



ePublisher Platform Documentation

Published date: 12/16/2023



Table of Contents

WebWorks Reverb 2.0.....	4
Choosing a Skin.....	6
Using SASS To Customize Reverb Interface.....	11
Using Custom SASS files in Reverb Projects.....	12
Previewing Reverb Output.....	13
Delivering Reverb Output.....	14
Top-Level Groups in Reverb.....	15
Searching Output.....	16
Using Baggage Files.....	17
Indexing Baggage Files and External URLs.....	18
Using Tidy for Indexing HTML Pages.....	19
Assigning Relevance Weight to Your Source Documents Styles.....	20
Assigning Relevance Weight to Your HTML and PDF Baggage Files.....	21
Search Highlighting in Baggage Files.....	23
Searching with URL Method.....	24
TOC or Index with URL Method.....	25
Launching Context Sensitive Help for WebWorks Reverb 2.0.....	26
End-user requirements in WebWorks Reverb 2.0.....	27
Analytics Event Tracking in Reverb 2.0.....	28
'Was This Helpful?' Buttons.....	29
'Was This Search Helpful?' Buttons.....	30
Document Last Modified Date in Reverb.....	31
Drop-down Expand/Collapse All Toggle Button.....	32
Google Translate Button.....	33
Customizable Header and Footer.....	34
Custom TOC Menu Items.....	35
Customizing a Bullet Icon using Font Awesome.....	36
Using the url_maps.xml reference file.....	37
PDF - XSL-FO.....	38
Why use PDF - XSL-FO output format?.....	39
PDF-XSL-FO Page Regions.....	40
PDF XSL-FO Font Inclusions.....	42
Dynamic HTML.....	44
Dynamic HTML Output Viewer.....	46

Delivering Dynamic HTML.....	47
ePUB.....	48
ePUB Platforms.....	49
ePUB Considerations.....	50
Meta Data.....	51
Book Title and ID.....	52
Long Content.....	53
Page Styles.....	54
Tables.....	55
Cover.....	56
Syncing with Apple iPad.....	57
Eclipse Help.....	58
Eclipse Help Viewer.....	59
Delivering Eclipse Help.....	60
HTML Help.....	61
Benefits of Microsoft HTML Help.....	62
Restrictions and Requirements for Microsoft HTML Help.....	63
HTML Help Viewer.....	64
Toolbar Pane in HTML Help.....	65
Navigation Pane in HTML Help.....	66
Topic Pane in HTML Help.....	67
Topic Only View in HTML Help.....	68
HTML Help Workshop.....	69
HTML Help Project File (.hhp).....	70
HTML Help Contents File (.hhc).....	71
HTML Help Index File (.hhk).....	72
HTML Help Mapping File (.h).....	73
Delivering HTML Help.....	74
Oracle Help.....	75
Oracle Help Viewer.....	77
Oracle Help Files.....	78
Helpset .hs File in Oracle Help.....	79
Control .xml Files.....	80
Full Text Search .idx Index File.....	81
Manifest .mft File.....	82

Delivering Oracle Help.....	83
Sun JavaHelp.....	84
Sun JavaHelp Files.....	85
Helpset .hs File in Sun JavaHelp.....	86
Contents toc.xml File in Sun JavaHelp.....	88
Index ix.xml File in Sun JavaHelp.....	89
Map .jhm File in Sun JavaHelp.....	90
Delivering Sun JavaHelp.....	91
WebWorks Help.....	92
The Frameset View in WebWorks Help.....	93
Navigation Pane in WebWorks Help.....	94
Toolbar Pane in WebWorks Help.....	95
Topic Pane in WebWorks Help.....	96
Topic Only View in WebWorks Help.....	97
WebWorks Help Output Files.....	98
Delivering WebWorks Help.....	99
Searching WebWorks Help.....	100

WebWorks Reverb 2.0

- [Choosing a Skin](#)
- [Using SASS To Customize Reverb Interface](#)
- [Previewing Reverb Output](#)
- [Delivering Reverb Output](#)
- [Top-Level Groups in Reverb](#)
- [Searching Output](#)
- [Using Baggage Files](#)
- [Searching with URL Method](#)
- [TOC or Index with URL Method](#)
- [Launching Context Sensitive Help for WebWorks Reverb 2.0](#)
- [End-user requirements in WebWorks Reverb 2.0](#)
- [Analytics Event Tracking in Reverb 2.0](#)
- ['Was This Helpful?' Buttons](#)
- ['Was This Search Helpful?' Buttons](#)
- [Document Last Modified Date in Reverb](#)
- [Drop-down Expand/Collapse All Toggle Button](#)
- [Google Translate Button](#)
- [Customizable Header and Footer](#)
- [Custom TOC Menu Items](#)
- [Customizing a Bullet Icon using Font Awesome](#)
- [Using the url_maps.xml reference file](#)

WebWorks Reverb 2.0 has many of the features found in WebWorks Help, and goes one step further by providing high-performance load times combined with optimal handling of the end-user's device resolution. This means that users viewing this type of output on a mobile device will be able to view the files in a fast, light-weight manner that is optimal for lower resolutions and/or touch operated screens. Furthermore, users on a more traditional higher-resolution computer monitor will still have the same advantages offered by a complete online help system. In other words, Reverb adapts to the traits of the device that is doing the reading of the content. Additionally, the JavaScript code for WebWorks Reverb 2.0 has been written in an inline-safe manner, which makes the output even more secure on the web.

WebWorks Reverb 2.0 also offers a commenting and end-user feedback mechanism using the Disqus commenting and discussion platform. This platform is a leading solution on the Internet for comment handling and has no cost and provides automatic hosting of your end-users comments in a way that is transparent and effective.

Reverb also provides easy Google Analytics integration. Using Google Analytics in your web files will be very easy to configure with this format.

If you have end-users that require localized help files, WebWorks Reverb 2.0 offers a seamless integration with the Google Translate Element. This feature is very powerful and can be used instead of an expensive and burdensome set of translated files and help volumes, which requires a separate set of content for each language that you are supporting. Using WebWorks Reverb 2.0, you can now create instantaneously

localized help volumes without any translation requirements and only one set of help files. For more information see “Google Translate Button”.

Choosing a Skin

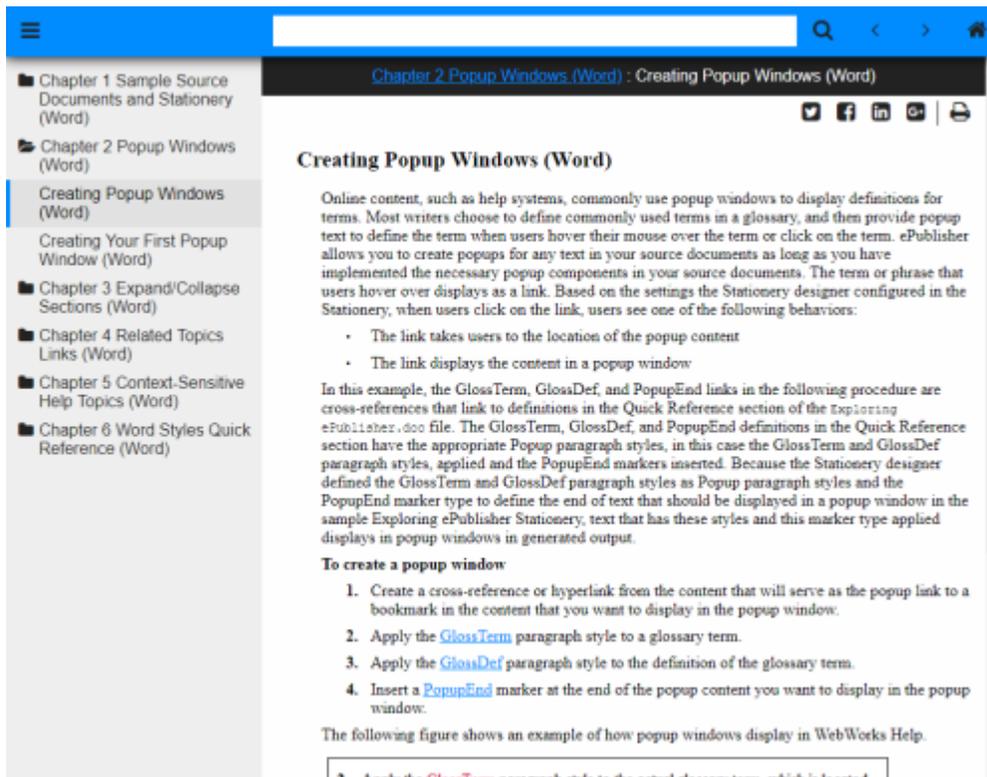
You can select from a number of predefined skins that are available for the WebWorks Reverb 2.0 output format. Each skin has been professionally designed with assistance from both graphic designers and web developers so that your users can get an ideal experience when browsing your documentation.

Note: If you plan on customizing the Reverb 2.0 toolbar or menu, we recommend not using the skins to simplify future upgrades.

You may select among the following types of skins:

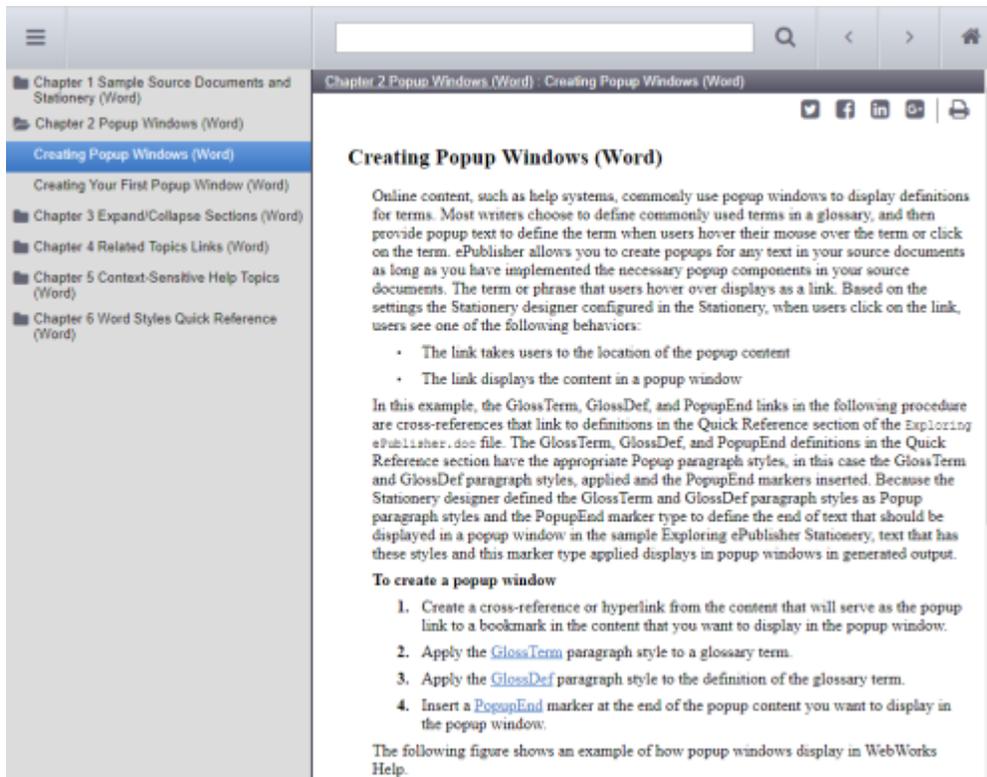
Neo

The default skin for WebWorks Reverb 2.0, Neo is the latest design for Reverb output and was designed with current web aesthetics in mind. It features a simplistic layout that looks great across many devices. This skin is very versatile and well suited for most purposes.



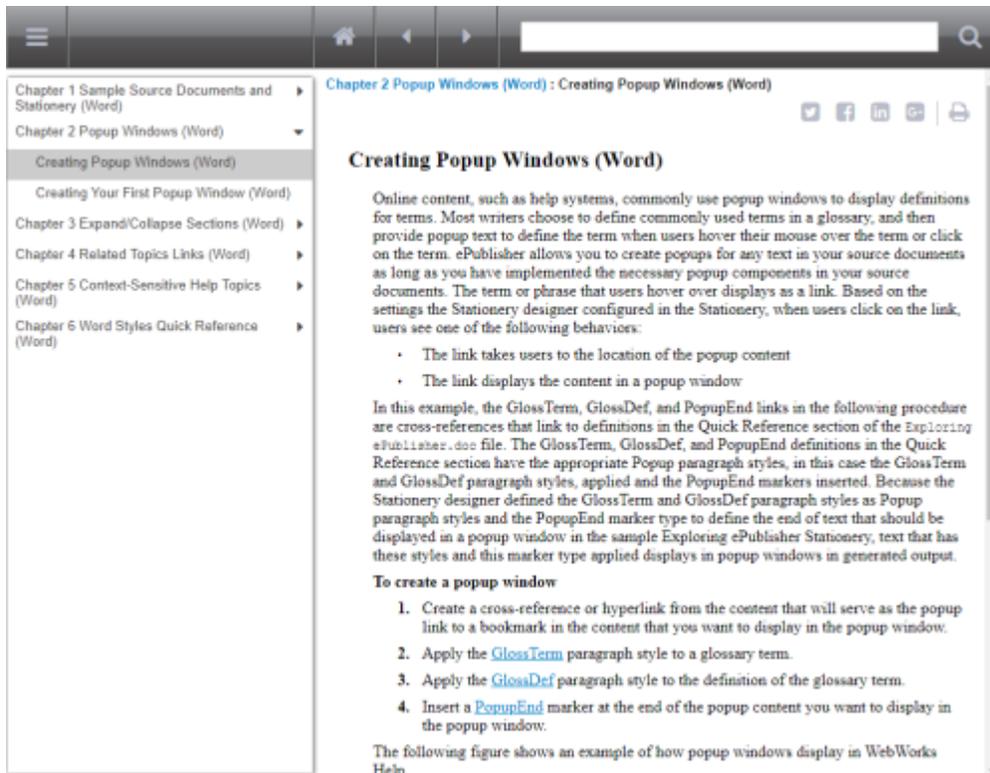
Classic

Based off of the original WebWorks Reverb skin, Classic makes a return with some modifications in WebWorks Reverb 2.0. This skin features a larger toolbar, and gradients across the layout to produce a traditional help look and feel. This skin is great for many purposes and offers existing users a similar skin if they were previously using the Classic skin with WebWorks Reverb.



