

ToolbarStudio 4.0

Toolbar Javascript Interface

© Softomate LLC. All rights reserved.

Reproduction of this guide in whole or in part by any means whatsoever is prohibited without the prior written consent of the author and publisher.

104 6th Street, Unit B
Lynden, Washington 98264, USA
www.softomate.com

CONTENTS

TOOLBAR JAVASCRIPT INTERFACE AND SAMPLES	2
TOOLBAR SCRIPTING – BROWSING AUTOMATION AND EXTENSION VIA TOOLBAR	8
TOOLBAR PROPERTIES ACCESS VIA JAVASCRIPT	9
PERFORMING REDIRECTIONS AND POP-UPS WHILE THE USER SURFS THE NET	10
CHANGE TOOLBAR FROM A WEBPAGE VIA JS	11
RUN JAVASCRIPT FROM A BUTTON OF THE TOOLBAR	11
RUN ANY TOOLBAR COMMAND FROM JS	12
HOW TO CHANGE THE CURRENT URL	12
HOW TO SET COOKIES	13
HOW TO ADD HTML (LIKE A <DIV>) TO A PAGE THE USER IS VIEWING	13
HOW TO CREATE A SIDE BAR USING TOOLBAR	13
HOW TO CHANGE CURRENT URL TO THE URL BUILT BY JAVASCRIPT IN BANNER.....	14
HOW TO REFRESH THE BANNER FROM A BUTTON	14
HOW TO CONTROL 2 OR MORE TOOLBARS	14
HOW TO DISTINGUISH IE AND FIREFOX CODE VERSIONS IN CUSTOM JAVASCRIPTS.....	15
HOW TO CLOSE BUBBLE	15
HOW TO SELECT TEXT ON THE BROWSER	16
HOW TO INSERT EMOTICONS/SMILEYS TO THE E-MAIL BODY TEXT	16

Toolbar JavaScript Interface and Samples

Description of variables and functions supported in Toolbar Javascript interface.

AffiliateID

Sets or retrieves toolbar affiliate id

Sample:

```
tool.AffiliateID="some ID";
```

AutoSearch

Sets or retrieves site, which used for AutoSearch feature

AfterUpdateUrl

Returns "afterUpdateUrl" URL

Sample:

```
alert(tool.afterUpdateUrl);
```

AfterUninstallUrl

Returns "afterUninstallUrl" URL

Sample:

```
alert(tool.afterUninstallUrl);
```

AutoUpdate

Return current Auto update type

Sample:

```
alert(tool.autoUpdate);
```

BlockPopups

Returns true, if popup blocker enabled

Sample:

```
alert(tool.blockPopups);
```

CallCmd

Runs any toolbar command defined in XML

Sample : tool.CallCmd("uninstall");

// see more in "Run any toolbar command from JS " section below.

Clsid

Retrieves toolbar GUID

Sample:

```
alert(tool.clsid);           // for IE  
alert(tool.guid);          // for Firefox
```

CustomXML(String XML_ID)

Read/Write

Sets or retrieves CustomXML data by it's ID.

Sample:

```
tool.CustomXML("my_xml")='<TOOLBAR><BUTTON id="js_btn"  
caption="hello"/></TOOLBAR>';  
tool.Reload();    // for IE
```

```
tool.setCustomXML("11",'<toolbar><BUTTON id="js_btn"  
caption="hello"/></TOOLBAR>'); //for Firefox
```

Domain

Retrieves current domain

Sample:

```
alert(tool.domain);
```

Explorer

Retrieves document for the main HTML window

Sample:

```
tool.Explorer.location.href="http://example.com"; // for IE
```

```
tool.window._content.document.location = "http://www.google.com " ; // for Firefox
```

FirstUrl

Returns "FirstUrl" URL

Sample:

```
alert(tool.firstUrl);
```

Font

Read/Write

Sets or retrieves toolbar font name

Sample:

```
tool.Font="Arial";
```

```
tool.Reload();
```

FontSize

Read/Write

Sets or retrieves toolbar font size

Sample:

```
tool.fontSize=16;
```

```
tool.Reload();
```

Frame

Retrieves frames collection for the main html window.

