

This is an archived version of Help Essentials and will no longer be updated. Downloads and purchases no longer work. Visit [Grainge.org](https://www.grainge.org) to access downloads. ✕

[Shop](#)[About](#)[Services](#)[Blog](#)[Contact](#)[My Account](#)[Basket ⁰](#)

Context sensitivity in WebHelp and FlashHelp

By Willam van Weelden on 7th August 2017

Contents [\[hide\]](#)

[Custom API's](#)[Topic ids and Map numbers](#)[Context Sensitivity and APIs](#)[Calling WebHelp for Merged Projects](#)[Calling Dynamic User Centric Content](#)[Methods for Calling WebHelp](#)[Regular start page](#)[Calling WebHelp](#)[Default method](#)[Using a window and not opening in a popup](#)[Internal method](#)[_csh start file](#)[_rhc start file](#)

Context sensitive help is help set up to open at a topic relevant to the field or screen that the user is viewing. This article provides an overview of the way context sensitivity works in WebHelp and FlashHelp generated by RoboHelp. This article shows you how you can create URLs to call your help context sensitive. This



Categories

[In-depth](#)[JavaScript](#)[Output manipulation](#)[Products](#)[RoboHelp](#)[Tips](#)[Tutorials](#)[Uncategorised](#)

Tags

[AIR Help](#) [Android](#)[Browse Sequences](#)[Captivate](#) [CHM](#)[Conference](#)[Context Sensitive Help](#)[CSH](#) [Downgrade](#)[Fonts](#) [FrameMaker](#)[HTML5](#) [Images](#)[Index](#) [JavaScript](#)

allows you to create your own context sensitive help calls and allows you to create your own API. I focus on developers and others who want to know what options are available to them. I assume you have some experience with RoboHelp and WebHelp as I will be using default RoboHelp terminology.

If you are looking for a quick tutorial to get working with context sensitive WebHelp or the default RoboHelp APIs, see [Calling WebHelp](#) on Peter Grainge's site and the article [Providing Context-Sensitive Help for Web Applications](#) by John Daigle.

Note:

- I tested the methods provided on this page with RoboHelp 7 and later. This description may work for earlier versions, but I haven't tested that.
- FlashHelp works **exactly** the same as WebHelp. In all the subarticles, I will omit 'FlashHelp' and use only the term 'WebHelp'.
- This article focus is solely on WebHelp. Calling WebHelp Pro using the [RoboHelp Server](#) is different from 'regular' WebHelp.
- If you have opened a WebHelp project in your browser and you feed a context sensitive URL into the address bar, the call may fail. There are two solutions to this:
 - First navigate to another URL such as <http://www.google.com> and then try the URL again.
 - Amend the call to avoid conflict. See the table at the start of http://www.grainge.org/pages/authoring/calling_webhelp/using_urls.htm for more information.

Lists LMS

Merged Help

Mobile Navigation

Navigation Pane

Patch

Responsive HTML5

Review RoboHelp

RoboSource Control

Scripts SharePoint

Social Media

Source Control

Subversion

Team Foundation Server

TFS Video

Visual Studio

WebHelp Widget

YouTube

ZoomSearch

Archives

November 2018

August 2017

July 2017

June 2017

March 2017

February 2017

November 2016

September 2016

August 2016

April 2016

March 2016

Custom API's

To help with developing your own API's, here are a few custom API's that you can use out of the box.

- [C# WebHelp API](#)
- [JavaScript WebHelp API](#)
- [PHP WebHelp API](#)

Topic ids and Map numbers

In order to work with context sensitive help, you or the author of the help system can create ids for the topics in your project you want to call. (You can also omit ids and use URLs.) In RoboHelp you set these ids using the Project Set-up pod. In the *Project Set-up pod*, go to *Context-Sensitive Help > Map Files*. You create the context sensitivity by opening a map file.

When using ids, you assign a unique topic id and a unique Map number to a topic or an anchor in a topic. The Map number is a unique number ranging from 1 to 4,294,967. The Map number is the id used for context sensitive help in CHM files and can also be used for WebHelp.

The topic id (called Map ID in the RoboHelp interface) is a unique alphanumeric string that identifies the topic. The topic id has a maximum of 255 characters and you can use letters, numbers and underscores. A topic id can also consist solely of numbers. The topic id is used by RoboHelp to connect a topic in the project to a map number.

Note:

November 2015

May 2015

November 2014

September 2014

August 2014

July 2014

March 2014

February 2014

January 2014

October 2013

September 2013

January 2013

December 2012

November 2012

October 2012

September 2012

August 2012

July 2012

- It is possible that a developer provides a map file that the author has to import in RoboHelp. The map numbers still need to be mapped to the correct topics.
- When the author has created context sensitive help in RoboHelp, the developer can use the project map files (files with the .h or .hh extension) and the alias file (.ali file) to import the context sensitivity in the application.