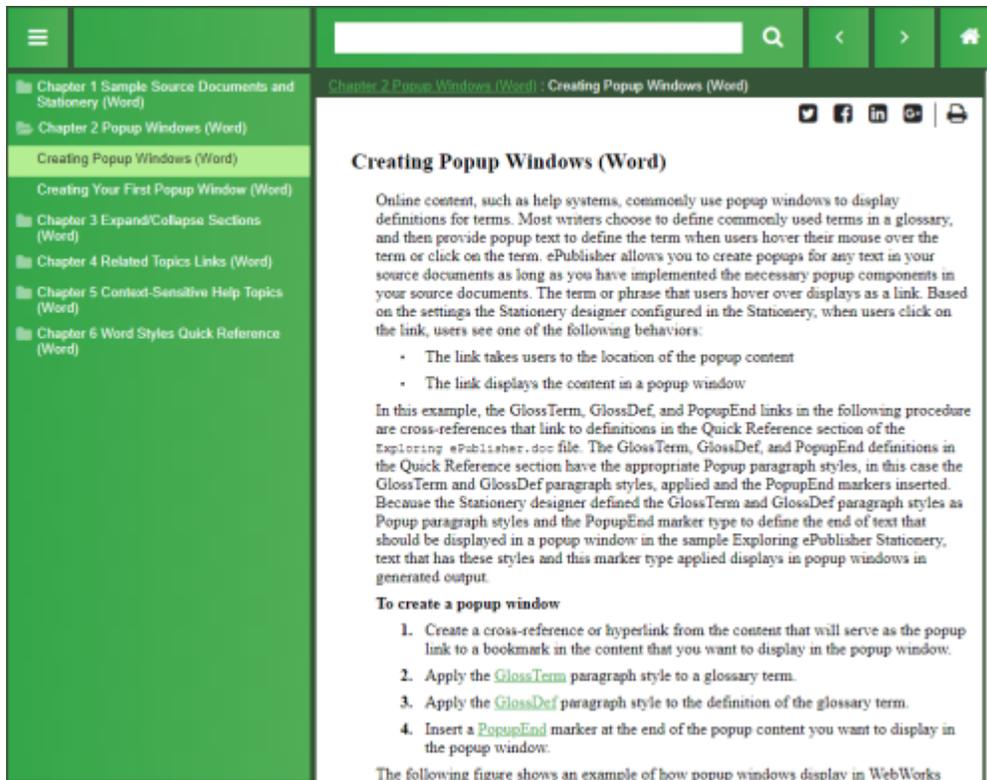
Corporate

The Corporate skin has been brought into the WebWorks Reverb 2.0 Format, with an updated yet familiar look to it. This skin has a polished look and would suit the needs of a high-profile technology company's help set.



Metro

The Metro skin has returned in WebWorks Reverb 2.0. This skin was designed in reference to the Windows 8.1 Metro interface design pattern. The Metro skin is useful for users wanting migrate from WebWorks Reverb and have a starting point that looks and feels like their previous Metro skin.



Social

The *Social* skin has been designed to provide a user experience most similar to that of social networking websites. It provides a familiar interface that is intuitive and casual enough to maximize your end-user participation.

Chapter 2 Popup Windows (Word) : Creating Popup Windows (Word)

Creating Popup Windows (Word)

Online content, such as help systems, commonly use popup windows to display definitions for terms. Most writers choose to define commonly used terms in a glossary, and then provide popup text to define the term when users hover their mouse over the term or click on the term. ePublisher allows you to create popups for any text in your source documents as long as you have implemented the necessary popup components in your source documents. The term or phrase that users hover over displays as a link. Based on the settings the Stationery designer configured in the Stationery, when users click on the link, users see one of the following behaviors:

- The link takes users to the location of the popup content
- The link displays the content in a popup window

In this example, the `GlossTerm`, `GlossDef`, and `PopupEnd` links in the following procedure are cross-references that link to definitions in the Quick Reference section of the `exploring ePublisher.doc` file. The `GlossTerm`, `GlossDef`, and `PopupEnd` definitions in the Quick Reference section have the appropriate `Popup` paragraph styles, in this case the `GlossTerm` and `GlossDef` paragraph styles, applied and the `PopupEnd` markers inserted. Because the Stationery designer defined the `GlossTerm` and `GlossDef` paragraph styles as `Popup` paragraph styles and the `PopupEnd` marker type to define the end of text that should be displayed in a popup window in the sample `Exploring ePublisher Stationery`, text that has these styles and this marker type applied displays in popup windows in generated output.

To create a popup window

1. Create a cross-reference or hyperlink from the content that will serve as the popup link to a bookmark in the content that you want to display in the popup window.
2. Apply the `GlossTerm` paragraph style to a glossary term.
3. Apply the `GlossDef` paragraph style to the definition of the glossary term.
4. Insert a `PopupEnd` marker at the end of the popup content you want to display in the popup window.

The following figure shows an example of how popup windows display in WebWorks Help.

Using SASS To Customize Reverb Interface

The WebWorks Reverb 2.0 Format uses a technology called SASS to present the user with a dynamic and responsive visual experience. To summarize, SASS is useful because it lets the designer make use of devices that are typically not available in the CSS language. SASS lets the user create and reuse variables across their style sheet, as well as create functions and mixins to enable efficiency and ease of access that other programming languages typically get to enjoy. Once a SASS design has been created, it is then processed and transcompiled to CSS for use in HTML layouts.

Because of the benefits of this, the WebWorks Reverb 2.0 design process is able to be simple and robust. With a large collection of intuitively-named variables, users are able to change settings like fonts and colors in one spot and watch it propagate throughout the layout.

For those interested, SASS is well documented and WebWorks' implementation of this technology conforms with the standards implemented by the creator. Below are a list of resources that may be useful in priming the user to interacting with SASS for the first time.

SASS Basic Guide: <https://sass-lang.com/guide>

SASS Reference: https://sass-lang.com/documentation/file.SASS_REFERENCE.html

CSS Tutorial: <https://www.w3schools.com/Css/>

Using Custom SASS files in Reverb Projects

The WebWorks Reverb 2.0 Format now allows you to use your own custom SASS files for your projects which allows for even more customization To do this simply add your SASS file as a Format or Target override.

Previewing Reverb Output

You can view the generated files directly in your browser, however, the social-media functionality will not be present when viewing WebWorks Reverb 2.0 files in this manner. In order to preview the social-media functionality, you will need to view the files from a web server. With ePublisher, you can preview your WebWorks Reverb 2.0 files through a web server without having to configure a separate web server.

To preview the output, simply select the top level group in the **Document Manager**. This will display an entry in the **Output Explorer** called: `View Output`. Double-click on this entry and your help system will be opened in your default browser connected to a web server built into ePublisher.

To fully preview WebWorks Reverb 2.0 using the built-in web server

1. Open your project and make sure that your WebWorks Reverb 2.0 target is active and fully generated.
2. In the **Document Manager** select the top level group.
3. In the **Output Explorer** underneath **Merge Output**, double click **View Output**.
4. A browser window will be opened using your default browser and a URL to a localhost web address will be displayed. You can now browse the entire help volume as if it were deployed on a dedicated web server.

Delivering Reverb Output

When you generate output for your project, ePublisher creates the `Output` folder in your project folder, and then creates a folder in the `Output` folder for each generated target. Then, ePublisher creates a folder named for the project itself in the target folder. For example, when you generate output, ePublisher creates the following folder structure:

```
<Project Area>\<Project Name>\Output\<Target Name>\<Project Name>
```

In this folder structure, `<Project Area>` is the name of the ePublisher component you are using, such as `ePublisher Express`, `<Project Name>` is the name of your ePublisher project, and `<Target Name>` is the name of your ePublisher target, such as `WebWorks Reverb`. To deliver your WebWorks Reverb generated output, you need to deliver all the files and subfolders in the `<Target Name>\<Project Name>` folder.

The `<Target Name>\<Project Name>` folder contains the entry-point file, `index.html` by default, which establishes the help set appearance. When the user opens the entry-point file, the browser uses all the files in the `<Target Name>\<Project Name>` folder to display the help, including all the topic files, generated `.css` files, `.pdf` files, images, and WebWorks Reverb components.

Top-Level Groups in Reverb

Top-Level Groups are used for many organizational and functional purposes in WebWorks Reverb 2.0. Once a help set is generated as a WebWorks Reverb 2.0 Output, Top-Level Groups are converted into a hierarchical structure to form the `Parcels` that make up the Table of Contents, and the structure in the file system. These `Parcels` can then be selected by the user when using Scoped Search to filter search results, and also using the URL method to filter a help set into a subset that contains a specific section of the entire set.

Searching Output

Use the following guide to assist users in finding terms in your help system:

Boolean

All search words and phrases have an implicit AND

Word Search

For example, searching: `eggs bacon` returns all documents containing “eggs” and “bacon”

Phrase Search

For example, searching `"Good Morning"` returns all documents containing “good” and “morning”. However, the two words will not necessarily be adjacent. In other words, at this time, phrase search returns the same results as a multi-word search.

Using Baggage Files

Baggage files are `PDF`, `HTML` and `ZIP` files that are not part of the ePublisher source files to be converted. ePublisher makes this determination by examining the file extension. If the file extension matches one of the following listed below then it's considered a baggage file.

`.pdf`

`.html`

`.htm`

`.shtml`

`.shtm`

`.xhtml`

`.xhtm`

`.zip`

In order to a file to be processed as a Baggage file ePublisher must know about it. This can be achieved in one of two ways.

- (1) Any link from a source file being converted by ePublisher.
- (2) Entry in the `baggage file info list`. For more information see “Indexing Baggage Files and External URLs”.

Baggage Files will be packaged within your Output folder.

Note: In **Target Settings > Links > Baggage File Target**, specify whether you want the link to open in the same browser window or in a different window.

Note: Set **Target Settings > File Processing > Insert Mark of the Web (MOTW) = Disabled**. Otherwise, the link will fail in Internet Explorer on the local file system. It will work in other browsers. It will work in IE on a web server.

Indexing Baggage Files and External URLs

With WebWorks Reverb 2.0, files can be indexed to produce as search results with the user's help set. An indexable **Baggage File** in this context is any PDF or HTML file that is linked from a source document that will be included in the generated output for producing useful search results. For more detailed information on baggage files, see "Targets".

Note: In order to determine what baggage files are indexed, ePublisher examines the file extension and if it matches one on the following then it will be indexed.

.pdf

.html

.htm

.shtml

.shtm

.xhtml

.xhtm

Baggage files are indexed in the same way that source documents are. Indexable baggage files will be indexed as long as the **Index baggage files** Target Setting is **Enabled**. External URLs will be downloaded & indexed as long as the **Index external links** Target Setting is **Enabled**.

Using Tidy for Indexing HTML Pages

In order to index an HTML baggage file, Reverb creates an XHTML copy of the file using Tidy (tool for cleaning up HTML files) to get a valid XML file that ePublisher can read. As useful as Tidy is, there may be times where it does not recognize a tag or generates something improperly. Tidy is configurable and can be adjusted to convert the HTML in the proper way.

When Tidy does not recognize a tag in an HTML file, an error like the following is produced:

```
line 33 column 3 - Error: <not_recognized_tag> is not recognized!
```

This error means that Tidy wasn't able to generate an XHTML copy of the HTML file, and therefore ePublisher won't be able to index it as a baggage file. With the right adjustments, this can be fixed.

Configuring Tidy To Recognize New Tags

1. Go to your Tidy directory under the installation directory in your local computer: `...\WebWorks\epublisher\<VERSION>\Helpers\tidy\`
2. Create a Format override of this helper. To do this: in the sub-folder of your project called: `Formats`, where the Format overrides live, create a new folder called `Helpers` and copy the entire folder called `tidy` (from step 1) to this new folder.
3. In the newly created `tidy` folder, open your `config.txt` file.
4. Depending on the kind of tag you want to add, you'll have to uncomment line 8 or 10, or maybe both in the `config.txt` file.
5. Substitute the placeholder we put there and after the colon, with your new tag name (for example: `not_recognized_tag`).
6. Save and close the file.

To know more about how to customize Tidy go to <https://www.w3.org/People/Raggett/tidy/>.

Assigning Relevance Weight to Your Source Documents Styles

Search results are displayed in the Search tab when a user types a word to search for. The search results are sorted by a relevancy ranking, which, in the case of source documents, is calculated based on the **Search relevance weight** option defined in your **Paragraph** and **Marker Styles**. By default, WebWorks Reverb 2.0 assigns relevance weight of 1 to all styles.

To Modify the Relevancy Ranking in Source Documents for Search Results

1. Open your project with ePublisher Designer.
2. Scan the document, to pull all styles into the Style Designer.
3. Open the **Style Designer** (**F10** or **View > Style Designer**).
4. Select the style you want to assign a weight to (either in **Paragraph Styles** or **Marker Styles**).
5. Open the **Options** window.
6. Change the **Value** of the **Search relevance weight** option to a decimal number you determine or you can just ignore it (which is going to be 0), meaning that the style is not going to be shown in your results.

Assigning Relevance Weight to Your HTML and PDF Baggage Files

The search results are sorted by relevancy ranking, which, in case of HTML baggage files, is calculated based on the scoring preference defined for the HTML tags in the `search_settings.xml` file. By default, WebWorks Reverb 2.0 assigns relevancy rankings based on where in a topic a particular item is found.

To Modify the Relevancy Ranking in Baggage Files for Search Results

1. Open your project with ePublisher Designer.
2. ***If you want to override the relevancy ranking for all WebWorks Reverb 2.0 targets***, create the `Formats\WebWorks Reverb 2.0\Transforms` folder in your `projectname` folder, where `projectname` is the name of your ePublisher project.
3. ***If you want to override the relevancy ranking for one WebWorks Reverb 2.0 target***, create the `Targets\WebWorks Reverb 2.0\Transforms` folder in your `projectname` folder, where `projectname` is the name of your ePublisher project.
4. Create a customization of your `search_settings.xml` file.
5. You'll see the following block of code:

```
<Settings version="1.0" xmlns="urn:WebWorks-Settings-Schema">
<ScoringPrefs default-weight="0.05" pdf-weight="0.05">
  <meta name="keywords" weight="1.0"/>
  <meta name="description" weight="1.0"/>
  <meta name="summary" weight="1.0"/>
  <title weight="1.0"/>
  <div class="myclass" weight="0.05"/>
  <div weight="0.05"/>
  <h1 weight="0.1"/>
  <h2 weight="0.1"/>
  <caption weight="0.1"/>
  <h3 weight="0.1"/>
  <th weight="0.1"/>
  <h4 weight="0.1"/>
  <h5 weight="0.1"/>
  <h6 weight="0.1"/>
  <h7 weight="0.1"/>
  <p weight="0.05"/>
</ScoringPrefs>
</Settings>
```

6. Modify the `weight` attributes for any tags, such as `h1` and `h2`, you want to change. You can also specify additional tags with or without class attributes to further refine weights for your HTML baggage files. You may use decimal values to modify the `weight` attribute value.

Note: If you wish to set a default weight to tags that are not defined in this file simply update the `default-weight` attribute value.

Note: You can change the default weight for all of the text in a PDF file by changing the `pdf-weight` attribute value.