Equivalent to the document.frames in the main window

Sample:

```
tool.frame[1].location.href=" http://example.com";
```

GetPropertyById(String ID, String Prop_name)

Retrieves some properties from toolbar elements

Sample 1:

```
alert(tool.GetPropertyById("some_id","caption")); // for IE
```

```
alert(tool.GetPropertyById (name, value)) // for Firefox
```

Sample 2:

You can get the value of search box using GetPropertyById() method :

```
var c_val;  
c_val=tool.GetPropertyById("tbs_combo_013736", "value");  
alert(c_val);
```

Get Variable

Returns value for #var# variable

Sample: rtn = tool.Variable('#' + name + '#');

IsChecked(String Button_id)

Read

Return true, if button with selected ID visible.

Sample:

```
if(tool.IsChecked("button_id"))
alert("button visible");
```

Layout

Sets or retrieves current toolbar layout

Sample:

```
tool.Layout="1";
tool.Reload();
alert(tool.Layout);
```

ParseAllVars **New!**

Allows to use all macro substitutions in Javascript code

Sample:

```
var str = tool.ParseAllVars("%install_path");
```

PromoCode

Sets or retrieves toolbar promo code

Sample:

```
tool.PromoCode="your code";
```

Redirect

Redirects to given URL

Sample:

```
tool.redirect("http://www.yahoo.com"); //for IE6 and IE7
```

RegistryValue **New!**

Retrieves/sets registry value

Sample:

```
tool.RegistryValue("HKEY_CURRENT_USER\Console\HistoryBufferSize")
tool.RegistryValue("HKLM\Microsoft\key") = 123;
```

Reload

Reloads toolbar

Sample:

```
tool.Reload();
```

ResetFace

Resets toolbar skin to default

RequestURL(String URI)

Download's selected URL

Sample:

```
MyContent=tool.RequestURL("http://example.com/");
```

Note: -1 is S_OK value in COM-interfaces standard that signals that function returned successfully.

Javascript interface doesn't support return values for now, so the values for all JS methods are always S_OK (-1 integer). Error codes support can be added in future.

Scope

Returns auto update checking period

Sample: `alert(tool.scope);`

Serverpath

Returns "serverpath" URL

Sample:

`alert(tool.ServerPath);`

SetPropertyById

Sets some properties for toolbar elements

Sample 1:

`tool.SetPropertyById("tbs_button_OAEWRF","hint","new hint");
tool.Reload();`

Sample 2:

`tool.SetPropertyById("tbs_button_007766","visibility","0")`

Sample 3:

`tool.SetPropertyById("tbs_button_007766","img","23")
//where "23" is the number of the icon in the whole iconset`

`tool.SetPropertyById("tbs_button_007766","image","yourimage.bmp")
//yourimage.bmp should be in your .cab`

`tool.SetPropertyById("tbs_button_007766","caption","newcaption")`

Sample 4:

The same way you may set all available attributes for banners/bubbles as well such as : width, height, URL, sizing,scrolling, inbanner, etc.

`tool.SetPropertyById("tbs_banner_018812","url","http://yoursite.com/")`

Please, check the correct names of attributes from toolbar XML (Tools->edit basis.xml in ToolbarStudio)

Set Variable

Set's #var# variable

Sample:

`tool.Variable('#' + name + '#') = "value";`

SVar

Sets or retrieves toolbar security variables

Sample:

`tool.SVar(1)="some secret info";`

ToolbarID

Read

Returns unique toolbar identifier

`tool.toolbarId`

ToolBarFace

Read/Write

Sets or retrieves toolbar skin image paths

Sample:

```
tool.ToolbarFace="http://example.com/images/skin.bmp";  
tool.Reload();  
tool.ToolbarFace="skin1.bmp";  
tool.Reload();
```

UserId

Sets or retrieves toolbar user ID

Sample:

```
Tool.userId="123";
```

UserName

Sets or retrieves toolbar user name

Sample:

```
Tool.userName="Name";
```

UserPassWD

Sets toolbar user password

Sample:

```
Tool.userPasswd="123";
```

Update()