Context Sensitivity and APIs

RoboHelp provides several APIs to help you call the WebHelp. These APIs are pieces of example code to help developers build contextual help calls into the application.

When you call WebHelp, you normally call the start file of the output. To make the call context sensitive, you provide one or more parameters in the URL to allow the WebHelp to display the correct topic. The APIs provided with RoboHelp are shortcuts to building these URLs.

You can use the default APIs provided by Adobe but I found that at least the default JavaScript API does not provide you with all the available options that WebHelp offers.

In this article I list the methods and options you can use for calling WebHelp. This allows you to create your own URLs to call the WebHelp and will allow you to create your own APIs.

Calling WebHelp for Merged

Projects

If you use merged projects, you can simply use the methods described in these articles. If the project cannot find the topic id or map number in the master project, the WebHelp will cycle through all the child projects until it finds the correct topic. You do not need to merge map files in the master project or anything like that.

Calling Dynamic User Centric Content

Dynamic User Centric Content (DUCC) allows you to define different sets of help for different types of readers. If you use DUCC, you need to know which start file you want to call. Apart from the output folder you need to choose, calling WebHelp context sensitive is the same for regular WebHelp and WebHelp with DUCC.

If and only if you use Dynamic User Centric Content, you have two options for calling the help:

1. Calling the start file of the project. This will automatically open the default content category.
2. Calling the desired content category directly.

For example, you have a project with two Content Categories, A and B. The folder structure of your output is as follows:

- Output folder
- index.htm
- A

- index.htm
- B
- index.htm

To call the main category, you can simply use the URL:

```
http://localhost/Output folder/index.htm
```

To call the content category A, use the URL:

```
http://localhost/Output folder/A/index.htm
```

To call the content category B, use the URL:

```
http://localhost/Output folder/B/index.htm
```

The rest of the call is the same as for regular WebHelp.

Methods for Calling WebHelp

RoboHelp offers three separate start files that you can use to call the WebHelp:

- The regular project start file, such as `newproject.htm`.
- The `_csh` start file, such as `newproject_csh.htm`.
- The `_rhc` start file, such as `newproject_rhc.htm`.

Every start file expects an URL with specific options.

The regular start file is all that you need and supports context sensitive help by using topic id's, map numbers and URLs. The other start files are legacy start files that may still be useful.

Every start file offers multiple ways of calling the WebHelp contextually. There are a total of five methods you can use to show the help topic you want. Note that there is no one start file that supports all the available methods.

- By using WebHelp internal id's. (I don't know why you would want to do this, but it can be used.)
- By using the Map number you have set in the RoboHelp project.
- By using the topic id's you have set in the RoboHelp project.
- By specifying the relative path to the topic from the project root.
- By specifying a file inside the project directory that doesn't need to be in the project. (Files you add after generating the WebHelp.)

In the remainder of this article I will describe every start file, its options and the methods it supports. I will first describe the regular start file as that is probably all you need.

Regular start page

The regular start file is used when you call the start file of the project, such as `newproject.htm`. This start file supports the following methods:

- Map numbers.
- Topic id's.

- Relative path to a file in the project.
- Relative path to a file in the project folder but not in the project.

This start file supports the following options:

- Specifying the window you want to use.
- Specify whether you want the help to open in a separate popup.

The regular start file is very flexible since you can use windows (*RoboHelp > Project Set-up > Windows*) to determine how your help will look: showing or hiding the navigation pane, showing or hiding the toolbar and setting the browser dimensions. I assume you are familiar with windows. One note: Always deselect Use Default Browser Settings in the window definition or the window will not work in Firefox. If you do not want to use a window definition at all but just open the help, consider using the `_csh` start file.

This start file supports two methods of calling the WebHelp. The first method is the regular method that is used by the default RoboHelp JavaScript API. The other method is an internal method that the WebHelp uses when you call the WebHelp using the regular method: When you call the help using the regular method, the WebHelp reads the window definition and other settings and redirects using the internal method.

Calling WebHelp

Say that we have a fictitious context sensitive WebHelp project published on

`http://localhost/newproject.htm`. The file `newproject.htm` is the start file of the project. In

this section I will show you how to create a URL to call this WebHelp project.

