

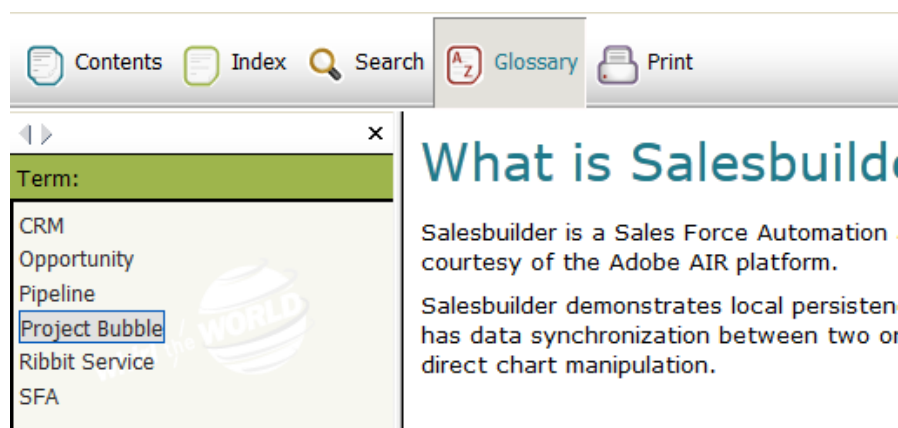
This is an archived version of Help Essentials and will no longer be updated. Downloads and purchases no longer work. Visit [Grainge.org](https://Grainge.org) to access downloads. ✕

[Shop](#) [About](#) [Services](#) [Blog](#) [Contact](#) [My Account](#) [Basket](#) <sup>0</sup>

# Adobe RoboHelp: Change WebHelp Glossary Highlight

By Willam van Weelden on 1st August 2012

When you click a term in the glossary, the selected term is highlighted by a gray background. With some changes, you can style the highlight any way you like. To do this, you have to modify your output files. This means that you need to copy the modified files in your output every time you generate your project.



To change the highlight, you have to change whutils.js, whghost.js and whgbody.htm. First, open whutils.js and search for the function `HighlightElement`. Once you found it, copy the function `HighlightElement` and the function `resetHighlight`. Paste the functions below the current functions and rename the

## 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](#)

copied functions by adding “Glos” to the function name:

```
function HighLightElementGlos(obj,sHighLi
function resetHighLightGlos(sNormalColor)
```

On the fifth line of the function

```
HighLightElementGlos , change the text
resetHighLight to resetHighLightGlos .
```

Before you can add the markup for the highlight, you need to address the correct element. The standard element (obj) of the function is not the element you can use to style the text. To style the text, you need to address the first child of the element `obj` . Add a new variable below the line

```
obj.style.backgroundColor=sHighLightColor;
```

:

```
obj.style.backgroundColor=sHighLightColor
var TextElem = obj.firstChild;
```

Now add any markup you want below the second if. Make sure that all markup you place in the function `HighLightElementGlos` is undone in the function `resetHighLightGlos` . When you want to change the text, start your command with `TextElem` , otherwise start with `obj` :

```
function HighLightElementGlos(obj,sHighLi
{
if(obj!=null)
{
```

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

```
resetHighLightGlos(sNormalColor);
if (obj.style) {
  obj.style.backgroundColor=sHighLightCo
  obj.style.display= "block" ;

  var TextElem = obj.firstChild ;
  TextElem.style.color= "#fff"
}
goHighLighted=obj;
}
}

function resetHighLightGlos(sNormalColor)
{
if(goHighLighted!=null)
{
  if (goHighLighted.style) {
    goHighLighted.style.backgroundColor=sN

    var TextElemReset = goHighLighted.firs
    TextElemReset.style.color="#2c2e74"
  }
  goHighLighted=null;
}
}
```

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

**Note:** To change any style, you have to use the JavaScript syntax. You'll find the syntax for every element on the [CSS reference site at w3schools](#).

Close whutils.js and open whgbody.htm. Find `setBackgroundcolor` and set the standard background colour for your glossary items. Next, go to `setActiveBgColor` and set the highlight colour for your glossary. Save your changes.

Open whghost.js and find the function `HighLight`. In the function `HighLight`, replace `HighLightElement` by `HighLightElementGlos`. Then find `gsBgColor` (a variable at the top of the script). Give this variable the same value as you gave `setBackgroundcolor` in whgbody.htm. Now find `gsABgColor` and give this variable the same value as you gave `setActiveBgColor` in whghost.js.

Find the function `HighLight` and on the fifth line of the function, change `HighLightElement` to `HighLightElementGlos`.

Save the file and make a backup of whutils.js, whgbody.htm and whghost.js. Copy these files in your output every time you generate your project.

---

Posted in [JavaScript](#), [Output manipulation](#) and tagged [JavaScript](#), [RoboHelp](#), [WebHelp](#).

---

Share

---

[Adobe RoboHelp: Change WebHelp Index Highlight](#)

[Adobe RoboHelp: Change WebHelp TOC Highlight](#)

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

© Help Essentials