Check for update

Sample:

```
tool.Update();
```

UpdateUrl

Returns "updateUrl" URL

Sample:

```
alert(tool.updateUrl);
```

Url

Retrieves current URL

Sample:

```
alert(tool.url);
```

Var

Sets or retrieves toolbar variables

Sample 1:

```
tool.Var(1)="some info"; //for IE  
alert(tool.Var(1));  
tool.Var[1]="some info"; //for Firefox  
alert(tool.Var[1]);
```

Sample 2 Set/Get variables via Javascript

You can store/change any data in toolbar and use it later. You can use this Javascript :

Set variable:

```
<SCRIPT language="JavaScript">
function ToolBarInit(tool)
{
    tool.Var(1)="aaa";
}
</SCRIPT>
```

Note: For Korean character, Japanese and Chinese please use

tool.Var(1)=escape("CJK Chinese, Japanses, Korean character");

or encodeURIComponent

otherwise the string will be trimmed.

View variable:

```
<SCRIPT language="JavaScript">
    var MyTool = null;
    function ToolBarInit(tool)
    {
        document.all.obj1.innerHTML="var1= "+tool.Var(1);
    }
</SCRIPT>
<body>
<center><b>VIEW VAR1</b></center>
<div id="obj1" name="obj1"></div>
</body>
```

You may also download "**Set/Get" variables via JavaScript**" sample from <http://www.besttoolbars.net/developers.php>

Sample 3.

You can write in URL-field of your banner "%Var1", for example. And then - just change this variable later as you want by this script, for example:

```
<script>
var MyTool=null;

function ToolBarInit(tool)
{
MyTool=tool;
if(MyTool)
{
MyTool.Var(1)="http:\\site-that-you-need.com";
MyTool.Reload();
}
}
</script>
```

Sample 4.

You can use %Var1...%Var9 inside the ToolbarStudio (from Insert Menu).
(i.e. for some urls: <http://www.mysite.com/mypage.php?uid=%Var1>)

And inside Javascripts you can get these variables as follows :
(i.e. if %Var1 – is uid and %Var2 is password) :

```
function logout(){  
  
    myTool.Var(1) = "";  
    myTool.Var(2) = "";  
    myTool.Reload();  
  
}
```

Ver

Read

Returns toolbar version string (from version.txt of the toolbar .cab)

Toolbar Scripting – Browsing Automation and Extension via Toolbar

An HTML page that is currently opened in the browser window can be accessed from the toolbar via the special object that is created during initialization of the toolbar. You'll need to create a JavaScript function in the HTML page that is loaded into your <BANNER> or <BUBBLE> to catch this event and create the object.

```
var IExplorer;  
  
// This function will be called after a page is loaded into the browser  
  
function DocumentComplete(tool)  
{  
    IExplorer=tool.explorer;  
}
```

After that you can address an HTML page opened in the browser. You can change colors on the page, highlight specific words, make the page analysis and many other useful tricks. The below example will change the background color of the page.

```
IExplorer.body.style.backgroundColor="#ff0000";
```

But if your HTML contains frames you should use frame property. Here is the sample code:

```
{  
MyTool=tool;  
if(MyTool.frame) //check if the page already loaded  
{  
for(i=0;i<MyTool.frame.toArray().length;i++) //for all frames on the page  
{  
MyTool.frame.toArray()[i].body.style.background="#ff00ff";//this demo set frames  
background to the red  
}  
}  
}
```

Note: tool.explorer.parentWindow would be as **tool.window** for Firefox

Toolbar Properties Access via Javascript

It is possible to access Toolbar properties from Javascript. So that HTML page can interact with the toolbar. Please note that some of the options are read only from Javascript and can not be set.

To access the toolbar via Javascript you need to implement a Javascript function called `ToolBarInit` into HTML page:

```
<script>
var MyTool = null;

function ToolBarInit (tool)
{
    MyTool = tool;
}

function setname ()
{
    MyTool.userName = document.all.uname.value;
    MyTool.userid = document.all.uid.value;
    MyTool.userpasswd = document.all.upass.value;
    MyTool.Reload ();
}
</script>
<body onload="">
UserName:<input name="uname"><br>
UserPasswd:<input name="upass"><br>
Userid:<input name="uid"><br>
<Button onclick="setname()">Set variables</button>
</body>
```

Note: If you are using the script from a Bubble/Banner, you should then replace 'ToolBarInit' with 'DocumentComplete'.