Default method

To trigger the context sensitivity, add #< after the path of the project start file:

```
http://localhost/newproject.html#<
```

After these characters, you can specify the options. You have to separate multiple options with >>. You can use the following options:

OPTION	EXPECTED VALUE	DESCRIPTION
id	Map number	Call using a map number. If you specify both a map number and a topic id, the help shows the topic corresponding with what is listed first in the URL.
str	Topic id	Call using a topic id. The topic id is case sensitive. If you specify both a map number and a topic id, the help shows the topic corresponding with what is listed first in the URL.
url	Path to file	A file in the project directory. This will show the file without the skin. The file may be a topic from the project or

		another file. The path must be relative from the project root.
topicurl	Path to file	A file in the project directory. This will show the file and the skin. The file may be a topic from the project or another file. The path must be relative from the project root.
wnd	Window name	The name of the window you want to use. The window name is case sensitive. You need to create the window in the RoboHelp project.
cmd	Pane code	The pane to open: toc for TOC, idx for index, fts for search, glo for glossary and none for the navigation pane hidden (but the toolbar is visible).
newwnd	Boolean	Set to false to open the help in the same window (do not create a popup). Set to true to create a popup. A popup is created by default. RoboHelp 8 and higher only.
java	Boolean	Set to false to open the help in the same window (do not create a popup). Set to true to create a popup. A popup is created by default. RoboHelp 7 (and possibly below) only.

Note the following:

- If you use the option `cmd`, you cannot use context sensitivity. The project will always open the default topic even if you specify an `id` and a `window`.
- You can only specify a window if you use `id`, `str`, `url` or `topicurl`.
- You may define the options in any order you want.
- The Firefox default popup blocker blocks the help popup for WebHelp. If you support Firefox, consider setting `newwnd=false`.

For example, if you want to call the topic with map number 3 without creating a popup, use the following URL:

```
http://localhost/newproject.htm#<id=3>>newwnd=false
```

Another example: if you want to call the topic with topic id MyTopic in a popup and using the window NoNavPaneThis is not a default RoboHelp window. You need to create all windows yourself in the RoboHelp project., use the following URL:

```
http://localhost/newproject.htm#<str=MyTopic>>newwnd=false
```

Using a window and not opening in a popup

When you use a window definition, you may expect to be able to use `>>newwnd=false` to open the help in the same browser window:

```
http://localhost/newproject.htm#<str=MyTo  
http://localhost/newproject.htm#<str=MyTo
```

to open the help in the same browser window. In RoboHelp 8 and 9, you can't just use

```
>>newwnd=false
```

 because of a small bug in WebHelp.

To fix this bug, you need to modify the file

```
whcsh_home.htm
```

 as follows:

1. Open `whcsh_home.htm` in a text editor or HTML editor.
2. Go to the function `SwitchURL();`
3. Find the second instance of `if(gsWindow)` on line 210 (RoboHelp 8).
4. Change the if statement `if(gsWindow)` to `if(gsWindow && gsNewWnd == "true"):`

```
else  
{  
    if(gsWindow && gsNewWnd == "true");//This  
    {  
        var nTop=getSValue(oWnd.sBTop,screen.  
        var nLeft=getSValue(oWnd.sBLeft,screen.  
        var nWidth=getSValue(oWnd.sBWidth,scr  
        var nHeight=getSValue(oWnd.sBHeight,s  
        window.resizeTo( nWidth , nHeight);  
        window.moveTo(nLeft , nTop);  
    }  
    document.location=sURL;  
}
```

5. Save changes and close `whcsh_home.htm`.

You can now use a window definition and open the help in the browser window instead of opening in a popup.

Internal method

To trigger the context sensitivity, add `#` or `#>>` after the path of the project start file:

```
http://localhost/newproject.htm#  
http://localhost/newproject.htm#>>
```

Then add the relative path to the topic you want to open:

```
http://localhost/newproject.htm#mydir/myt
```

The difference between `#` and `#>>` is that the first method expects a relative path to a topic. If you do not want to specify a topic, you must add `#>>` after the URL. If you only add a hash, the WebHelp will think that the first option you specify is a relative path to a topic. This means that the `>>` directly after the `#` is optional when you specify a relative path.

Note: If you do not specify a relative path, the project opens the default topic.

After specifying the relative path, you can specify the options. You have to separate multiple options with `>>`. You can use the following options:

OPTION	EXPECTED	DESCRIPTION
--------	----------	-------------

	VALUE	
cap	string	Set the title of the help project.
cmd	number	Sets the default pane that is shown: 0 for no pane, 1 for the TOC, 2 for the index, 3 for the search and 4 for the glossary.
pan	number	Determine whether the navigation pane and the toolbar are shown: 1 to show only the topic (navigation pane and toolbar hidden) or a 2 to show the toolbar and navigation pane. The script only checks for a 1 or any other number. The scripts in the WebHelp itself use 2.
pbd	string	Sets the default pane that is shown. This parameter only works when use the regular method. Use cmd instead of this function. The available settings are: toc for the TOC, ndx for the index, nls for the search and gls for the glossary.
pbs	string	Available panes set by the window definition. This parameter has no effect on the WebHelp.
pot	number	Holds the window option. The only option I could find: If set to 1, it disables browse sequences in the minibar.