7. Save and close the `search_settings.xml` file.
8. Regenerate your project to review the changes.

Search Highlighting in Baggage Files

When you click on a result in your Search Results, you'll open the associated source document or baggage file. If what you are clicking is a baggage file and you want to get the highlighting feature in your baggage file, you'll have to copy next to your file and then reference the `reverb-search.js` script in the `<head>` tag of your HTML file. The `reverb-search.js` file lives in the installation directory at `...\WebWorks\ePublisher\<VERSION>\Formats\WebWorks Reverb 2.0\API\reverb-search.js`.

To Reference the `reverb-search.js` File From Within Your HTML File (After Copying the Script Next to Your HTML File)

1. Open your HTML document.
2. Locate the `<head>` tag.
3. Create the following line inside the `<head>` tag, pointing to the script you just copied:

```
<script type="text/javascript" src="../../../reverb-search.js"></script>
```
4. Save your HTML file.
5. You can either Enable in your **Target Settings** under **Baggage Files** the **Copy baggage file dependents** (this will copy the script to the Output folder, see “Copy baggage file dependents”), or you can manually copy the file to the Output directory, next to your baggage file.

Searching with URL Method

Search actions can be initiated via URL. For example you can go to the link here:

<http://www.webworks.com/Documentation/Reverb/index.html#search/ePublisher>

When clicked, the link above searches for the term “ePublisher”. This can be changed to any search term, which will produce results accordingly.

TOC or Index with URL Method

You can toggle between the Table of Contents or the Index in the Reverb 2.0 output. For example, you can go to:

<http://www.webworks.com/Documentation/Reverb/index.html#toc/>

http://www.webworks.com/Documentation/DITA_1.2_Specification/index.html#index/

Clicking on these links will bring up that part of the help as opposed to the regular view.

Launching Context Sensitive Help for WebWorks Reverb 2.0

WebWorks Reverb 2.0 allows you to open a specific topic. Context sensitive help is a way to identify a topic by using a standard location combined with a topic identifier (topic alias marker). For example, you can go to:

<http://www.webworks.com/Documentation/Reverb/index.html#context/welcome/whatsnew>

You can also use just the topic name without the group context.

<http://www.webworks.com/Documentation/Reverb/index.html#context/whatsnew>

End-user requirements in WebWorks Reverb 2.0

End-users must have JavaScript enabled in their browser. If not, the user will be prompted with a localized message asking them to enable JavaScript to view the content. Local deployments must have DOM storage enabled in the browser. Web server deployments do not require DOM storage to be enabled. However, some features will be disabled.

End-users must also consume a WebWorks Reverb 2.0 help set with a browser that is not prior to Internet Explorer 11. Modern browsers are recommended for the best possible experience. If an end-user tries to access a WebWorks Reverb 2.0 help set with an unsupported browser, they will be presented with a message instructing them that the browser they are using is not supported.

It is possible to customize these messages by overriding the proper Page Templates (`.asp`).

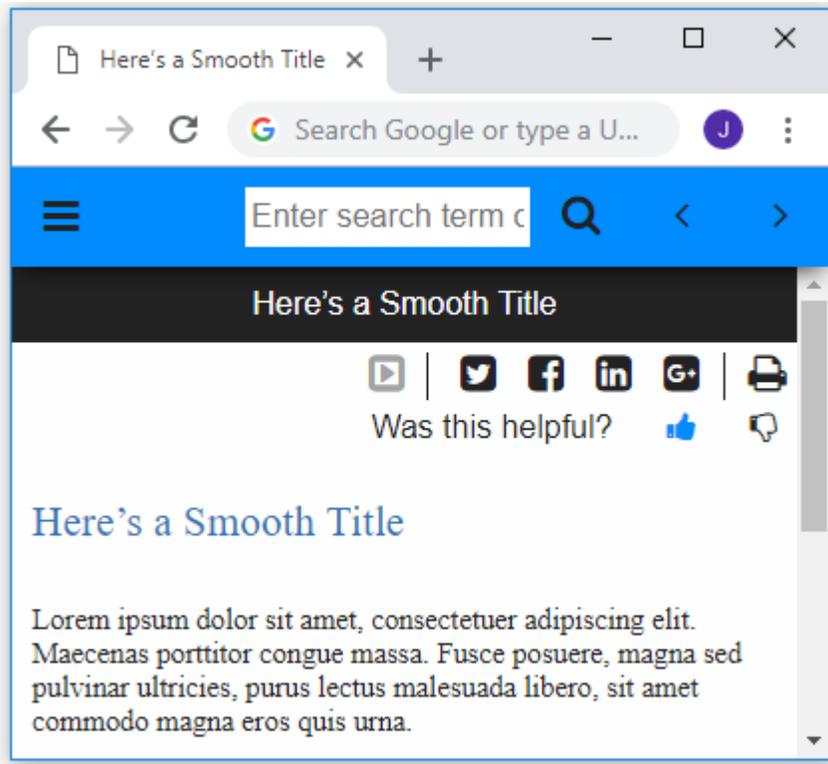
Analytics Event Tracking in Reverb 2.0

In WebWorks Reverb 2.0, analytics has been expanded to anonymously track several types of events that better allow you to understand the performance of your content with your end-users. The updated analytics features include:

- All page views are tracked with easy to read page names.
- New categories and labels have been introduced to highlight high performing and low performing pages in your reports. For more information, see “‘Was This Helpful?’ Buttons”.
- All Reverb interface components, such as toolbar buttons and menu items have their interactions tracked as events.
- All search queries, inputs, cancellations, and search result views are tracked and recorded.
- Search page results including the **Was This Helpful?** buttons are tracked for easy reporting. For more information, see “‘Was This Search Helpful?’ Buttons”.

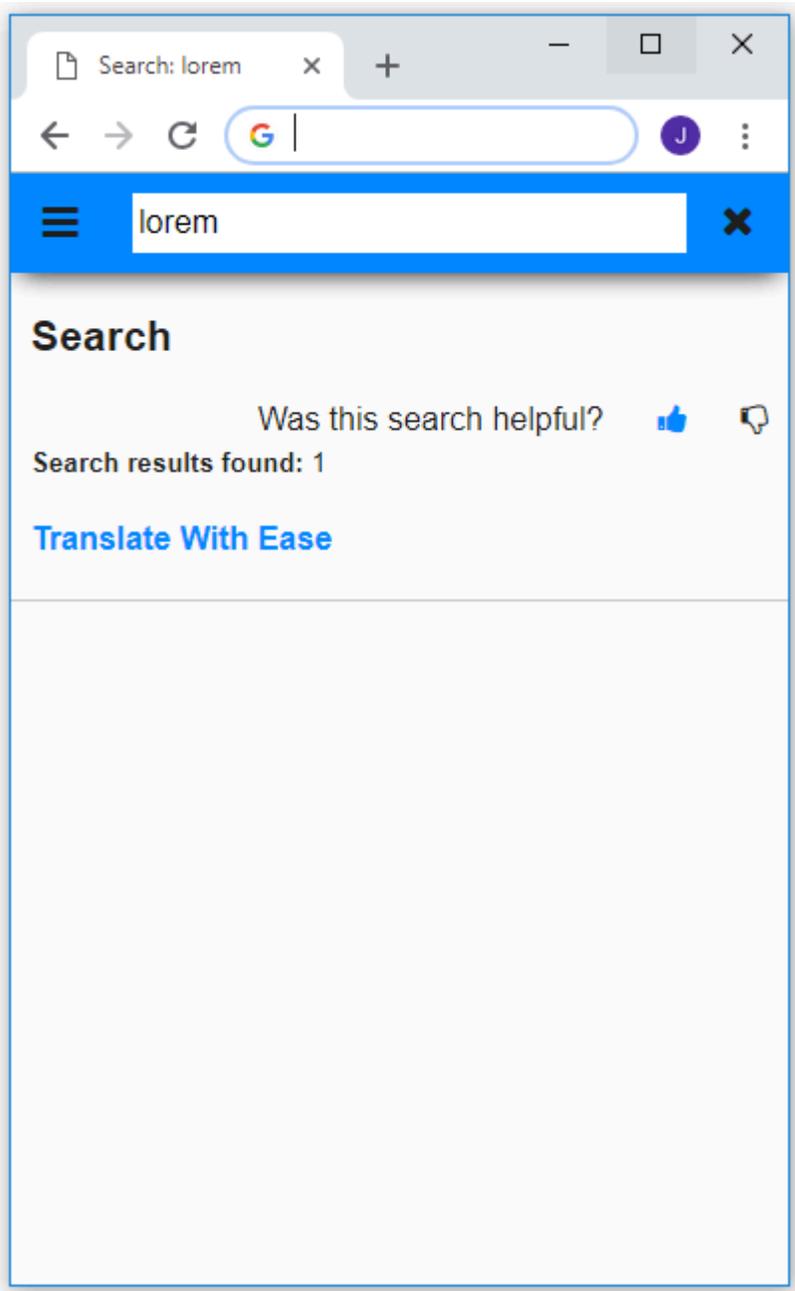
'Was This Helpful?' Buttons

The **Was This Helpful?** buttons feature allows you to provide your end-users with the ability to give anonymous feedback about their experience. The **Was This Helpful?** buttons feature can be used to record feedback on the current page being viewed. A Google Tracking ID is required for these buttons to record analytic events.



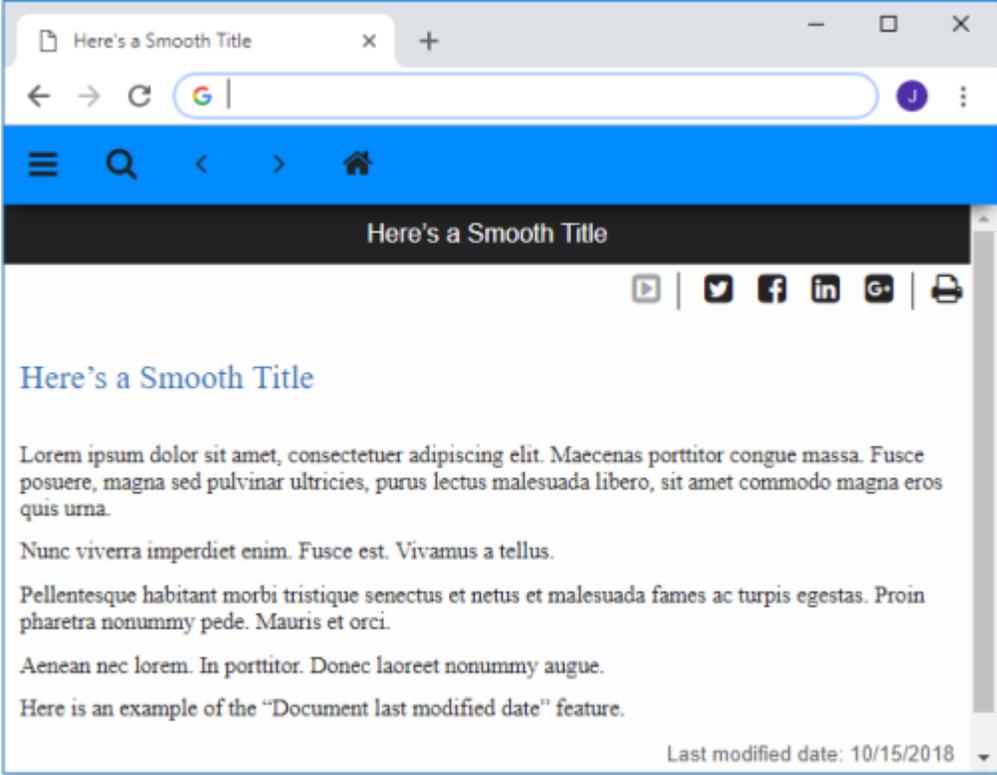
‘Was This Search Helpful?’ Buttons

The **Was This Search Helpful?** buttons feature allows you to provide your end-users with the ability to give anonymous feedback about their search results. The **Was This Search Helpful?** buttons feature can be used to record feedback and search queries of the search results page. A Google Tracking ID is required for these buttons to record analytic events.



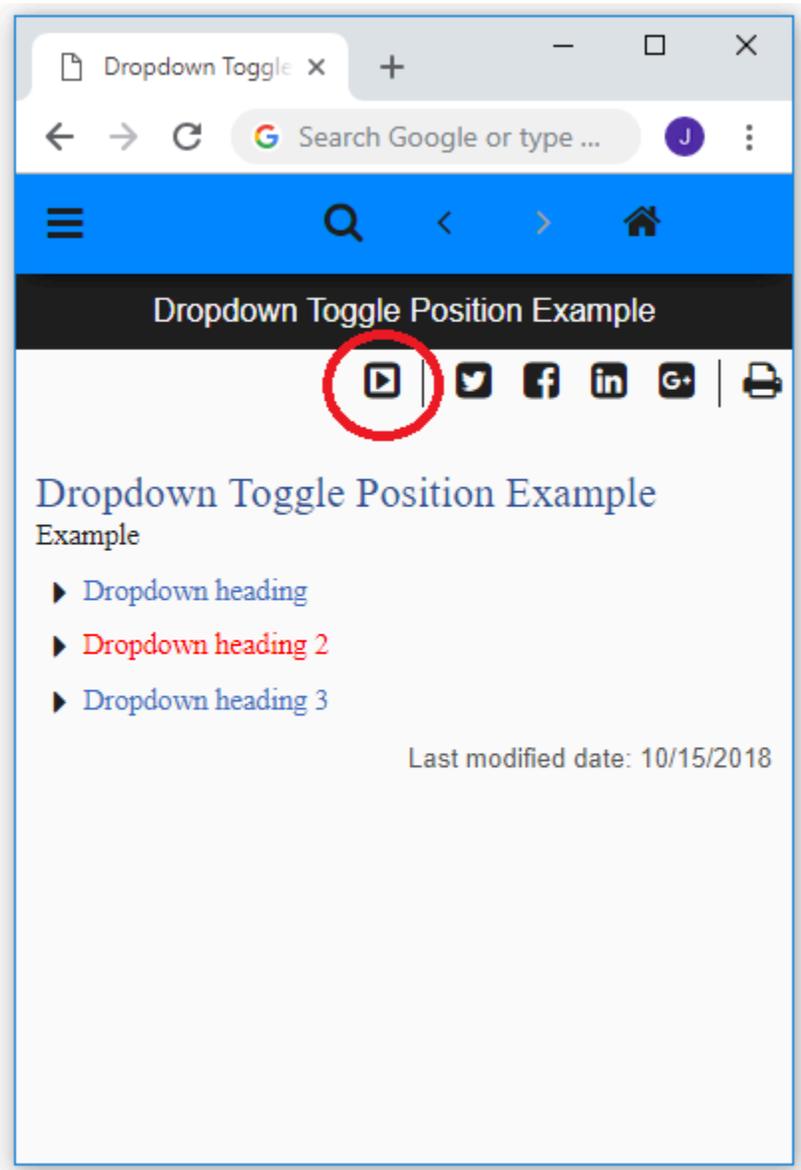
Document Last Modified Date in Reverb

In WebWorks Reverb 2.0, there is a target setting that allows you to display the **Last modified date** of each document in your generated output pages. If you enable this target setting, your end-users will be able to see the date of that last time the document that generated a particular page was edited. If you examine the image below, you can see the date string in a light colored gray.