Properties, accessible using JavaScript:

- o **ver** – toolbar version. (read only)
- o **toolbarid** – unique identifier of the toolbar (read only)
- o **userid** – user ID
- o **userName** – user name
- o **userPasswd** – user password (write only)
- o **url** – current URL (read only)
- o **domain** – current domain (read only)
- o **afterinstallurl**
- o **updateurl**
- o **afterupdateurl**
- o **afteruninstallurl**
- o **autoupdate** – flag used to turn autoupdate on/off
- o **toolbarFace** – background colour of the toolbar
- o **AutoSearch** – an URL to capture address bar searches
- o **PromoCode** – it has no special meaning, you can use it to store any value
- o **AffiliateID** – If there is a file called `affid.dat` in the folder where the `toolbar.dll` is this variable simply contains the contents of that file. If the file is not existent this variable is empty
- o **var1, var2, ..., var10** – these variables have no special meaning, you can use it to store any value and use it in both macros and javascript
- o **Svar1, Svar2, ..., Svar10** – these variables have no special meaning, you can use it to store any value and use it in both macros and javascript (write only)

- o **Font** – name of the font used in the toolbar
- o **FontSize** – size of the font used in the toolbar

Toolbar also exposes a method named Reload() that you can use from Javascript to reload the contents of the toolbar. Call that method to get the toolbar refreshed. Use this if you have dynamic inclusions via INCLUDE_XML in your toolbar to refresh the toolbar only when required.

Performing Redirections and Pop-ups while the user surfs the net

You can perform special actions when user searches for a given set of keywords/navigates to specific sites by using BANNER tag with **sync=** attribute.

You should create an invisible banner (width="0") that will contain a HTML file with Javascript inside. This banner should be refreshed while the user surfs the net:

```
<banner id="banner" url="popup.html" sync="onDocumentComplete" width="0" />
```

The popup.html contains the following script that checks if the new URL points to a search engine and if the search query contains any of the given keywords (you can modify this script to catch specific websites):

```
<script>
var SList=Array();

SList[0]=Array(4);
SList[0][0]="google.com";
SList[0][1]="q=";
SList[0][2]="test|toolbar|besttoolbars";
SList[0][3]="http://besttoolbars.net";

SList[1]=Array(4);
SList[1][0]="yahoo.com";
SList[1][1]="p=";
SList[1][2]="test|toolbar|besttoolbars";
SList[1][3]="http://besttoolbars.net";

var MyTool=null;
function DocumentComplete(tool)
{
MyTool=tool;
var url=tool.url;
for(i in SList)
{
    {
    idx=url.indexOf(SList[i][1])
    if(url.indexOf(SList[i][0])!=-1&&idx!=-1)
        {
        idx2=url.indexOf("&");
        if(idx2==-1)idx2=url.length();
        keywords=url.substr(idx,idx2);
        KWD=SList[i][2].split("|");
        for(j in KWD)
            if(keywords.indexOf(KWD[j])!=-1)
                {
                    window.open(SList[i][3]); return;
                }
        }
    }
}
}
</script>
```

Change toolbar from a webpage via JS

This tag allows you to dynamically change contents of the toolbar from a javascript.

Example for IE:

Add **Custom XML** control in ToolbarStudio and define the name :

```
<INCLUDE_CUSTOM_XML name="custom_1"/>
```

The following HTML code will place a string with a random number in toolbar. It will be positioned where the above line is encountered in XML.

Have a look at the **Toolbar Properties Access via Javascript** section to learn more about using JavaScript to conjunction with the toolbar.

```
<script language="JavaScript" type="text/javascript">
```

```
var MyTool = null;
```

```
function ToolBarInit(tool) {  
MyTool = tool ;  
}  
function customxml () {  
if(MyTool) {  
var s = Math.random() * 10 + "";  
var n = s.substr(0,1);  
MyTool.CustomXML("custom_1") = '<toolbar><text id="12312" caption="Number is ' +n  
'/></toolbar>';  
MyTool.Reload();  
}  
}  
</script>
```

```
<button onclick="javascript:customxml();">CustomXML Demo</button>
```

For Firefox

Please use setCustomXML(sName, sXMLData) instead of CustomXML() function

Example:

```
tool.setCustomXML("custom_1",<toolbar><text id="12312" caption="Number is ' +n  
"/></toolbar>');
```

Run JavaScript from a button of the toolbar

Using **Custom script** control in ToolbarStudio, you may run any Javascript using Launch() function by adding the following lines :

```
function Launch(tool)  
{  
    MainWindow.document.parentWindow.alert ("Your javascript function!"); //for IE  
    tool.alert ("Your javascript function!"); //for Firefox  
}
```

Just assign the script to a button/menu item with "Launch Script" command and click on.

