

ToolbarStudio 4.2

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.

901 N. Pitt Street, Suite 325
Alexandria, VA 2231
www.softomate.com

CONTENTS

TOOLBAR JAVASCRIPT INTERFACE AND SAMPLES – NEW!	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
USING XMLHTTPREQUEST IN TOOLBAR	14
HOW TO CREATE A SIDE BAR USING TOOLBAR	14
HOW TO CHANGE CURRENT URL TO THE URL BUILT BY JAVASCRIPT IN BANNER.....	14
HOW TO REFRESH THE BANNER FROM A BUTTON	15
HOW TO CONTROL 2 OR MORE TOOLBARS	15
HOW TO DISTINGUISH IE AND FIREFOX CODE VERSIONS IN CUSTOM JAVASCRIPTS.....	15
HOW TO CLOSE BUBBLE	16
HOW TO SELECT TEXT ON THE BROWSER	16
HOW TO INSERT EMOTICONS/SMILEYS TO THE E-MAIL BODY TEXT	16
WIDGET PLUGIN- JAVASCRIPT API – NEW!	18

Toolbar JavaScript Interface and Samples – **New!**

Description of variables and functions supported in Toolbar Javascript interface.

AffiliateID

Sets or retrieves toolbar affiliate id

Sample:

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

AutoSearch //forIE

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);
```

AttachEvent **New! //for IE**

attachEvent and detachEvent methods are now available to bind event handler or disengage the binding through the toolbar object. User's function will be called when a certain event takes place.

Syntax:

```
bSuccess = tool.attachEvent(sEventName, fpNotify);
```

Parameters

sEventName	String. Event name
fpNotify	Function. Function to be called when sEventName takes place

Return value - Boolean

true	Event binding was successful
false	Event binding was Not successful

Sample :

```
tool.attachEvent("ToolbarReload", function() {  
  // Here is the code which will run after toolbar reload complete  
});  
tool.Reload();
```

Events - ToolbarReload

An event will take place after toolbar **Reload**. No need to detach from this event.

AutoUpdate // for IE

Return current Auto update type

Sample:

```
alert(tool.autoUpdate); //for IE
```

BlockPopups //for IE

Retuns 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 FF  
tool.setCustomXML(name,value); //for FF  
tool.getCustomXML(name); // for FF
```

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
```

or

```
tool.redirect //for IE and Firefox
```

FirstUrl

Returns "FirstUrl" URL //after_install.html

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 //for IE

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);
```

IsChecked(String Button_id) //for IE

Read

Return true, if button with selected ID visible.

Sample:

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

Layout //for IE

Sets or retrieves current toolbar layout

Sample:

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

ParseAllVars **New! //for IE**

Allows to use all macro substitutions in Javascript code

Sample:

```
var str = tool.ParseAllVars("%install_path") //for IE
```

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 IE and Firefox
```

RegistryValue New! // for IE

Retrieves/sets registry value

Sample:

```
tool.RegistryValue("HKEY_CURRENT_USER\Console\HistoryBufferSize")
```

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

Reload

Reloads toolbar

Sample:

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

ReloadAllToolbars() New! // for IE

Reloads all toolbars (created in ToolbarStudio)

ResetFace //for IE

Resets toolbar skin to default

RequestURL(String URI) – taken out of platform

Download's selected URL // use **XMLHttpRequest** from standard Javascript

Scope //for IE

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")
```

```
tool.SetPropertyById("tbs_combo_007766","value","newvalue")
```

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/Get Variable

Sets custom variable

Sample:

```
tool.SetVariable("name", "John");  
var nm = tool.GetVariable("name");
```

SVar

Sets or retrieves toolbar security variables

Sample:

```
tool.SetSVar(1, "someSecureVariableForParse");  
var someSecureVariable = tool.GetSVar(1);
```

ToolBarID

Read

Returns unique toolbar identifier

```
tool.toolbarId
```

ToolBarFace //for IE

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();
```

Userld

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.SetVar(1, "someVariableForParse");  
var someVariable = tool.GetVar(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.SetVar(1,"aaa");  
}  
</SCRIPT>
```

Note: For Korean character, Japanese and Chinese please try

```
tool.SetVar(1, escape("CJK Chinese, Japanese, 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.GetVar(1);  
    }  
</SCRIPT>  
<body>  
<center><b>VIEW VAR1</b></center>  
<div id="obj1" name="obj1"></div>  
</body>
```