Drop-down Expand/Collapse All Toggle Button

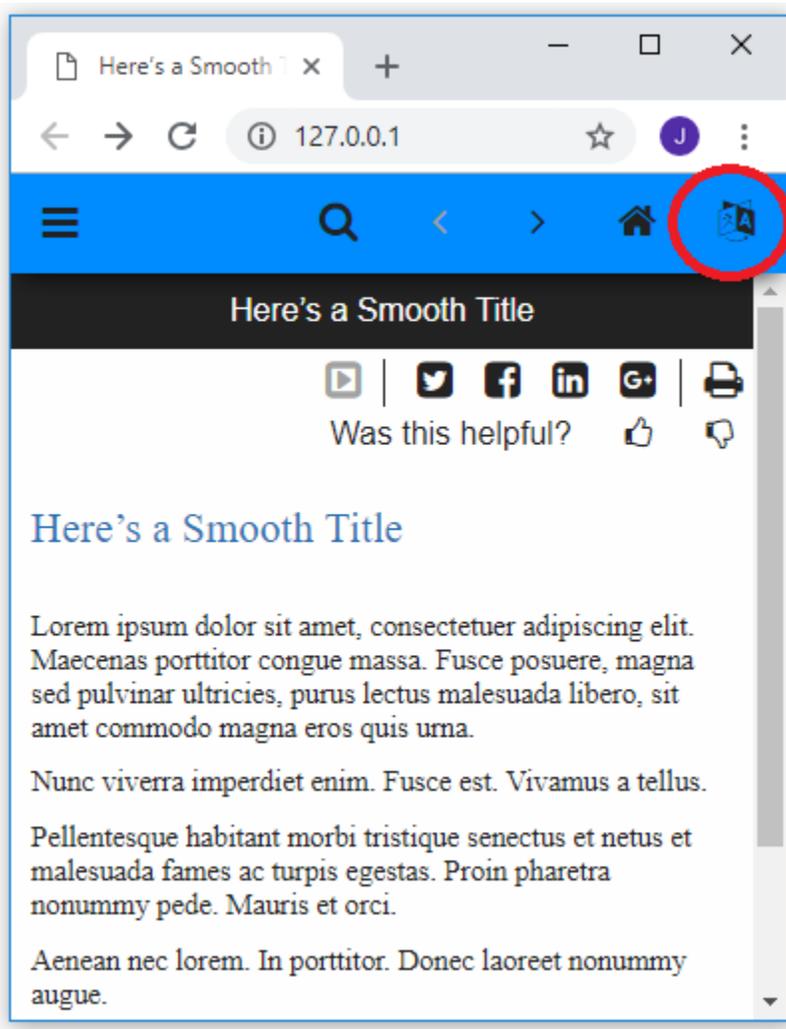
In WebWorks Reverb 2.0, you can centrally control all of the drop-down paragraphs on a page using the **Dropdown Expand/Collapse Toggle** button. It is enabled by default, but can be disabled in the target settings. The image below highlights the toggle button.



Google Translate Button

In WebWorks Reverb 2.0, you can use the target settings to enable a Google Translate button to be part of your toolbar. When your end-users select this toolbar button, all of the text in the content area will be translated using Google's Translate web service. The navigation menus however are not translated, just the content area. This feature will only be displayed when the output is deployed to a web server, which is a requirement of the Google Translate web service.

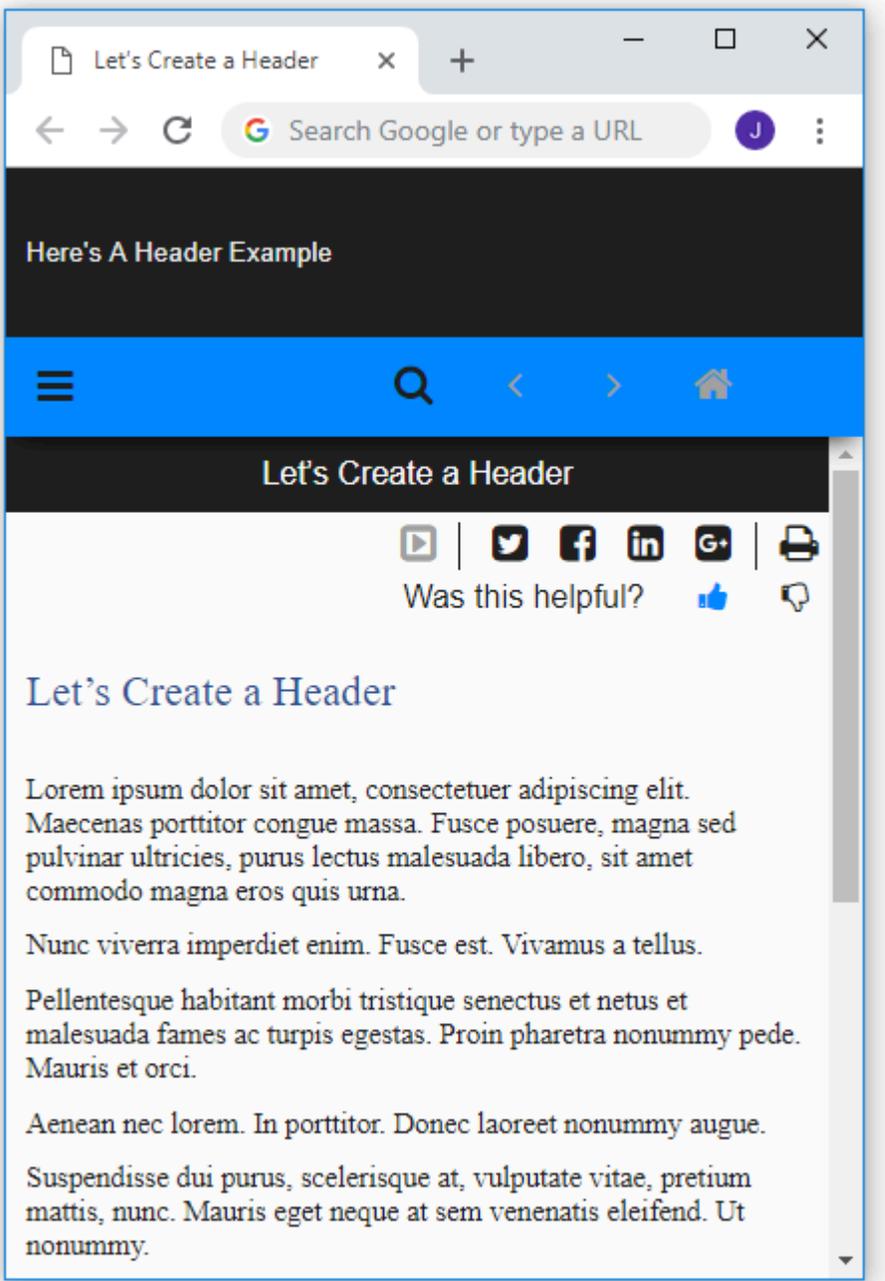
Note: As an added benefit, Reverb 2.0 is smart enough to know when to display the Google Translate button and when to suppress it. By not displaying the Google Translate button when loading directly from a file system (i.e. as packaged help), you will need only one configuration (target) to support both your web deployment and your packaged help deployments.



Customizable Header and Footer

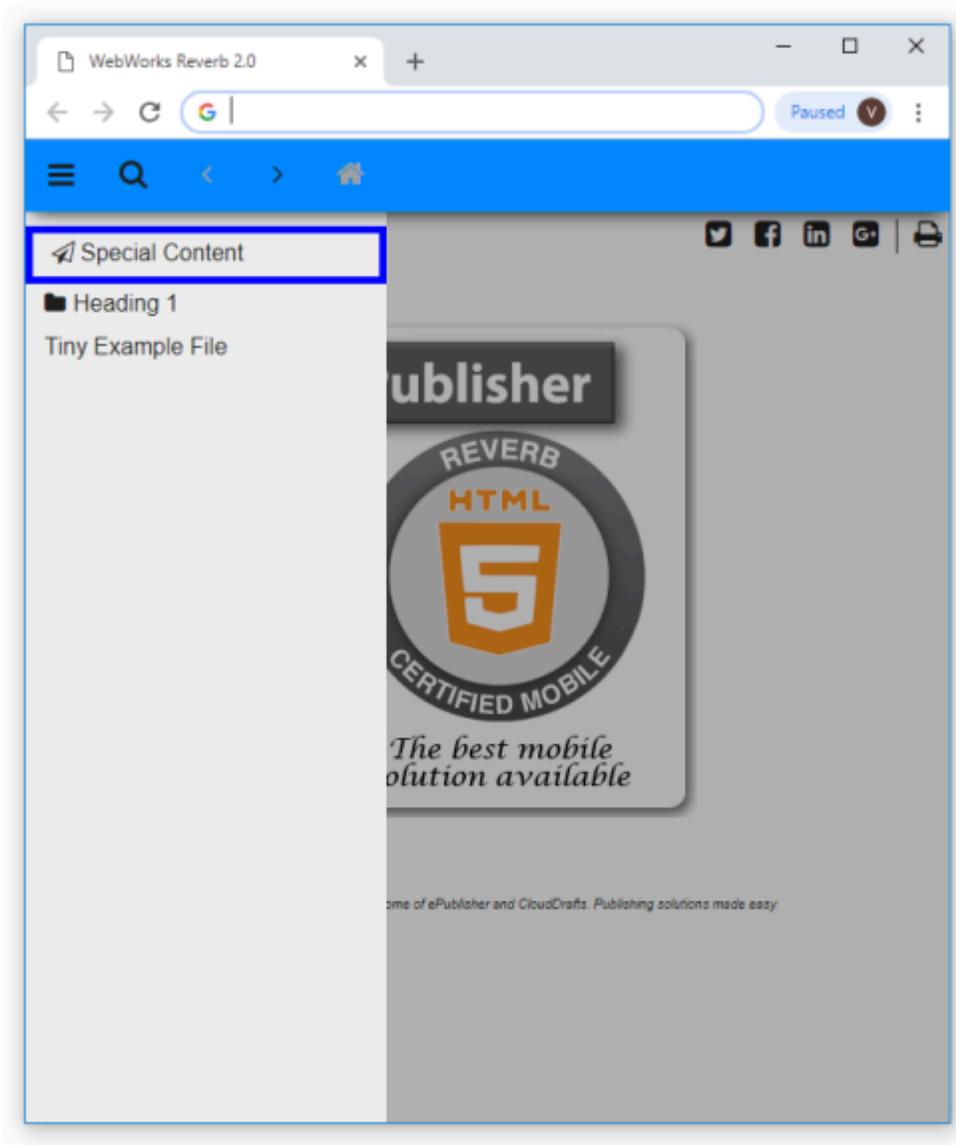
In WebWorks Reverb 2.0, using target settings, a customizable Header and Footer can be enabled. Each of these parts uses its own ASP file that can be customized. In addition, you can specify that company information be displayed or another logo image of your choosing.

The following image demonstrates a custom header above the Reverb 2.0 toolbar.



Custom TOC Menu Items

In WebWorks Reverb 2.0, you can customize specific TOC (table of contents) Menu items based the target paragraph that the item links to. For example, if a TOC item links to a paragraph that contains a marker assigned to the ePublisher marker type called: TOC Entry class, then that TOC item will inherit the value of the marker as a CSS class name. Using the class name, you can completely change the appearance of the TOC item in the Menu. See the image below for an example of a TOC Menu item that has a different icon and blue borders.



Customizing a Bullet Icon using Font Awesome

It is possible to use a Font Awesome icon for the bullet in a list. In order to do this, take the following steps:

1. In the Style Designer, create a new Character Style that will represent the Font Awesome bullet.
2. In the Character Style Options, under Additional CSS Classes, add fa, a whitespace, and the class of the icon you want to use. (For example: fa fa-warning) For a reference of all the icons available, refer to the Font Awesome Cheatsheet at:

<https://fontawesome.com/v5/cheatsheet/free>

3. Select the Paragraph Style that is to have the bullet added.
4. In the Bullet Properties area for the Paragraph Style, under the Property called Character Style, select the name of the Character Style that has the Font Awesome icon applied to it.
5. (Optional) If the Paragraph Style was initially an unordered list in the source document, it is also necessary to add a single whitespace to the Bullet Property called Text so the default bullet does not emit from the source document.

Using the url_maps.xml reference file

In the Output Explorer, the **url_maps.xml** file can be used to view the links available in an output helpset. When opened in a text editor, this file contains all of the available links in the output. This can be useful for web designers or anyone needing a reference to use these links externally, such as on a website or a custom page. The file can also be used for sitemap generation if you know how to work with XML.

The **TopicMap** element contains all of the Topic Alias Links in the output. The **PageMap** element contains all of the Page URL links available in the output.

PDF - XSL-FO

[Why use PDF - XSL-FO output format?](#)

[PDF-XSL-FO Page Regions](#)

[PDF XSL-FO Font Inclusions](#)

With ePublisher, you can quickly and easily create PDF output from Adobe FrameMaker, Microsoft Word, and DITA-XML source documents. Based on your requirements, you can either create PDF output so that it mirrors the styling and layout of your source document's pre-configured settings (using the *PDF* output format), or you can choose to manage all these settings through ePublisher using the *PDF - XSL-FO* output format.