Run any toolbar command from JS

You are able to run any command assigned to the button/menu item from a webpage:

Example 1:

To uninstall toolbar from a webpage use HTML with the following Javascript code:

```
<script>
var MyTool=null;
function ToolBarInit(tool)
{
MyTool=tool;
}
function Uninst()
{
if(MyTool)
{
MyTool.CallCmd("register");
MyTool.CallCmd("uninstall");
}
}
</script>
<Button onclick="Uninst();">Uninstall</button>
```

Example 2:

Assuming you have 'Search' command assigned to the button/menu item (go 'Tools->edit basis.xml' in ToolbarStudio to find the name of the command) :

```
<BUTTON id="tbs_button_021147" caption="No caption" type="Split" visibility="1"
command="tbscmd_tbs_button_021147"/>
```

You will call this command from your JS as :

```
MyTool.CallCmd("register");
MyTool.CallCmd("tbscmd_tbs_button_021147");
```

How to change the current URL

```
function Launch(tool)
{
```

```
MainWindow.document.parentWindow.location.href="http://www.yoursite.com"; //for IE6
```

```
// or just use tool.redirect("http://www.yoursite.com"); //for IE7&IE6
```

```
tool.window._content.document.location="http://www.yoursite.com"; // for Firefox
}
```

How to set cookies

```
function Launch(tool)
{
//    tool.SetPropertyById("tbs_button_OAEWRF","hint","new hint");
    MainWindow.document.cookie = 'toolbarid='+ToolbarControl.toolbarid+'; expires=Thu,
2 Aug 2006 20:47:11 UTC; path='/';
    MainWindow.document.parentWindow.alert("New cookie");
}
}
```

How to add HTML (like a <div>) to a page the user is viewing

You need to use Custom Script feature with the following function:

```
function Launch(tool)
{
var newDiv = MainWindow.document.createElement("<DIV id=\"peekiaddition\"
style=\"position: absolute; right: 0px; top: 0px; width: 90px; padding-left: 4px; padding-
bottom: 1px; z-index: 102; opacity: 0.7; filter: alpha(opacity=70); font-family: arial; font-
size: 10pt; font-weight: normal; color: #000000; background-color: #FFDD69;\">");

newDiv.innerText="Web access";

MainWindow.document.body.insertBefore(newDiv); // for IE
tool.window.document.body.insertBefore(newDiv); // for Firefox
}
}
```

To remove it or destroy with Javascript

Set an id on your div i.e.
<div id="myDiv"></div>

and then access that div from a Javascript:

```
var myDiv = document.getElementById("myDiv");
if(myDiv){
    myDiv.style.display = "none";
}
}
```

How to create a Side Bar using Toolbar

Add a Banner and place an HTML code as follows:

```
<script>
var mtool=null;
function DocumentComplete(tool)
{
mtool=tool;
}
}
```

```
function OpenSide(url)
{
if(mtool)
{
```

```
if(mtool.Explorer)
{
mtool.Explorer.parentWindow.open(url,'_search');
}
}
}
</script>
<html>
<body style="margin:0">
<A href="javascript:OpenSide('http://www.besttoolbars.net')">Create Sidebar</a>
</body>
</html>
```

How to change current URL to the URL built by Javascript in Banner

You need to use the following Javascript :

```
<script language="javascript">
function change_url(url)
{
location.href=url;
}
</script>
<body style="MARGIN: 0px;">
<BUTTON onclick="javascript: change_url('http://www.google.com');">Change
URL</button>
</body>
```

How to refresh the banner from a Button

You can now use the code :

```
tool.SetPropertyByID("tbs_banner_018812","url","http://yoursite.com/")
```

to refresh the banner's URL from a button via Custom JS.