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.SetVar(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.SetVar(1, "");
    myTool.SetVar(2, "");
    myTool.Reload();
}
```

Working with Big Data and Files New! // for IE

Now you can use :

```
var x = tool.LoadFile("name", "default_value");
tool.SaveFile("name", "value");
```

to save data to Files or load data from Files, in case you need to save a big number of data in registry.

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.toArray()) //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 sets 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 **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()** or **ReloadAllToolbars()** (for IE) that you can use from Javascript to reload the content 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.

Also toolbar allows to store any value in the containers **Var1...Var10** or **SVar1...SVar10**(for secure 'read only' data) using methods of Set/Get Variables. You can use these variables in macros in the URLs (**%Var1..%Var10**) and Javascript

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
```

```
tool.redirect("http://www.yoursite.com"); //for IE&Firefox
```

```
tool.window._content.document.location="http://www.yoursite.com"; // for Firefox 2.0  
}
```

How to set cookies

```
//for IE  
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");  
  
}  
//for Fire fox please use tool.setCookie() and getCookie() functions.
```

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";  
}
```

Using XMLHttpRequest in Toolbar

When adding 'Custom Javascript' control in your toolbar and making XMLHttpRequest , use the following format to create the object :

// IE - all versions

```
var XMLHttpRequest = tool.CreateObj("Msxml2.XMLHTTP");
```

// Firefox

```
var XMLHttpRequest = new XMLHttpRequest();
```

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>
```

Widget Plugin- Javascript API – New!

requires Widget plugin to be added to the toolbar.

XML Format description:

```
<SHOWWIDGET name="widgetwindow__GadgetRX0" template="template.htm"
width="400" height="400">
<PROPERTY key="content"><?[CDATA[ Content goes here ...]]></PROPERTY>
<PROPERTY key="x" value="100"/>
<PROPERTY key="y" value="100"/>
<PROPERTY key="caption" value="No caption"/>
<PROPERTY key="sizeable" value="false"/>
<PROPERTY key="minwidth" value="200"/>
<PROPERTY key="minheight" value="200"/>
</SHOWWIDGET>
```

Attributes

Name	Type	Description
name	string	string to identify the widget window
url	string url	URL that will be opened inside the widget plugin
template	string url	Local HTML file used as frame for window. Contains close button, movable layer, sizeable scroll and other UI;
width	integer	
height	integer	Width of the widget window in pixels
caption	string	Height of the widget window in pixels
x	integer	Position of the widget window in pixels from left side of the screen
y	integer	Position of the widget window in pixels from top side of the screen
position	string	Determine position of the window on the screen;
use_mainwindow	0/1	Determine navigation frame; if use_mainwindow="1" , all links on the widget window will be opened in the main IE window;

Properties

Name	Type	Description
sizeable	bool	If sizeable="true" a small corner picture will appear to be used for sizing widget window
maxwidth	integer	Determine maximal size of widget window in sizeable mode
minheight	integer	Determine minimal size of widget window in sizeable mode

Position

- ? top
- ? bottom
- ? left
- ? right
- ? right-top
- ? left-top
- ? right-bottom
- ? left-bottom

Javascript Api Interface

```
function GetProperty(name, value); // Returns the value of the property by name
function SetProperty(name, value); // Sets the value of the property by name
function SetWindowPos(...); // Changes the position/size of the Widget window on the screen
function ModifyStyle(...); // Changes the styles of the window
function MoveWindow(...); // Changes the position of the Widget windows on the screen
function ResizeWindow(...); // Changes the size of the window
function MakeWindowTransparent(...); // Makes the window semi-transparent
```

```
function Capturemouse(); // Moves the window with a mouse cursor  
function GetWindowRect(...); // Returns the coordinates of the window
```