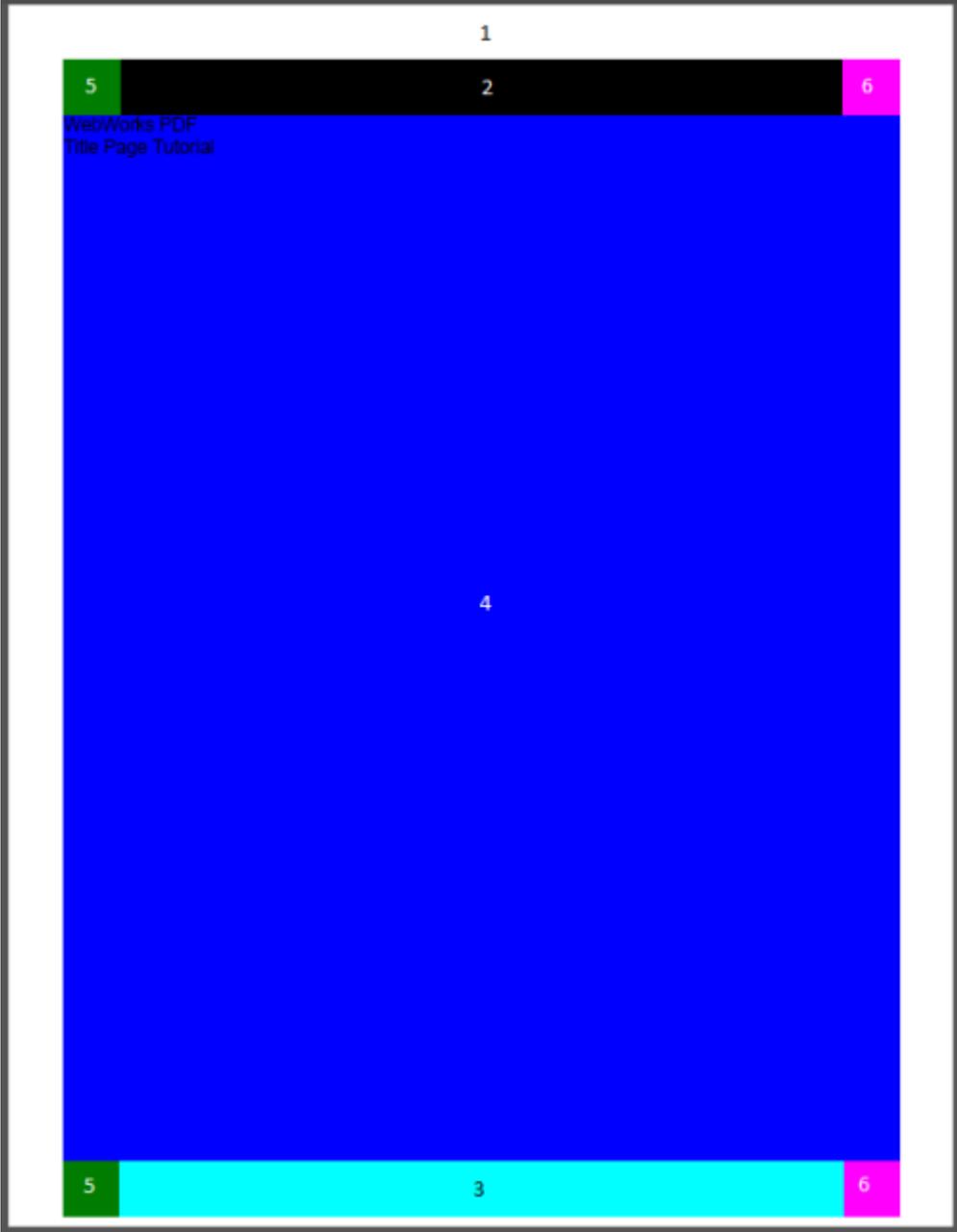
Why use PDF - XSL-FO output format?

If you want to keep your original document's style and layout settings, then you will want to configure ePublisher to use the *PDF* output format. However, if you want to directly control your document's style and layout settings, then you will need to configure ePublisher to use the *PDF - XSL-FO* output format.

The PDF - XSL-FO format creates a PDF file for each of your source files, a single PDF file containing all source files within a group, or both. PDF - XSL-FO is recommended for producing a PDF output file that conforms to the settings you configure in ePublisher. These settings can be controlled independent of your source file's authoring environment or they can be used to work in conjunction with your source file's paragraph and character level settings. This format allows you to create and deliver PDF files that conform to the style and layout settings that you can centrally control in ePublisher. This format may be especially useful when mixing documents from different input formats or different layouts and/or styles.

PDF-XSL-FO Page Regions

When creating a PDF-XSL-FO output, it can be useful to understand the regions of the page when styling a PDF document.



The numbered regions corresponding with the list items below, which can be modified in the Page Styles section of the Style Designer.

6. Master Page Margins (Top, Left, Right, Bottom).
7. Master Page: Before Region - This is the region where a header will be rendered
8. Master Page: After Region - This is the region where a footer will be rendered
9. Master Page: Body Region - This is where the content of the page is rendered
10. Master Page: Start Region
11. Master Page: End Region

Note: It should be noted that the Start and End regions are actually rectangular regions that span from the top to the bottom of the page. In this image the Body Region is covering the middle of the Start and End regions, giving the appearance of two small squares in the corners of the document.

The **Extent** property can be used to set the sizing of these different regions, with the exceptions of the Master Page margins and the Body Region.

PDF XSL-FO Font Inclusions

To ensure non-standard fonts are included into your generated PDF, you may need to modify the Apache FOP configuration files, `apache-fop-2.6.xconf`.

To do this, you must either create a Format or Target Override, Create the override in one of these places, for more information on overrides, Depending on how you prefer to set up your project, you will create one of the following:

Per target:

```
Targets\<target name>\Helpers\apache-fop-2.6.xconf Targets\<target name>\Helpers  
\apache-fop-2.6.xconf
```

Per format:

```
Formats\<format name>\Helpers\apache-fop-2.6.xconf Formats\<format name>\Helpers  
\apache-fop-2.6.xconf
```

Project-wide:

```
Formats\Helpers\apache-fop-2.6.xconf Formats\Helpers\apache-fop-2.6.xconf
```

Once the override has been created, open the file in a text editor of your choosing, and you will see the following markup:

```
<font>  
  <!-- Example -->  
  <!--  
  <font kerning="yes" embed-url="file:///C:/Windows/Fonts/REFSAN.TTF">  
    <font-triplet name="Microsoft Sans Serif" style="normal" weight="normal" />  
  </font>  
  <font kerning="yes" embed-url="file:///C:/Windows/Fonts/REFSAN.TTF">  
    <font-triplet name="Microsoft Sans Serif" style="italic" weight="normal" />  
  </font>  
-->  
<!-- automatically detect operating system installed fonts -->  
<auto-detect/>  
</font>
```

Modify the lines in between the comments to reflect the font you are wanting and the directory path in which it is located. Save this file, and now you can add them by using ePublisher Pro.

To add fonts in ePublisher Pro

1. In ePublisher Pro, select the **View -> Style Designer** menu
2. With the **Paragraph Styles** visible, select the **[Prototype]** style

3. On the **Properties** tab, select **Font** then click the icon for **Family**
4. If your fonts are not listed, then manually add the names of your desired font using the **Custom Font Family** text box
5. Select **OK**

Dynamic HTML

Dynamic HTML Output Viewer
Delivering Dynamic HTML

There are several HTML-related standards, such as HTML 3.2, HTML 4, and XHTML 1.0. HTML 4 is also referred to as Dynamic HTML (DHTML). In addition, there are multiple browsers and device types that can display content based on these standards. With all these variables, getting the content you need formatted the way you need it for your specific environment and requirements is critical. ePublisher provides the Dynamic HTML output format to allow you to generate the HTML-based output you need. You can also customize this formatting to meet your requirements.

You can use the Dynamic HTML format to produce XHTML output that conforms to XHTML 1.0 standards and uses cascading style sheets that conform to the CSS1 standard. XHTML became a W3C recommendation in 2000. Dynamic HTML is recommended to produce output that you will publish on a Web server and provide to users running a Web browser, such as Firefox or Internet Explorer. You can also customize the Dynamic HTML output format to create a powerful, full-featured web site.

As the standards evolve, browsers and device platforms adjust to support the newer standards. With mobile devices, additional platforms and browsers have been introduced, which can complicate the decision about which standards your content can use.

The Dynamic HTML output format allow you to generate HTML content to integrate into your Web site. You can also create content for HTML-based release notes and content for hand-held devices, such as PDAs. These output formats provide the flexibility and control you need, with the ability to add a basic table of contents, index, and browse navigation. You decide whether to use all the aspects of DHTML and XHTML in the Dynamic HTML output format, or to use simplified HTML to support a wide range of browsers and platforms, including mobile devices and PDAs.

The Dynamic HTML output format produces HTML content that conforms to the HTML 4 and XHTML 1.0 standards, and uses cascading style sheets that conform to the CSS1 standard. Dynamic HTML (DHTML) is a collection of technologies developed to make HTML more dynamic and interactive. DHTML uses the following technologies to give the content developer control over the appearance and behavior of HTML elements in a browser window:

Static markup language (HTML 4)

HTML 4 extends HTML with mechanisms for style sheets, scripting, frames, embedding objects, improved support for right-to-left and mixed-direction text, richer tables, enhancements to forms, and improved accessibility for people with disabilities.

Presentation definition language (cascading style sheets)

Cascading style sheets (CSS) provide style definitions, such as fonts, colors, spacing, and positioning to HTML documents.

Client-side scripting language, such as JavaScript

JavaScript and other scripting languages provide compact, object-based scripting support for developing client and server Web applications.

Document Object Model

The Document Object Model provides a standard that allows programs and scripts to dynamically access, process, and modify the content of a page.

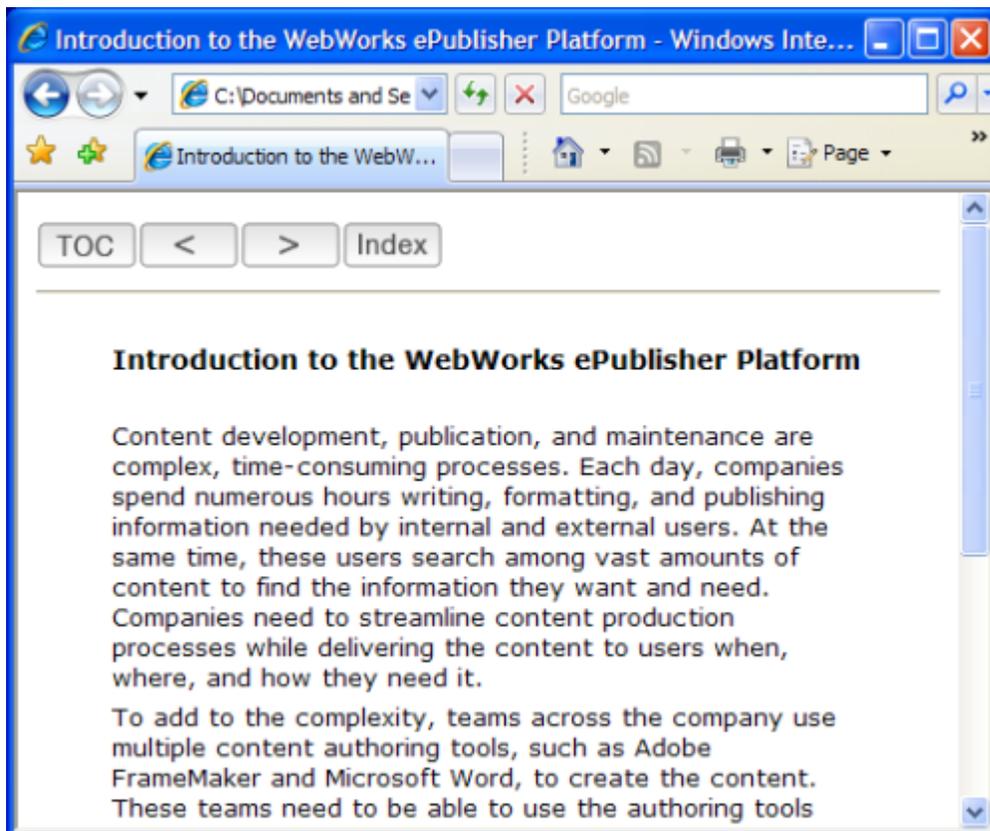
XHTML is an abbreviation for Extensible Hypertext Markup Language. XHTML 1.0 is similar to HTML 4, with tagging rules that conform to the requirements of XML. If you modify the page templates or styles in a Dynamic HTML project, make sure your changes conform to the XHTML requirements for future maintenance. However, as long as you create valid HTML, most current browsers can correctly display your output.

To determine whether the Dynamic HTML output format is what you need, review the following considerations:

- If all your users have current browsers and their viewing environment is not restricted, use the Dynamic HTML output format.
- If all your users have current browsers and you want to provide enhanced navigation controls, such as an expand/collapse table of contents, full text search, and related topics buttons, consider using the WebWorks Help or WebWorks Reverb output format.

Dynamic HTML Output Viewer

Dynamic HTML does not provide a multi-pane viewer. This output format is displayed in a browser. By default, ePublisher adds a navigation bar at the top of the files it creates. ePublisher also creates a table of contents page and an index page.



You can specify whether to include the navigation bar at the top or the bottom of the page. You can also add company information to the page, and define the table of contents and index pages.

Delivering Dynamic HTML

When you generate output for your project, ePublisher creates the `Output` folder in your project folder, and then creates a folder in the `Output` folder for each generated target. Then, ePublisher creates a folder named for the project itself in the target folder. For example, when you generate output, ePublisher creates the following folder structure:

```
<Project Area>\<Project Name>\Output\<Target Name>\<Project Name>
```

In this folder structure, `<Project Area>` is the name of the ePublisher component you are using, such as `ePublisher Express`, `<Project Name>` is the name of your ePublisher project, and `<Target Name>` is the name of your ePublisher target, such as `Dynamic HTML`. To deliver your Dynamic HTML generated output, you need to deliver all the files and subfolders in the `<Target Name>\<Project Name>` folder.

The `<Target Name>\<Project Name>` folder contains the entry-point file, `toc.html` by default, which displays the table of contents. When the user opens the entry-point file, the browser uses all the files in the `<Target Name>\<Project Name>` folder to display the help, including the topic files and images.

ePUB

ePUB Platforms

ePUB Considerations

ePublisher can deliver content to your mobile users and eReader platforms with the IDPF ePUB 2.0 eBook standard. This format makes sense for long form content or reference materials which are not suitable for web delivery. eBooks enable users to access content offline, track their reading progress, and otherwise enjoy the benefits of traditional press books.

ePUB Platforms

ePUB eBooks can be accessed and viewed on a variety of desktop and mobile devices. To ensure maximum usability, this release was tested against the following ePUB readers:

- Adobe Digital Editions

<http://www.adobe.com/products/digitaleditions/>

Microsoft Windows

- Apple iBooks

<http://itunes.apple.com/us/app/ibooks/id364709193?mt=8>

Apple iPad, Apple iPhone, Apple iPod Touch

- Lexcycle Stanza (Apple iPad, iPhone, and iPod Touch)

<http://www.lexcycle.com/iphone>

Apple iPad, Apple iPhone, Apple iPod Touch, Microsoft Windows (Beta)

In addition UX tests performed on these eBook platforms, generated output was validated using the publicly available EpubCheck tool during the development cycle.

- EpubCheck 1.0.5

<http://code.google.com/p/epubcheck/>

By closely tracking the ePUB standard and conducting eReader specific tests, ePublisher attempts to deliver high quality eBooks across a wide range of ePUB capable platforms.

ePUB Considerations

The ePUB standard specifies a container and page definition syntax. ePUB eBook readers will render this information as best they can given available screen real-estate and platform considerations. Users who wish to maximize the ePUB reading experiences for a broad audience should pay careful attention to ePUB specific details.

Meta Data

You can specify standard ePUB eBook meta data in the Target Settings dialog.

- Author Name
- Author Name (File As)

Book Title and ID

All ePUB books contain a unique identifier. ePublisher will synthesize a unique ID for your book, though you may require explicit control over this value. You can specify an explicit ID via the Merge Settings dialog with the Group Context control. Further, localized book titles can be specified in the Merge Settings dialog as well.