For example:

```
http://localhost/newproject.htm#mydir/myt  
http://localhost/newproject.htm#>>mydir/m
```

Note: When you want to call the relative path, you must specify the relative path before you specify any options. Otherwise the help will not open the correct topic.

This method for context sensitivity uses the following files:

- Start file
- whstart.j
- whcsh_home.htm
- whcshdata.htm

_csh start file

For the first legacy start file, you need to call the file `_csh start file`, such as `newproject_csh.htm`. This start file supports the following methods:

- Map numbers.
- Topic id's.
- Relative path to a file in the project.
- Relative path to a file in the project folder but not in the project.

This legacy method is useful if you do not want to use window definitions and simply want to open the help in the browser.

Note: I've seen queries on forums wondering where the _csh start file is in RoboHelp 9. It's still there, but there is a catch if you are using Dynamic User Centric Content (DUCC). If you are using DUCC, RoboHelp generates multiple instances of the help, one for each content category. The main folder only contains a start file that loads the default content category. The _csh file is not available in the main folder, but is available in every subfolder holding a content category.

To trigger the context sensitivity, add `#` after the path of the start file:

```
http://localhost/newproject_csh.htm#
```

After specifying the relative path you can specify the context sensitivity. You can use the following options:

OPTION	EXPECTED VALUE	DESCRIPTION
empty, add value after hash	Map number	Call using a map number.
topicnumber	Map number	Call using a map number.
empty, add value after hash	Topic id (must be alphanumeric or it will be seen as a map	Call using a topic id.

	number)	
topicid	Topic id	Call using a topic id.
remoteurl	Path to file	A file in the project directory. This can be a topic from the project or another file. The path must be relative from the project root.
,newwnd	Boolean	Show or hide the navigation pane.

The toolbar and navigation pane are hidden by default.

If you want to show the skin, add

`,withnavpane=true` to the URL:

```
http://localhost/newproject_csh.htm#3,withnavpane=true
http://localhost/newproject_csh.htm#MyTopic
```

```
http://localhost/newproject_csh.htm#topicid
http://localhost/newproject_csh.htm#topicid
```

If you want the help to show in a popup, you have to add window options. You can use the options available in the JavaScript [window.open](#) method.

Note: The Firefox default popup blocker blocks the popup for local WebHelp.

```
http://localhost/newproject_csh.htm#topicid
http://localhost/newproject_csh.htm#topicid
```

This method for context sensitivity uses the following files:

- `_csh` start file (Legacy start file)
- `cshdat_webhelp.htm`

`_rhc` start file

For the second legacy start file, you need to call the `_rhc` start file, such as `newproject_rhc.htm`. This start file supports the following methods:

- Map numbers.
- Internal WebHelp ids.
- Relative path to a file in the project.
- Relative path to a file in the project folder but not in the project.

I added this legacy method for the sake of completion. I recommend using one of the other methods because they both provide context sensitivity using topic ids.

To trigger the context sensitivity, add `#` after the path of the start file:

```
http://localhost/newproject_rhc.htm#
```

After specifying the relative path you can specify the context sensitivity. You can use the following options:

OPTION	EXPECTED VALUE	EFFECT
context	Map	Call using a map number.

	number	
HelpIdFromHTMLMap_	Map_ number	Call using a map number. If you use this option, add the map number directly after the string. Do not add an equals (=) sign.
topicnumber	Internal id	Call using an internal WebHelp id.
empty, add value after hash	Internal id	Call using an internal WebHelp id.
remoteurl	Path to file	A file in the project directory. This can be a topic from the project or another file. The path must be relative from the project root.
withnavpane	Boolean	Show or hide the navigation pane.

Note: The internal id is assigned only in the WebHelp output. This id is not available in the RoboHelp project.

Example:

```
http://localhost/newproject_rhc.htm#HelpI
http://localhost/newproject_rhc.htm#conte
```

The toolbar and navigation pane are hidden by default. If you want to show the panes, add

`,withnavpane=true` to the URL:

```
http://localhost/newproject_rhc.htm#conte
```

If you want the help to show in a popup, you have to add window options. You can use the options available in the JavaScript [window.open](#) method.

Note: The Firefox default popup blocker blocks the popup for local WebHelp.

```
http://localhost/newproject_rhc.htm#conte  
http://localhost/newproject_rhc.htm#conte
```

This method for context sensitivity uses the following files:

- [_rch start file](#)
- [cshdat_robohelp.htm](#)

Posted in [In-depth](#), [RoboHelp](#) and tagged [Context Sensitive Help](#), [CSH](#), [RoboHelp](#).

Share

Webinar Overview

Context Sensitivity in Responsive and Multiscreen

HTML5

[About](#) [Services](#) [Blog](#) [Contact](#) [Terms](#) [Privacy](#)



© Help Essentials