Example :

```
var oldValue = 300;  
var alwaysOnTopState = false;  
var isMaximized = true;  
var captionSize = 24;  
var isBorder = false;  
var tool = null;  
var ie = true;  
function ToolBarInit(_tool)  
{  
    tool = _tool;  
    ie = false;  
    onReady();  
}  
function GetProperty(name, value)  
{  
    if (tool)  
    {  
        return tool.getProperty(name, value);  
    } else  
        return window.external.GetProperty(name, value);  
}  
function SetProperty(name, value)  
{  
    if (tool)  
    {  
        return tool.setProperty(name, value);  
    } else  
        return window.external.SetProperty(name, value);  
}  
function AlwaysOnTopTriger() {  
    AlwaysOnTop(alwaysOnTopState);  
    alwaysOnTopState = !alwaysOnTopState;  
}  
  
function AlwaysOnTop(b) {  
    try{  
        if (b) window.external.SetWindowPos(-2, 0, 0, 0, 0, 3);  
        else window.external.SetWindowPos(-1, 0, 0, 0, 0, 3);  
    } catch (err) {}  
}  
  
function MinimizeMaximizeTriger() {  
    if (isMaximized) {  
        Minimize();  
    }  
}
```

```
else {
Maximize();
}
isMaximized = !isMaximized;
}
```

```
function BorderTriger() {
if (isBorder) {
BorderOff();
}
else {
BorderOn();
}
isBorder = !isBorder;
}
```

```
function BorderOn() {
try{
//WS_THICKFRAME, SWP_FRAMECHANGED
window.external.ModifyStyle(0, 262144, 32);
} catch (err) {}
return false;
}
```

```
function BorderOff() {
try{
//WS_THICKFRAME, SWP_FRAMECHANGED
window.external.ModifyStyle(262144,0, 32);
} catch (err) {}
return false;
}
```

```
function Minimize() {
if (tool)
{
var size = tool.getSize();
oldValue = size[1];
tool.resize(size[0], captionSize);
} else {
try{
window.external.GetWindowRect(function(left, top, right, bottom) {
oldValue = bottom - top;
if (oldValue < 100) oldValue = 100;
window.external.MoveWindow(left, top, right - left, captionSize);
moveSmoothCorner(right - left, captionSize);
document.getElementById("_content").style.display = "none";
});
} catch (err) {}
}
}
```

```
return false;
```

```
}

function Maximize() {
if (tool)
{
var size = tool.getSize();
tool.resize(size[0], oldValue);
} else {
try{
window.external.GetWindowRect(function(left, top, right, bottom) {
window.external.MoveWindow(left, top, right - left, oldValue);
moveSmoothCorner(right - left, oldValue);
document.getElementById("_content").style.display = "block";
});
} catch (err) {}
}

return false;
}

function Close() {
if (tool)
{
tool.close();
} else
try{
setTimeout("smoothChangeTransparency(220, 0, -10)", 10);
setTimeout("window.external.Close()", 210);
} catch (err) {}
return false;
}

function Capturemouse() {
try{
window.external.Capturemouse();
} catch (err) {}
}

function whichElement(e) {
if (tool)
{
tool.captureMouse(e);
return;
}
var targ
if (!e) var e = window.event
if (e.target) targ = e.target
else if (e.srcElement) targ = e.srcElement
if (targ.nodeType == 3) // defeat Safari bug
targ = targ.parentNode
var tname
```

```
tname = targ.tagName
if (tname != "INPUT") Capturemouse()
}
function smoothChangeTransparency(from, to, step)
{
try
{
from += step;
from = (from > 255) ? 255 : ((from < 0)? 0 : from);
window.external.MakeWindowTransparent(0xEEFF00, from);
if ((step>0 && from<=to) || (step<0 && from>=to))
setTimeout("smoothChangeTransparency("+from+", "+to+", "+step+)", 10);
} catch (err) {}
}

function onReady(){
try
{
//BorderTriger();

var userVarsStr = GetProperty("user_vars", "");
placeWindowButtons(true, userVarsStr, !tool, true);
document.getElementById("_header").innerHTML = GetProperty("caption", "");
if (!tool)
{
AlwaysOnTopTriger();
window.external.MakeWindowTransparent(0x00FFEE, 0);
placeSmoothCorner();
}
if (userVarsStr && (GetProperty("first_start", "true") == "true"))
{
showSettings();
} else {
showContent(false);
}
if (!tool)
{
window.external.Show();
setTimeout("smoothChangeTransparency(0, 220, 10)", 10);
}
} catch (err) {}
}
function prepeareString($0, $1)
{
return GetProperty($1, $1);
}
function prepeareWidget(src)
{
return src.replace(/\[(\w*?)\]/ig, prepeareString);
}
function showSettings()
```

```

{
var userVarsStr = GetProperty("user_vars", "");
if (userVarsStr)
{
var userVars = eval("(" + userVarsStr + ");");
var html = "<center><table>";
for(variable in userVars)
{
html += "<tr><td class='name_field'>" + userVars[variable].screenName + "</td>";
if (userVars[variable].type == "enum")
{
html += "<td><SELECT class='input_field' id='"+variable+"'>";
for(val in userVars[variable].values)
{
html += "<OPTION value='"+val+"'";
if (val == GetProperty(variable, ""))
html += ' SELECTED';
html += ">" + userVars[variable].values[val] + "</OPTION>";
}
html += "</SELECT></TD>";
} else {
html += "<td><input type='text' class='input_field' id='"+variable+"'";
html += "value='"+GetProperty(variable, "")+"'></td>";
}
}
html += "</tr>";
html += "<tr><td colspan=2 align=right>";
html += "<input type=button value='Cancel' class='save_button'";
html += "onClick='javascript:showContent(false);'>";
html += "<input type=button value='Save' class='save_button'";
html += "onClick='javascript:showContent(true);'></td></tr>";
html += "</table>";
document.getElementById("_content").innerHTML = html;
}
}