Long Content

The ePub standard recommends breaking long content into smaller chunks to speed display rendering and reduce memory requirements on mobile devices. This can be accomplished with ePublisher's standard page break controls. You may specify a page break priority for paragraphs and tables in the Options panel of the ePublisher Style Designer.

Page Styles

Content page styles can be controlled using ePublisher's standard page style support. For the ePUB format, ePublisher also allows users to select a page style for use with the cover and index pages. These styles can be specified in the Target Settings dialog under the Cover and Index sections.

Tables

By default, ePublisher's ePUB format renders tables using paragraph markup. Users may enable true table markup via the table styles Options panel in the ePublisher Style Designer. Reasons for rendering tables using paragraph markup include:

- All tested eBook readers lacked the capability to view tables larger than the available screen space.
- The Apple iBooks reader, prior to version 2.0, fails to render images inside of table cells.

ePublisher emits conversion warnings when an image is rendered inside a table cell.

Tables can be used effectively inside of ePUB eBooks, though authors should be aware of the inherent space limitations associated with mobile reading devices. The addition of proportional (percentage based) table cell widths, used in conjunction with a table width set to 100%, may provide users with sufficient control over table rendering behavior. The behavior can be specified in the Options tab for table styles in the ePublisher Style Designer.

Cover

ePUB books support the display of a cover page or image. This format allows users to select a cover image via the Target Settings dialog. Additionally, users may customize the cover page by creating overrides for the `Pages\Cover.asp` page template.

When creating your cover page and image, keep the following points in mind:

- eBook icons
 - Adobe Digital Editions uses the cover page for the eBook icon
 - Apple iBooks and Lexcycle Stanza use the cover image for the eBook icon
 - ePublisher will embed the selected cover image into the cover page to ensure correct operation across platforms
- Cover image limitations
 - Certain readers, such as the Apple iBook reader, prefer images with a 3x4 aspect ratio. Otherwise, the eBook will not display a custom icon in iTunes.
 - Certain readers, such as the Apple iBook reader, prefer 600x800 pixel images. Otherwise, the eBook will not display a custom icon in iTunes.

Syncing with Apple iPad

1. Ensure iTunes 9.1 or greater is installed on your system. Ensure the Apple iBooks application is installed on your iPad.
2. Launch iTunes.
3. Click “Add File to Library” and select your generated ePUB eBook.
4. Attach your iPad.
5. Sync your iPad normally.
6. When synchronization completes, disconnect the iPad from your computer.
7. Launch the iBooks application on the iPad.
8. Within iBooks, you will see your new book on the virtual bookshelf.

These instructions also apply when adding ePUB eBooks to Apple iPhone and Apple iPod Touch devices.

Eclipse Help

Eclipse Help Viewer Delivering Eclipse Help

To deliver help on multiple platforms, you need a solution that runs on all those platforms. This solution helps you avoid complications of each platform and allows you to deliver one solution that meets the needs on all your platforms. This common solution can save development time and effort.

ePublisher supports several Java-based help output formats, such as Eclipse Help, Oracle Help, and Sun JavaHelp, that provide a common solution for multiple platforms without JavaScript support. WebWorks Help and WebWorks Reverb also provide a cross-platform solution, but it requires JavaScript support. If you are delivering a Java-based application, or if your environment does not support JavaScript, a Java-based help solution can help you deliver your help content.

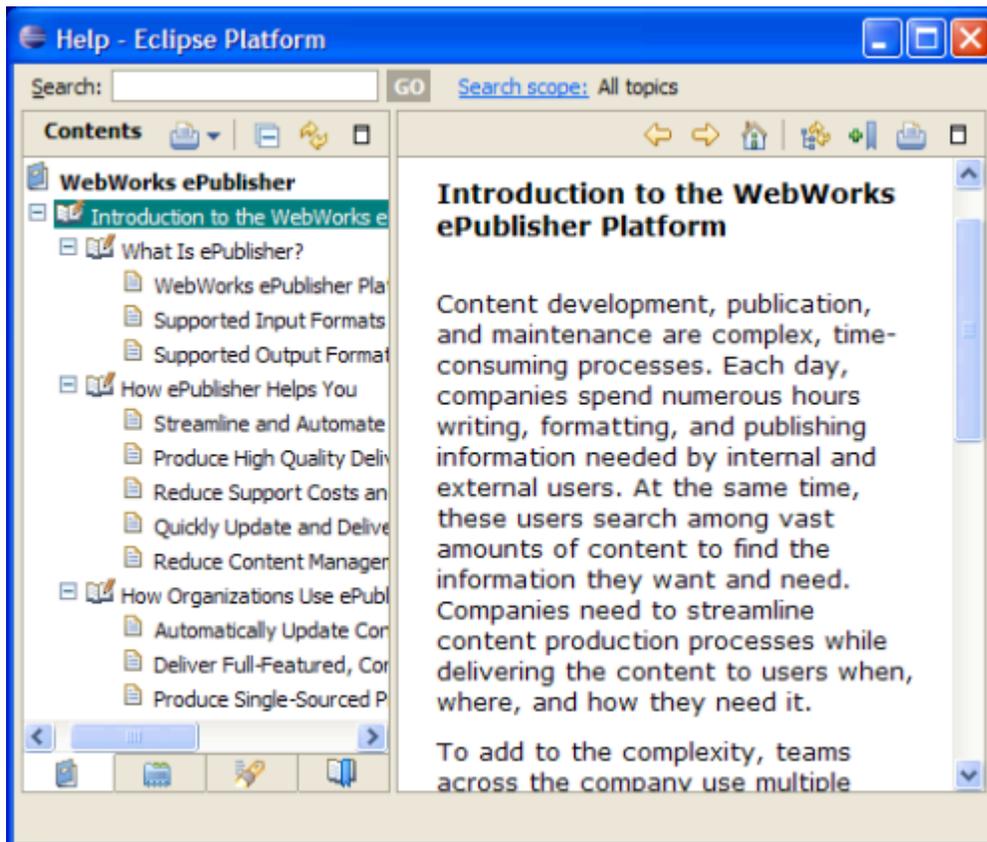
The Eclipse Help format uses a Java-based delivery environment to provide a comprehensive help viewer. Eclipse Help delivers content in HTML files and uses an XML-based table of contents. Once you install the Eclipse platform, you can view Eclipse Help files. ePublisher provides a standalone viewer for Eclipse Help so you can develop and view your Eclipse Help.

Eclipse Help requires the Eclipse integrated development environment (IDE), which makes it a good output format choice for products that already install and use the IDE. For more information about Eclipse Help, see the Eclipse SDK documentation at: help.eclipse.org/help33/index.jsp?topic=/org.eclipse.platform.doc.isv/guide/ua_help.htm.

You can also use Eclipse Help to deliver an Infocenter. You can use all the supported input formats to develop content for an Infocenter. Then, ePublisher allows you to publish that content in the required format. For more information about creating an Infocenter, see dita.xml.org/wiki/setting-up-the-eclipse-help-infocenter-for-publishing-dita-content.

Eclipse Help Viewer

The Eclipse Help viewer uses an embedded Apache Tomcat server. Similar to other help viewers, the Eclipse Help viewer provides a navigation pane with multiple tabs and a topic pane.



Delivering Eclipse Help

You can provide a help system in Eclipse Help format either as a collection of individual files or in a single, compressed Java archive `.jar` file. A `.jar` file, similar to a `.zip` file, compresses and stores a collection of files.

When you generate output for your project, ePublisher creates the `Output` folder in your project folder, and then creates a folder in the `Output` folder for each generated target. Then, ePublisher creates a folder named for the project itself in the target folder. For example, when you generate output, ePublisher creates the following folder structure:

```
<Project Area>\<Project Name>\Output\<Target Name>\<Project Name>
```

In this folder structure, `<Project Area>` is the name of the ePublisher component you are using, such as `ePublisher Express`, `<Project Name>` is the name of your ePublisher project, and `targetname` is the name of your ePublisher target, such as `Eclipse Help`. To deliver your Eclipse Help generated output, you can provide the complete contents of the `<Target Name>\<Project Name>` folder to your Java application developers, and they can determine whether to deliver the `.jar` file or the individual files to application users.

HTML Help

[Benefits of Microsoft HTML Help](#)

[Restrictions and Requirements for Microsoft HTML Help](#)

[HTML Help Viewer](#)

[Topic Only View in HTML Help](#)

[HTML Help Workshop](#)

[Delivering HTML Help](#)

Delivering help information in a consistent manner on Windows computers was an important concern for Microsoft. Initially, Microsoft provided the WinHelp format for help content delivery. As the Windows platform advanced, Microsoft introduced the Microsoft HTML Help format. To help authors deliver content through the HTML Help Viewer provided with the Windows platform, Microsoft provides a help compiler and toolkit, HTML Help Workshop, that allows you to build Microsoft HTML Help `.chm` files.

The Microsoft HTML Help output format provides a standard help format for products on computers running the Windows operating system. The Microsoft HTML Help format delivers a single, compiled file that includes multiple source files compressed into one `.chm` file. Unlike the RTF-based Microsoft WinHelp format, the Microsoft HTML Help format is based on HTML/XHTML. Microsoft HTML Help is recommended to produce online help for 32-bit applications that run on a computer running Microsoft Windows 95 or later, including Windows Vista:

Microsoft HTML Help provides a comprehensive help format, including a table of contents, index, full-text search, and favorites in one integrated viewer window. The HTML Help Viewer uses standard Internet Explorer components and supports many Web technologies, such as HTML, ActiveX, Java, JavaScript, JScript and other scripting languages, and the Web image formats, such as `.gif`, `.jpg`, and `.png`.

In HTML Help, writers list a collection of source files in a help project `.hhp` file with other project-related settings. The writers then use HTML Help Workshop to compile the help and create the `.chm` file. During compilation, HTML Help Workshop uses the help project `.hhp` file to determine how HTML topic files, image files, contents `.hhc` files, index `.hhk` files, and any other elements appear in the single, compressed help file. ePublisher automates this process and integrates it into help generation.

Benefits of Microsoft HTML Help

The Microsoft HTML Help output format generates output files that conform to the HTML 4 and XHTML 1.0 standards. The content uses Cascading Style Sheets that conform to the CSS1 standard. This output format produces all the files required to create the Microsoft HTML Help `.chm` file.

ePublisher streamlines the help generation process. ePublisher uses your Stationery and project settings to transform your source documents to the input files needed by Microsoft HTML Help Workshop. Then, ePublisher uses the help compiler to build the help project and create the `.chm` file. This integrated process simplifies the authoring environment, but it also creates all the underlying files to help you customize the process, if needed.

Microsoft HTML Help provides several important benefits:

- You have a single, compressed `.chm` output file to deliver.
- Your audience can read your content in the standard Windows HTML Help Viewer.
- The HTML Help Viewer provides a powerful search engine that supports partial words, synonym searching, and other advanced search options.

Restrictions and Requirements for Microsoft HTML Help

The Microsoft HTML Help output format is recommended to produce standard help for 32-bit applications that run on a computer running Windows 95 or later. For more information, see “Requirements”. Review the following considerations when deciding whether to deliver HTML Help:

- Your audience cannot read the `.chm` file over a network or over the Web. The `.chm` file must be installed on the local computer.
- On some systems, CHMs cannot be viewed if they were generated in an ePublisher project that is located on a network drive. To resolve the problem, move the entire ePublisher project to a local drive, on the same computer where ePublisher is installed. It is OK to keep the source documents on a network drive.
- Some security settings can interfere with `.chm` file use.
- A `.chm` file supports many Web technologies, such as scripting languages, which also allow them to contain and transport viruses and security risks. For this reason, many email systems remove `.chm` files when attached to an email message.

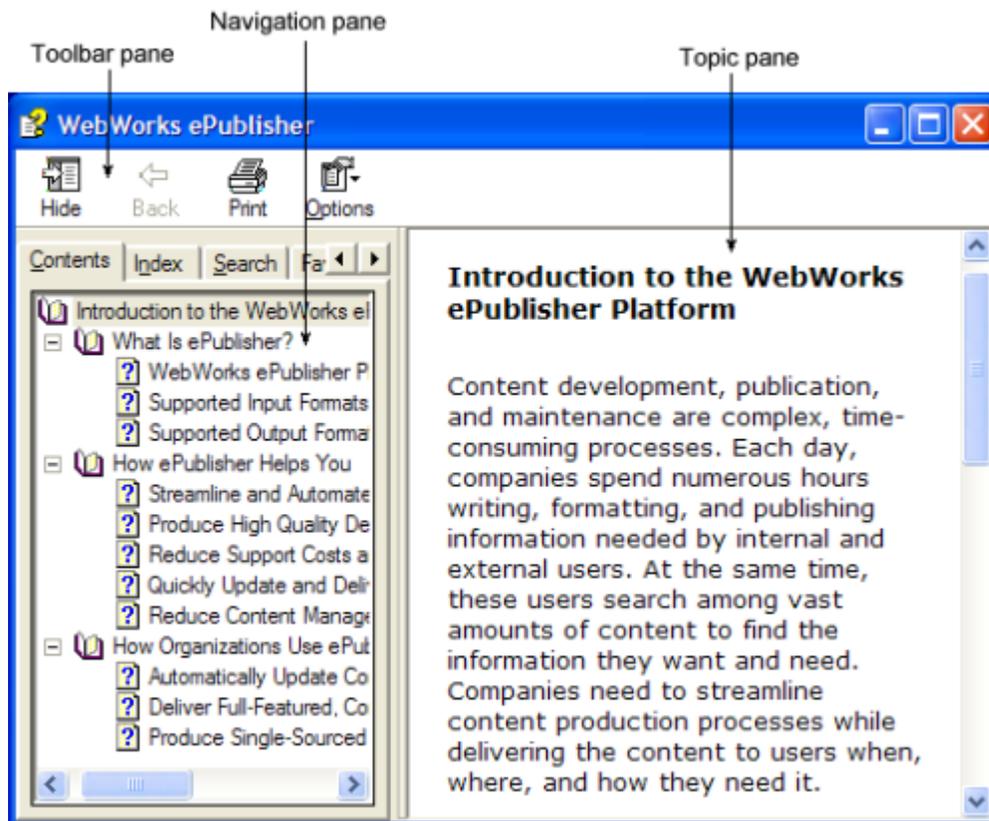
HTML Help Viewer

When you open a Microsoft HTML Help `.chm` file, Windows displays the content in the Help Viewer window. Microsoft HTML Help Workshop allows the writer to compile and customize the `.chm` file. ePublisher uses your source documents to generate all the files needed to compile the `.chm` file.

By default, the tripane HTML Help Viewer includes a table of contents, index, full-text search, and favorites in one, integrated viewer window. You can customize this viewer window. The user also has options to show or hide the navigation pane, based on the options you defined.