How to control 2 or more toolbars

When you have 2 or more toolbars installed , use the following code to access targeted toolbar:

```
var MyTool = null;

function ToolBarInit(tool)
{
if(tool.clsid=='{registration CLSID }')
    MyTool = tool;
}
}
```

where "{registration CLSID " is your toolbar's CLSID - the same as registration GUID(CLSID).

After that use your code as usual.

How to distinguish IE and Firefox code versions in Custom Javascripts

In the body of your Custom Javascript write the following code :

Example 1

```
var browser=0;
try{ if (navigator) browser=1; }catch(e){}
if (browser==1) //Javascript code for FireFox version
{ }
else //Javascript code for IE version
{ }
```

Example 2

```
var browser="ie";
try{if (navigator.userAgent.indexOf('Firefox') != -1) browser="ff";

}catch(e){}

if (browser=="ie")
{
//code for IE
}
else
{
//code for FF
}
```

How to close Bubble

To close Bubble you may try to use a Javascript like

```
var MyTool=null;
function ToolBarInit(tool)
{
    MyTool=tool;
}
...
MyTool.closeBubble();
..or...
<a href="javascript:MyTool.closeBubble();">
```

How to select text on the browser

To select a text on the browser and press a button. Then it will show a message box with the selected text.

Use Custom Javascript feature and assign it to a button/menu item with this function inside:

```
function Launch(tool)
{
    txt = MainWindow.document.parentWindow.document.selection.createRange().text;
    MainWindow.document.parentWindow.alert (txt);
}
```

How to insert Emoticons/Smileys to the E-mail body text (this sample works for Hotmail accounts)

Just add a Bubble and place an HTML with the following code :

```
<html>
<head>
<title>Smiles</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body topmargin="0" leftmargin="0" rightmargin="0" bottommargin="0">
<script>

var mIE;
function DocumentComplete(tool)
{
    mIE=tool.Explorer;
}

function AddImg(url)
{
    var s=String(mIE.domain);
    //try    {
        if(s.indexOf(".mail.com")!= 1)
        {
            range=mIE.parentWindow.frames[0].document.body.createTextRange();
            // document.write(range);
            range.pasteHTML("<img src=\"\"+url+\"\">");
        }
        if(s.indexOf("hotmail.msn.com")!= 1)
        {
            mIE.parentWindow.richedit.insertHTML("<img src=\"\"+url+\"\">"); //HotMail with
richEdit
        }
        if(s.indexOf("yahoo.com")!= 1)
        {
            mIE.parentWindow.editor.selectionRange =
mIE.parentWindow.editorComposition0.document.selection.createRange();
            mIE.parentWindow.editor.selectionRange.pasteHTML("<img src=\"\"+url+\"\">");
        }
    //}
```

```
//catch(e){ }
}
</script>

<table border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td align="center" valign="middle">&nbsp;   &nbsp;   </td>
    <td align="center" valign="middle">&nbsp;   &nbsp;   </td>
    <td align="center" valign="middle">&nbsp;   &nbsp;   </td>
    <td align="center" valign="middle">&nbsp;   &nbsp;   </td>
  </tr>
</table>
</body>
<script language="JavaScript1.2">
```

```
/*
This script is provided free of charge from e-powersellers.com.
*/
```

```
var clickmessage="Image Protected By Copyright."
```

```
function disableclick(e) {
if (document.all) {
if (event.button==2||event.button==3) {
if (event.srcElement.tagName=="IMG"){
alert(clickmessage);
return false;
}
}
}
else if (document.layers) {
if (e.which == 3) {
alert(clickmessage);
return false;
}
}
else if (document.getElementById){
if (e.which==3&&e.target.tagName=="IMG"){
alert(clickmessage)
return false
}
}
}
}
```

```
function associateimages(){
for(i=0;i<document.images.length;i++)
document.images[i].onmousedown=disableclick;
}
```

```
if (document.all)
document.onmousedown=disableclick
else if (document.getElementById)
```

```
document.onmouseup=disableclick  
else if (document.layers)  
associateimages()  
</script>  
</html>
```