```

```

function showContent(first_start)
{
if(first_start)
{
var fields = document.getElementsByTagName('input');
for (i=0; i<fields.length; i++) {
SetProperty(fields[i].id, fields[i].value);
}
fields = document.getElementsByTagName('select');
for (i=0; i<fields.length; i++) {
SetProperty(fields[i].id, fields[i].value);
}
SetProperty("first_start", "false");
}
//document.getElementById("_content").innerHTML = "<textarea>" +

```

```
prepeareWidget(GetProperty("widgetpath","")) + "</textarea>";
document.getElementById("_content").innerHTML =
prepeareWidget(GetProperty("widgetpath",""));

}
function onHover(obj, src)
{
obj.src = src;
}

function placeWindowButtons(bMinimize, bSettings, bOnTop, bClose)
{
var buttons = "";
if (bMinimize) buttons += '<span onmousedown="MinimizeMaximizeTriger();"></span>';
if (bSettings) buttons += '<span onmousedown="showSettings();"></span>';
if (bOnTop) buttons += '<span onmousedown="AlwaysOnTopTriger();"></span>';
if (bClose) buttons += '<span onmousedown="Close();"></span>';
buttons += '<span>&nbsp;</span>';
document.getElementById("_window_buttons").innerHTML = buttons;
}

function placeSmoothCorner()
{
var html = "";
var width = 1*window.external.GetProperty("width", 0)-6;
var height = 1*window.external.GetProperty("height", 0)-6;
html += "<img id=_lt src='lt.gif' border=0 style='position:absolute; top:0px; left:0px;'>";
html += "<img id=_rt src='rt.gif' border=0 style='position:absolute; top:0px;
left:"+width+"px;'>";
html += "<img id=_rb src='rb.gif' border=0 style='position:absolute; top:"+height+"px;
left:"+width+"px;'>";
html += "<img id=_lb src='lb.gif' border=0 style='position:absolute; top:"+height+"px;
left:0px;'>";
document.body.innerHTML += html;
}
function moveSmoothCorner(width, height)
{
try{
document.getElementById("_lb").style.top = 1*height-6 + "px";
document.getElementById("_rb").style.top = 1*height-6 + "px";
document.getElementById("_rb").style.left = 1*width-6 + "px";
document.getElementById("_rt").style.left = 1*width-6 + "px";
}
```

```
} catch(err){}  
}
```