The default tripane view includes the following panes:

- The **toolbar pane** displays buttons that provide additional functions and navigation options to the user. HTML Help Workshop allows you to select which buttons to include in your help file.
- The **navigation pane** displays tabs to access the table of contents, index, full-text search, and favorite topics. HTML Help Workshop allows you to select which tabs to include in your help file.
- The **topic pane** displays the information contained in the source documents.



Toolbar Pane in HTML Help

The toolbar pane provides several buttons that provide additional functions, such as returning to the previously viewed topic, showing and hiding the navigation pane, and printing a help topic. The writer can customize which buttons are included in the toolbar pane through HTML Help Workshop. The following table lists the supported buttons.

Button	Description
Hide/Show	Opens or closes the Navigation pane.
Back	Displays the previously viewed topic like the Back button in a browser.
Forward	Displays the previously viewed topic if the user clicked the Back button. This button functions like the Forward button in a browser.
Stop	Stops retrieving the file information and contents.
Refresh	Updates the topic that is currently displayed in the topic pane.
Home	Displays the default topic you defined for the help file.
Options	Opens a menu that provides many commands, such as Home, Show, Back, Stop, Refresh, Print, Search Highlight On/Off, and Internet Options. This menu also includes commands for all the buttons included on help windows.
Print	From the Contents tab, this button allows the user to print the selected topic, and optionally all subtopics. From the Index or Search tab, opens the Print dialog box to print the current topic.
Locate	Displays in the table of contents the location of the current topic is not listed in the table of contents, this command will not work.
Jump 1	Jumps to an author-designated Web page or help topic.
Jump 2	Jumps to an author-designated Web page or help topic.

Navigation Pane in HTML Help

By default, the navigation pane provides the following tabs, which you can include or exclude from your help output through HTML Help Workshop:

Contents tab

Displays the table of contents in the form of an expand/collapse tree view. The table of contents includes all paragraph styles that you assigned a TOC level in Style Designer. When the user selects an entry in the Contents tab, the information from that topic is displayed in the topic pane. You can customize the icons used for specific topics within the table of contents.

Index tab

Displays an alphabetical list of keywords associated with topics in the source documents. Writers use their source document authoring tool to create standard index markers or field codes that define these keywords in their source documents.

Search tab

Provides a powerful full-text search feature with several advanced search options. The user can type a search string and then select **Display** to view a list of topics that contain the word or phrase specified.

Favorites tab

Displays a list of topics in the help that the user has added to his or her personal list of favorites. This tab allows users to bookmark help topics they often use or want to quickly find in the future. When the user selects a topic on this tab, the information from that topic is displayed in the topic pane.

Topic Pane in HTML Help

The output pages generated from your source documents are displayed in the topic pane. When a user selects a topic on any of the tabs in the navigation page, the content of that topic is displayed in the topic pane.

Topic Only View in HTML Help

The show/hide button in the toolbar pane allows users to add or remove the navigation pane. The writer can also define a window in HTML Help Workshop that does not include the navigation pane, or with the navigation pane hidden by default. Without the navigation pane, more screen area is available to display the help topic content, or for the application itself. However, the navigation pane helps users view where they are in the organization of topics, and quickly browse and access other topics in the table of contents.

HTML Help Workshop

HTML Help Workshop is a help authoring tool that allows you to create and manage Microsoft HTML Help projects and their related files. You can use this tool, which is installed with ePublisher, to further customize your help, such as changing the appearance of the Contents and Search tabs. You can also create an override for your project `.hhp` file in your ePublisher project, which allows you to implement the custom HTML Help Viewer window you need. For more information about Microsoft HTML Help and its related files and customization options, see the help for HTML Help Workshop.

HTML Help Project File (.hhp)

The HTML Help project `.hhp` file identifies all the elements of an HTML Help project. This project file is separate from the ePublisher project. The HTML Help project file contains the information HTML Help Workshop needs to combine the source files, images, index, and table of contents into a single, compiled help `.chm` file.

The HTML Help project file also defines the appearance and behavior of the HTML Help Viewer window. ePublisher creates the HTML Help project file based on the `template.hhp` file and the settings in your ePublisher project. You can override the default `template.hhp` file to adjust the default appearance of your HTML Help, such as the default size and position of the HTML Help Viewer window or the buttons displayed in the toolbar pane. You can also define additional windows in the `template.hhp` file, such as a window without the navigation pane. For more information, see “Adjusting the HTML Help Viewer Window Size and Toolbar Buttons” and “Creating an Additional HTML Help Window Definition”.

HTML Help Contents File (.hhc)

The HTML Help contents `.hhc` file defines the table of contents for the help. This file includes entries for each occurrence of the paragraph styles for which you assigned a TOC level in your ePublisher project. HTML Help displays the contents of this file on the Contents tab of the navigation pane. ePublisher generates the `toc.hhc` file for your project using the `template.hhc` file.

HTML Help Index File (.hhk)

The HTML Help index `.hhk` file defines the index for the help. This file includes entries based on the index markers and field codes in your source documents. HTML Help displays the contents of this file on the Index tab of the navigation pane. ePublisher generates the `index.hhk` file for your project using the `template.hhk` file.

HTML Help Mapping File (.h)

The HTML Help mapping `.h` file identifies the mapping information for context-sensitive help links. This file includes an entry for each TopicAlias marker or field code in your source documents. Application developers build this file with their project and use the defined aliases to display the appropriate topic in the help. This file is linked in the `MAP` section of the `.hhp` file, and its entries coordinate with the entries in the `ALIAS` section of the `.hhp` file.

Delivering HTML Help

When you generate output for your project, ePublisher creates the `Output` folder in your project folder, and then creates a folder in the `Output` folder for each generated target. Then, ePublisher creates a folder named for the project itself in the target folder. For example, when you generate output, ePublisher creates the following folder structure:

```
<Project Area>\<Project Name>\Output\<Target Name>\<Project Name>
```

In this folder structure, `<Project Area>` is the name of the ePublisher component you are using, such as `ePublisher Express`, `<Project Name>` is the name of your ePublisher project, and `<Target Name>` is the name of your ePublisher target, such as `Microsoft HTML Help 1.x`.

By default, ePublisher compiles the `.chm` and stores it in the `<Target Name>` folder. To deliver your HTML Help, you need to deliver the `.chm` file.

If you created context-sensitive help, your application developer needs to build with the `<Target Name>\<Project Name>\<Project Name>.h` file. If you created What's This field-level help, your application developer also needs to build with the `<Target Name>\<Project Name>\whatisthis.h` file. For more information about these mapping files, see “Using Context-Sensitive Help in HTML Help” and “Defining What's This (Field-Level) Help in HTML Help”.

Oracle Help

[Oracle Help Viewer](#)
[Oracle Help Files](#)
[Delivering Oracle Help](#)

To deliver help on multiple platforms, you need a solution that runs on all those platforms. This solution helps you avoid complications of each platform and allows you to deliver one solution that meets the needs on all your platforms. This common solution can save development time and effort.

ePublisher supports several Java-based help output formats, such as Eclipse Help, Oracle Help, and Sun JavaHelp, that provide a common solution for multiple platforms without JavaScript support. WebWorks Help and WebWorks Reverb also provide a cross-platform solution, but it requires JavaScript support. If you are delivering a Java-based application, or if your environment does not support JavaScript, a Java-based help solution can help you deliver your help content.

The Oracle Help format is a Java-based help format that produces the complete set of files required to deliver online help based on the Oracle Help for Java technology. Oracle Help for Java is a set of Java components and an API for developing and displaying HTML-based help content in a Java environment. Oracle Help is recommended to produce help for an application written in the Java programming language.

Note: Oracle also offers a separate technology called Oracle Help for the Web. ePublisher does not support Oracle Help for the Web.

To view Oracle Help, users must have a Java Virtual Machine (JVM) installed. If you produce help for a Java application, the JVM will be installed on computers running the application. The Oracle Help components must also be installed on the computer.

The Oracle Help output format produces content that conforms to the HTML 4 and XHTML 1.0 standards and uses Cascading Style Sheets that conform to the CSS1 standard. This output format also produces files that are specifically required for Oracle Help. For more information about the Oracle Help technology, see the Oracle Web site at www.oracle.com/technology/tech/java/help.

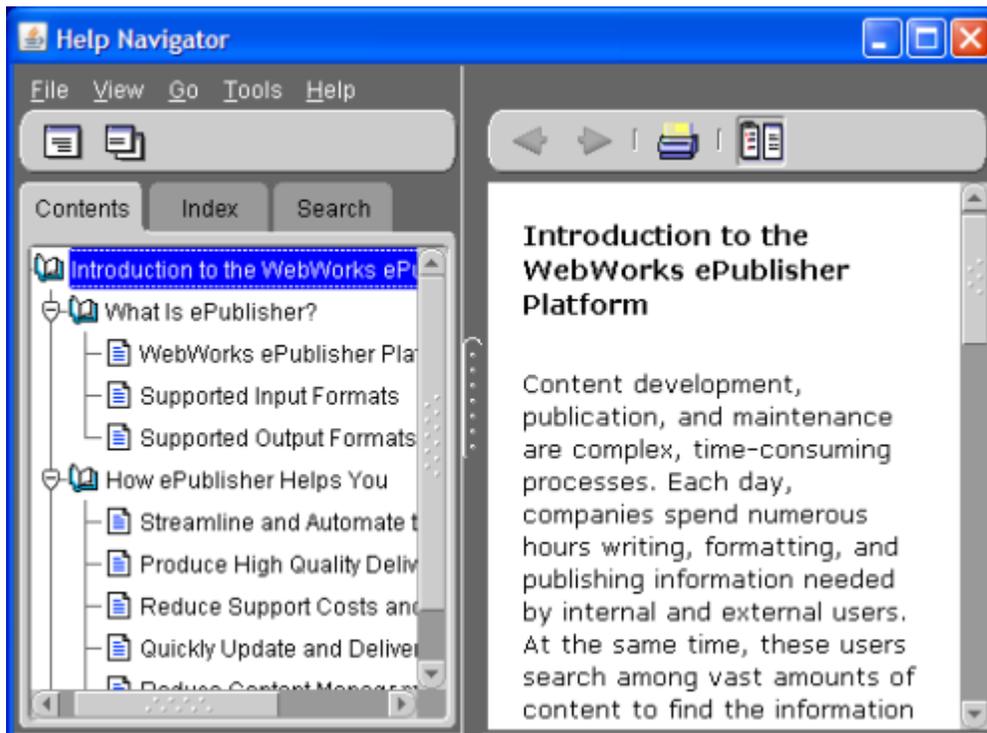
When deciding whether to use the Oracle Help output format, review the following considerations:

- Oracle Help is one of the output formats you can use to produce help for a Java application. You can also use the Oracle Help output format to produce content for the Web if all your users have, or are prepared to install, a Java Virtual Machine.
- If you need to produce help for a pure Java application, you may also consider using the Eclipse Help and Sun JavaHelp output formats. Each output format has slightly different requirements, appearance, and behavior.

- If you need to produce help for a Web-based application hosted on a Web server or on an intranet server that is not a pure Java application, you may consider using the WebWorks Help or WebWorks Reverb output format.

Oracle Help Viewer

The Oracle Help output format produces Java-based online help. Oracle Help includes a navigation pane and a topic pane that is similar to other help viewers.



Oracle Help Files

Oracle Help has several files that work together to define and create the Oracle Help solution. ePublisher incorporates these files into the compressed helpset file, so you do not need to distribute them to users.

Helpset .hs File in Oracle Help

The helpset `.hs` file contains descriptive information about your Oracle Help helpset. ePublisher generates this file using the `template.hs` page template. For more information about the helpset file, see the Oracle Help documentation.

Control .xml Files

The control `.xml` files define the table of contents, index, and topic aliases in your Oracle Help project. These files are named `projectTOC.xml`, `projectMap.xml`, `projectLinks.xml`, and `projectIndex.xml`, where *project* is the name of your ePublisher project. For more information about these files, see the Oracle Help documentation.

Full Text Search .idx Index File

The internal full text search `.idx` index file provides support for the full text search feature.

Manifest .mft File

ePublisher automatically generates the manifest `.mft` file. This internal file is used to create the final Java archive `.jar` file that you distribute to users.

Delivering Oracle Help

You can provide a help system in Oracle Help format either as a collection of individual files or in a single, compressed Java archive `.jar` file. A `.jar` file, similar to a `.zip` file, compresses and stores a collection of files.

When you generate output for your project, ePublisher creates the `Output` folder in your project folder, and then creates a folder in the `Output` folder for each generated target. Then, ePublisher creates a folder named for the project itself in the target folder. For example, when you generate output, ePublisher creates the following folder structure:

```
<Project Area>\<Project Name>\Output\<Target Name>\<Project Name>
```

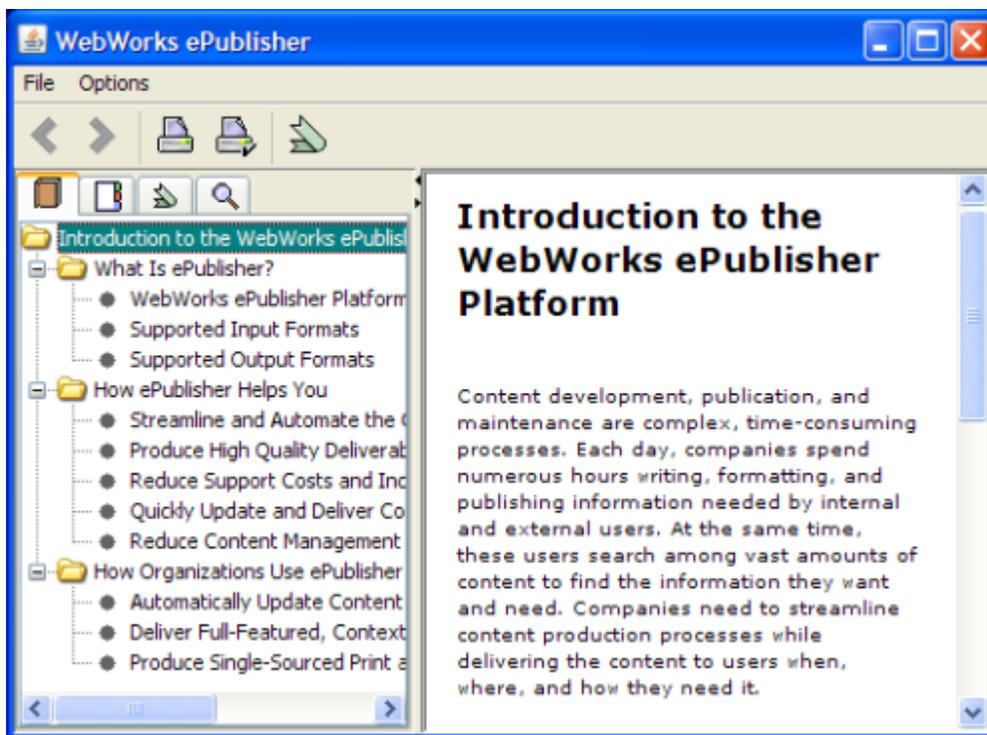
In this folder structure, `<Project Area>` is the name of the ePublisher component you are using, such as `ePublisher Express`, `<Project Name>` is the name of your ePublisher project, and `<Target Name>` is the name of your ePublisher target, such as `Oracle Help`. To deliver your Oracle Help generated output, you can provide the complete contents of the `<Target Name>\<Project Name>` folder to your Java application developers, and they can determine whether to deliver the `.jar` file or the individual files to application users.

