 7, Firefox 2.0, and most other modern browsers. However, unlike the scriptless version, this one is also supported on MSIE 6 and needs only a single skin for both **quirks mode** and **strict mode**. JavaScript support must be enabled within the browser to achieve these gains, however.

This component is used as the navigation control for the component Guide site.

## Vertical Site Map

Dynamically displays basic vertical navigation based on the Site Map.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Page Element/Navigation/Vertical Site Map.component`

The Vertical Site Map component displays navigation links for the site based on the Site Map and styled using CSS.

Nodes are presented in an indented list. If a node has a description, it will be used as a tooltip, visible when a user moves their mouse over that node. Parent folder nodes are not linked and represented with a different style. The current node is also given a separate style.

## Feedback

Allows site visitors to rate content thumbs up or thumbs down and displays the current average results.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Social Computing/Feedback.component`

Asks the user a question with a boolean answer (for example, “Was this helpful to you? yes/no”). Answers may be given inline, and aggregate results are displayed after the user submits an answer.

## Manual RSS Viewer

Displays the specified RSS feed.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Social Computing/Manual RSS Viewer.component`

The Manual RSS Viewer component presents the most recent items in a specified RSS feed. A maximum number of items to show can be specified, and CSS classes can be used to customize the appearance of the feed output.

## Polling

Allows site visitors to answer a poll and displays the results to date.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Social Computing/Polling.component`

The Polling component displays a set of options for the user to vote on. After voting for one of the candidates, the user is shown the voting totals so far, including their own vote. Authentication can be required or not, and the user may optionally be allowed to see the results to date before casting their own vote. After a configurable period of time, voting is closed and visitors see only the outcome.

---

## Rate This - 5 Star

Allows site visitors to rate content from 1-5 stars and displays the current average results.

Requires:

- LiveSite: No
- Targeting: No

`/Interwoven/Social Computing/Rate This - 5 Star.component`

The Rate This - 5 Star component is a typical rating component that allows an authenticated user to submit a rating for either the entire page or some content on the page. Because login is required, unauthenticated users will be given a link to login, while authenticated users will be allowed to submit a new rating.

Rating results are shown using a scale of one to five stars by default, but other images can be used, such as checkmarks, smiley faces, thumbs up, etc.

To add text describing the subject of the rating or to pose a lead in question, use the Basic Content component positioned adjacent to this component.

## Selectable RSS Viewer

Displays an RSS feed. A predefined list of RSS sources is provided for the page author.

Requires:

- LiveSite: Yes
- Targeting: No

`/Interwoven/Social Computing/Selectable RSS Viewer.component`

The Selectable RSS Viewer component is similar to the Manual RSS Viewer component, but rather than a text input field being shown to the page author for an RSS URL, they are presented with a select list provided by the component developer from which to select one.

## Component Equivalencies

If you are using any of the version 3.0 predefined components, see Table 6 to determine the nearest equivalent version 6.7.2 component. In some cases, the new component is functionally equivalent and can be used as a direct replacement. In other cases, the new component demonstrates equivalent techniques, SitePublisher APIs, or design patterns, but does not necessarily perform the same function in a page. Wherever possible, you should move to the version 6.7.2 predefined component to take advantage of future enhancements.

**Table 6** *Predefined Components Equivalencies*

<b>6.7.2 predefined component</b>	<b>3.0.1 predefined component equivalent</b>
Interwoven/Debug/Display HTTP Header.component	ShowHttpHeaderOfUrl.component
Interwoven/Debug/Dump Context.component	DumpContext.component
Interwoven/Debug/Generate Context XML.component	GenerateContextXML.component
Interwoven/Media/Flash/Basic Flash.component	Flash Movie.component
Interwoven/Media/Images/Linked Image.component	Basic Image.component
Interwoven/Media/HTML/Basic iFrame.component	Basic iFrame.component
Interwoven/Media/HTML/Embed HTML with Ajax.component	FetchUrl.component
Interwoven/Form Builder/Checkbox.component	Checkboxes.component
Interwoven/Form Builder/Multi Select List.component	Multi Select List.component
Interwoven/Form Builder/Radio Button.component	Radio Buttons.component
Interwoven/Form Builder/Single Select List.component	Single Select List.component
Interwoven/Form Builder/Reset Button.component	Reset Button.component
Interwoven/Form Builder/Submit Button.component	Submit Button.component
Interwoven/Form Builder/Text Area.component	Text Area.component
Interwoven/Form Builder/Text Field.component	Text Field.component
Interwoven/File Management/Dynamic Asset List.component	File List.component

**Table 6** *Predefined Components Equivalencies*

<b>6.7.2 predefined component</b>	<b>3.0.1 predefined component equivalent</b>
Interwoven/File Management/Expandable File Tree List.component	Ajax File Explorer.component
Interwoven/Lists/Linkable Bulleted List.component	Abc News Top Stories.component
Interwoven/Promotions/Promotion.component	Book Form Entry.component
Merged into Interwoven/Basic Content.component	Text Area.component Spiegel Online.component
Interwoven/Buttons/Advanced Button.component Interwoven/Buttons/Advanced Button with Rollover.component Interwoven/Buttons/Basic Button.component	Live Page Edit.component
Interwoven/Site Maps/Horizontal Site Map.component	Site Map Navigation.component
Interwoven/Site Maps/Vertical Site Map.component	Site Map Navigation.component
Interwoven/Editing/Edit This Page.component	Live Page Edit.component
Interwoven/Editing/In Context Edit.component	Live In-Context Edit.component
Interwoven/Access/Login.component	Login.component
Interwoven/RSS/Manual RSS Viewer.component	News Headline (Directory).component News Headline (2 Columns).component

**NOTE**

If a 3.0.1 component is not listed in Table 6, it is no longer being developed by Interwoven.

## Developer.spar

A `developer.spar` file is included with SitePublisher. You can install the `developer.spar` file using Archive Manager. For instructions, see “SitePublisher Archives” on page 235. The following components are included in the `developer.spar` file.

## Static Hello World

Displays text defined in the appearance XSL.

Developer Tool/Code Samples/Static Hello World.component

## Hello World

Displays text from a datum value.

Developer Tool/Code Samples/Hello World.component

## External Hello World

Displays text from external Java code.

Developer Tool/Code Samples/External Hello World.component

## Generator

Uses generators to dynamically populate options for selection lists used in Component Properties.

Developer Tool/Code Samples/Generator.component

## Set Page Title and Keywords

Sets the title of a page and adds keywords to the page metadata.

Developer Tool/Code Samples/Set Page Title and Keywords.component

## Debug

Displays context and tokens for debugging purposes.

Developer Tool/Utility/Debug.component

## Generator with Dependent Datums

Demonstrates the use of inter-dependent datums populated by generator methods.

Developer Tool/Code Samples/Generator with Dependent Datums.component

## Throw Exception

Purposely throws an exception to demonstrate LiveSite exception handling.

Developer Tool/Utility/Throw Exception.component