Widget Support Box

For supporting widgets from widgetbox.com the script has been written that adds a button to the page, which adds a widget to the custom_xml of toolbar

```
function alert(msg)  
{  
MainWindow.document.parentWindow.alert(msg);  
}  
  
function addWidget(widget)  
{  
try  
{  
if (widget==null)  
return;  
if(!MainWindow.document.parentWindow.confirm("Are you sure you want to add this widget  
on the toolbar?"))  
return;  
var context = MainWindow.document.parentWindow.document;  
  
var widgetWindow = {width: 300, height: 300, name: "Widget", sourceId: widget.sourceId}  
  
var blidgetWidthField = context.getElementById("appInstance.width");  
if (blidgetWidthField)  
widgetWindow.width = 40 + 1*blidgetWidthField.value;  
var blidgetHeightField = context.getElementById("appInstance.height");  
if (blidgetHeightField)  
widgetWindow.height = 60 + 1*blidgetHeightField.value;  
  
var blidgetNameField = context.getElementById("wdgt-name");  
if (blidgetNameField)  
widgetWindow.name = blidgetNameField.innerText;  
var sWidgetList = gTool.LoadFile("toolbarWidgetList", "");  
var widgetList = new Array();  
if (sWidgetList != "")  
widgetList = JSON.parse(sWidgetList);  
  
widgetList.push(widgetWindow);  
  
gTool.SaveFile("toolbarWidgetList", JSON.stringify(widgetList));  
  
updateCustomXML(widgetList);  
} catch(e){alert(e);}  
}  
function xmlEscape(text)  
{  
return text.replace(/&/, "&amp;").replace(/</, "<lt;").replace(/>/, "&gt;").replace(/"/, "&quot;");  
}
```

```

}

function updateCustomXML(widgetList)
{
try
{
var commands = "";
var buttons = "";

for(i=0; i<widgetList.length; i++)
{

var widget = widgetList[i];
commands +=
" <SHOWWIDGET name=\"command_\"+widget.sourceId+\" \"
template=\"template_widget.htm\" width=\"\"+widget.width+\" \" height=\"\"+widget.height+\" \"
caption=\"\"+xmlEscape(widget.name)+\" \">"+
" <PROPERTY key=\"sizeable\" value=\"true\"/>"+
" <PROPERTY key=\"minwidth\" value=\"\"+widget.width+\" \"/>"+
" <PROPERTY key=\"minheight\" value=\"\"+widget.height+\" \"/>"+
" <PROPERTY key=\"content\"><![CDATA[if (WIDGETBOX)
WIDGETBOX.renderWidget(\"+widget.sourceId+\");]]></PROPERTY>"+
" </SHOWWIDGET>";

buttons += "<BUTTON id=\"button_\"+widget.sourceId+\" \"
command=\"command_\"+widget.sourceId+\" \" caption=\"\"+xmlEscape(widget.name)+\" \"
type=\"Split\" visibility=\"1\"/>";

}

var str =
"<TOOLBAR><COMMANDS>"+commands+"</COMMANDS>"+buttons+"</TOOLBAR>";
SetXml("WidgetCustomXML", str);
} catch(e)
{
alert("updateCustomXML: " + e);
}

}

function SetXml(section, textXml)
{
try
{
if (gTool.CustomXML(section) != textXml)
{
gTool.CustomXML(section) = textXml;
gTool.Reload(); //-- reload it
}
} catch(e){alert(e)}
}

```

```

}

var gTool;

function DocumentComplete(tool, type)
{
gTool = tool;
if ((tool.url.indexOf("widgetbox.com")>0) && (type=="window"))
{
try
{
addWidgetButton();
} catch(e){alert(e)}
}
}

function addWidgetButton()
{
try
{
var context = MainWindow.document.parentWindow.document;
if (MainWindow.document.parentWindow.WbxGetWidget)
{
var box = context.getElementById("otherInstallTargets");
if (box)
{
var div = context.createElement("IMG");
//div.className = "install-tile";
div.style.width = "41px";
div.style.height = "41px";
div.style.background = "transparent url(http://softomate.net/ext/widget/tb.gif) no-repeat scroll
50% 50%;";
div.src = "http://pub.widgetbox.com/images/tile-overlay.png";

var anc = context.createElement("A");
anc.className = "install-tile";
anc.attachEvent("onmouseover", function(){MainWindow.document.parentWindow.status="";
return true;});
anc.href = "javascript: if(WbxGetWidget) ToolbarAddWidget(WbxGetWidget);";
anc.appendChild(div);
box.insertBefore(anc, box.firstChild);

MainWindow.document.parentWindow.ToolbarAddWidget = addWidget;
}
}
} catch(e){alert("addWidgetButton: "+e);}
}

```

////////////////////////////////////

// JSON Parser & Stringifier
// Here goes implementation of JSON parser