Sun JavaHelp

Sun JavaHelp Files

Delivering Sun JavaHelp

Similar to other help systems, Sun JavaHelp provides a navigation pane and a topic pane. The navigation pane provides Contents, Index, Favorites, and Search tabs. The topic pane displays the content of the help topic selected on a tab in the navigation pane.



Sun JavaHelp Files

Sun JavaHelp delivers content in HTML files. The table of contents, index, and helpset `.hs` files use the Extensible Markup Language (XML) format.

Helpset .hs File in Sun JavaHelp

The helpset `.hs` file contains configuration information in XML format that Sun JavaHelp needs to display your help system. The following figure shows a sample helpset file:

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE helpset PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp HelpSet Version
1.0//EN" "http://java.sun.com/products/javahelp/helpset_1_0.dtd">
<helpset version="1.0">
  <title>Interesting help project</title>
  <maps>
    <homeID>home</homeID>
    <mapref location="test.jhm" />
  </maps>
  <view>
    <name>TOC</name>
    <label>Interesting help project</label>
    <type>javax.help.TOCView</type>
    <data>testt.xml</data>
  </view>
  <view>
    <name>Index</name>
    <label>Index</label>
    <type>javax.help.IndexView</type>
    <data>testi.xml</data>
  </view>
  <view>
    <name>Search</name>
    <label>Search</label>
    <type>javax.help.SearchView</type>
    <data engine="com.sun.java.help.search.DefaultSearchEngine">
      JavaHelpSearch
    </data>
  </view>
</helpset>
```

The helpset file begins with an XML declaration. The next part of the file, as shown in the following sample, defines the help version, the title of the help, and the location of the mapping file that is used to support context-sensitive help:

```
<helpset version="1.0">
  <title>Interesting help project</title>
  <maps>
    <homeID>home</homeID>
```

```
<mapref location="test.jhm" />
```

```
</maps>
```

The remaining sections of the file define several views, such as the table of contents (TOC), index, and search. Each `<view>` tag includes several tags that define the view, such as the `<name>` tag that specifies the name of the view and the `<data>` tag that specifies the data file for the view.

You can also define your own views in Sun JavaHelp by overriding the `template.hs` file. For example, you can create an alternate list of topics and provide that list as a separate view.

Contents toc.xml File in Sun JavaHelp

The contents `toc.xml` file defines the items in the table of contents as shown in the following example file:

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE toc PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp TOC Version 1.0//EN"
"http://java.sun.com/products/javahelp/toc_1_0.dtd">
<toc version="1.0">
  <tocitem target="tutorial_htm_1003580" text="Hybrid Web/CD:">
  <tocitem target="tutorial_htm_1004263" text="Roadmap"/>
  <tocitem target="tutorial_htm_1005107" text="Explosion"/>
  <tocitem target="tutorial_htm_999374" text="Bandwidth"/>
  ...
</tocitem>
</toc>
```

Index ix.xml File in Sun JavaHelp

The index `ix.xml` file lists your index entries as shown in the following example file:

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE index PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp Index Version 1.0//EN"
"http://java.sun.com/products/javahelp/index_1_0.dtd">
<index version="1.0">
  <indexitem target="tutor4_htm_999680" text="1996 Telecom Act"/>
  <indexitem target="tutor11_htm_999661" text="active catalog"/>
  <indexitem target="tutor10_htm_999632" text="ATM"/>
  <indexitem target="tutor10_htm_999634" text="audio synchronization"/>
  <indexitem text="bandwidth">
    ...
</index>
```

Map .jhm File in Sun JavaHelp

Sun JavaHelp supports context-sensitive help through a mapping `.jhm` file. The mapping file contains information that links your Sun JavaHelp topics to particular locations in the Java application. For more information, see “Using Context-Sensitive Help in Oracle Help and Sun JavaHelp”.

Delivering Sun JavaHelp

You can provide a help system in Sun JavaHelp format either as a collection of individual files or in a single, compressed Java archive `.jar` file. A `.jar` file, similar to a `.zip` file, compresses and stores a collection of files.

When you generate output for your project, ePublisher creates the `Output` folder in your project folder, and then creates a folder in the `Output` folder for each generated target. Then, ePublisher creates a folder named for the project itself in the target folder. For example, when you generate output, ePublisher creates the following folder structure:

```
<Project Area>\<Project Name>\Output\<Target Name>\<Project Name>
```

In this folder structure, `<Project Area>` is the name of the ePublisher component you are using, such as `ePublisher Express`, `<Project Name>` is the name of your ePublisher project, and `<Target Name>` is the name of your ePublisher target, such as `JavaHelp`. To deliver your Sun JavaHelp generated output, you can provide the complete contents of the `<Target Name>\<Project Name>` folder to your Java application developers, and they can determine whether to deliver the `.jar` file or the individual files to application users.

WebWorks Help

[The Frameset View in WebWorks Help](#)

[Topic Only View in WebWorks Help](#)

[WebWorks Help Output Files](#)

[Delivering WebWorks Help](#)

[Searching WebWorks Help](#)

Delivering help content on multiple platforms and browsers can be a challenge. You need a solution that provides advanced help features, such as full-text search, expand/collapse sections, browse and breadcrumb navigation, related topics, and a table of contents tree control. WebWorks Help delivers these features and others on multiple platforms and browsers. Whether your help needs to be viewed on the Macintosh or Windows operating system, displayed with Firefox, Safari, or Internet Explorer, WebWorks Help provides the flexible, adaptable help solution you need.

Note: Users considering a web help output should first consider a Reverb 2.0 output before choosing Reverb or WebWorks Help. Reverb 2.0 is designed to work across modern browsers, and includes our latest features. WebWorks Help 5 is recommended for web server use only.

The WebWorks Help format combines standard XHTML, CSS1, and CSS2 with JavaScript to provide many online help navigation controls and features, including an expandable/collapsible table of contents, an index, full text search, and Related Topics menus. You can easily integrate WebWorks Help with applications written in C, C++, Java, or Visual Basic, as well as Web-based applications. This comprehensive help format allows you to quickly create and deploy context-sensitive help and comprehensive help that you can fully customize to match your corporate branding.

WebWorks Help enables you to publish your content in a variety of ways:

- On a Web server
- On a network server available to multiple users through a share
- On the local computer where the application is installed
- On a CD-ROM

The Frameset View in WebWorks Help

When you open WebWorks Help, you can choose whether to display the navigation pane and the content, or only the content in the browser. You can control the way WebWorks Help opens based on whether the user selected a context-sensitive help link or opened the comprehensive help. This flexibility allows you to deliver your content the way the user needs it.

The frameset view divides the browser window into several areas:

Navigation pane

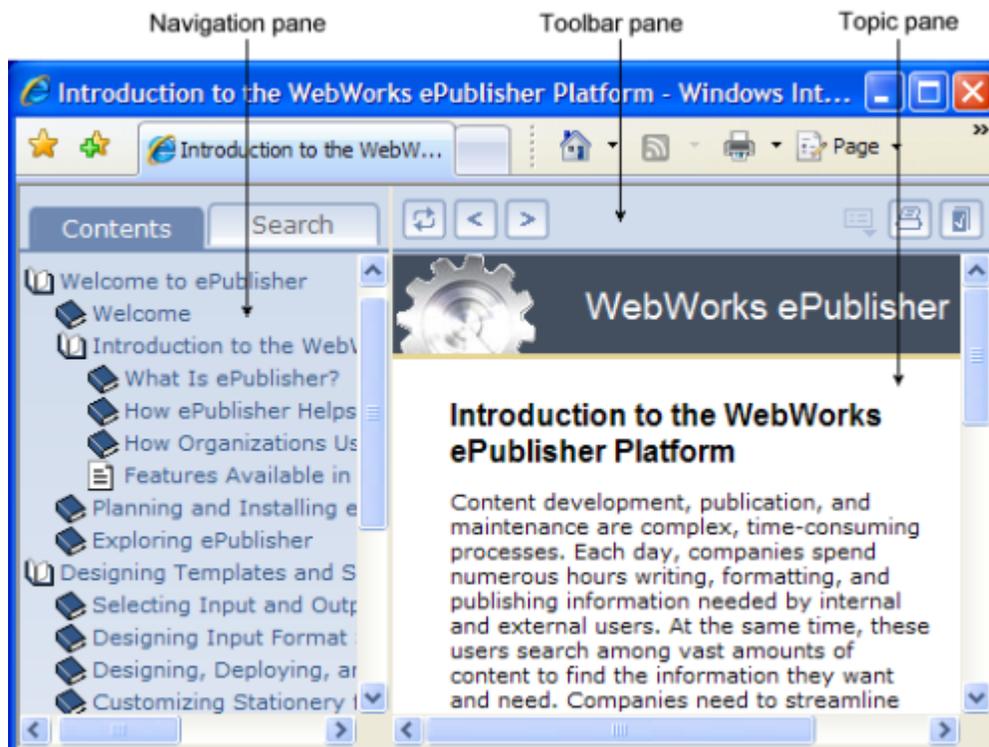
Provides multiple tabs for navigation elements, such as the table of contents and full-text search.

Toolbar pane

Provides several standard toolbar buttons to perform common tasks, such as show or hide the navigation pane, browse to the next topic, and print the content.

Topic pane

Displays the content of the topic selected in the navigation pane.



Navigation Pane in WebWorks Help

By default, the navigation pane displays several tabs that provide core navigational features for your help:

Contents tab

Displays the table of contents, including entries for each style that you assigned a TOC level value in Style Designer. When a user clicks on item in the Contents tab, the selected topic is displayed in the topic pane. This tab displays the table of contents as an expand/collapse tree view. Each book icon represents a table of content entry that has subentries.

Index tab

Displays an alphabetical list of keywords associated with topics. To view index entries, select a letter to display the entries that start with that letter. When the user clicks on an index entry, the related topic is displayed in the topic pane. The writer defines the keywords as index entries in the source documents.

Search tab

Provides a full-text search. When the user clicks **Go**, WebWorks Help lists the titles of the topics whose content contains the words specified in the **Search** field. This tab provides **Rank** and **Title** columns. Each listed topic has a relevancy ranking number, which reflects how well the topic matches the search criteria. The ranking is specified in the `wwh_files.xml` file. For more information about how to modify the ranking, see “Modifying the Search Ranking”.

Favorites tab

Lists the topics that the current user added to his or her list of personal favorites. In WebWorks Help, users can add frequently accessed or important help topics to their personal list of favorites. When the user clicks on a topic on the Favorites tab, the help topic is displayed in the topic pane. The Favorites tab also provides a Remove button that allows users to delete any unwanted topics from their list.

Toolbar Pane in WebWorks Help

By default, the toolbar pane displays a set of buttons that provide users with additional features. These buttons allow users to navigate through the help and access common functions, such as printing the displayed topic and emailing a link to a topic. The standard WebWorks Help toolbar includes the following buttons:

Show in Contents button

Allows users to locate the topic they are viewing in the table of contents. When a user clicks this button, WebWorks Help highlights the entry in the Contents tab that corresponds to the currently displayed topic.

Show Navigation button

Allows users to show the hidden navigation pane. When a context-sensitive help link displays a topic with the navigation pane hidden, this button is displayed in the toolbar. When a user clicks this button, WebWorks Help displays the navigation pane and highlights the entry in the Contents tab that corresponds to the currently displayed topic.

Previous button

Allows users to navigate back to topics that precede the currently displayed topic in the help.

Next button

Allows users to navigate forward to topics that follow the currently displayed topic in the help.

PDF button

Displays a PDF file of the source document from which the currently displayed topic was generated.

Related Topics button

Displays a list of topics that share a common or related theme with the currently displayed topic. The writer must define related topics in the source documents and the Stationery designer must enable related topics support for the help to display them. For more information, see “Defining Related Topics”.

Print button

Prints the currently displayed topic.

Email button

Allows you to collect feedback from your users. This button opens a blank email message addressed to the email address you specify. The subject line of the email message identifies the displayed topic when the user clicked the Email button.

Topic Pane in WebWorks Help

This pane displays the content of the help topics generated from your source documents. When a user selects a topic on the Contents, Search, Index, or Favorites tab, the topic pane displays the content of that topic.

Topic Only View in WebWorks Help

You can display a topic-only view, which hides the navigation pane and displays only the toolbar pane and the topic pane. With this topic-only view, an application can open a smaller browser window that leaves more area on the screen for the application itself. Context-sensitive help often uses the topic-only view.

WebWorks Help Output Files

When you generate the WebWorks Help output format, ePublisher creates all the files required to deliver and display WebWorks Help. The `index.html` file is the default entry file that defines the frameset used by the help. To open the help, users open this entry file. ePublisher creates the `.html`, `.css`, `.js`, and various image files included in the WebWorks Help output.

Delivering WebWorks Help

When you generate output for your project, ePublisher creates the `Output` folder in your project folder, and then creates a folder in the `Output` folder for each generated target. Then, ePublisher creates a folder named for the project itself in the target folder. For example, when you generate output, ePublisher creates the following folder structure:

```
<Project Area>\<Project Name>\Output\<Target Name>\<Project Name>
```

In this folder structure, `<Project Area>` is the name of the ePublisher component you are using, such as `ePublisher Express`, `<Project Name>` is the name of your ePublisher project, and `<Target Name>` is the name of your ePublisher target, such as `WebWorks Help`. To deliver your WebWorks Help generated output, you need to deliver all the files and subfolders in the `<Target Name>\<Project Name>` folder.

The `<Target Name>\<Project Name>` folder contains the entry-point file, `index.html` by default, which establishes the help set appearance. When the user opens the entry-point file, the browser uses all the files in the `<Target Name>\<Project Name>` folder to display the help, including all the topic files, generated `.css` files, `.pdf` files, images, and WebWorks Help components.

Searching WebWorks Help

Use the following guide to assist users in finding terms in a WebWorks Help 5.0 help system:

Boolean

All search words and phrases have an implicit AND

Word Search

For example, searching: `eggs bacon` returns all documents containing “eggs” and “bacon”

Phrase Search

For example, searching `"Good Morning"` returns all documents containing “good” and “morning” where the two words are adjacent

Wildcard

For example, searching `e*` returns all documents containing words that start with e.

To further customize the way the search results are displayed, See [Modifying the Search Ranking](#)