Table of Contents

Molsoft ActiveICM plug−in

Introduction to the Molsoft ActiveICM plug−in

Molsoft, LLC, 3366 Torrey Pines Court, Suite 300, La Jolla, CA 92037, USA

ActiveICM enables you to view and display 3D ICM graphical slides and animations interactively inside Microsoft PowerPoint and web browsers such as Internet Explorer, Safari and Mozilla Firefox.

The document describes technical aspects of using activeICM plug−in in web pages and communication with JavaScript.

Release Notes

Version 1.1−5

- Added an ability to simultaneously rotate several activeICM controls on the same page. See `AddToGroup` and `RemoveFromGroup` method
- Fixed bug with non−preserving center of rotation in slides after clipping planes were moved.
- Fixed bug when user click on the active link while some script is still working
- ICB preview in the thumbnail mode (Windows only)
- Fixed several issues with fonts rendering on Mac OS.
- Provides separate packages for debian/ubuntu

Version 1.1−4

- loading file progress indicator shows estimated remaining time (if loading takes more than 5 sec)
- Fixed bug with random freezing during rotation on Firefox 32 bit (Win7 64 bit OS)
- Anitaliased scalable fonts in 3D.
- Added `onLoad` and `onDisplayChange` callbacks.
- Added `GetShellVar` variable

ActiveICM in web pages

To embed an ActiveICM control into your web page `OBJECT` tag should be used.

```
<OBJECT ID="ActiveIcmCtl" WIDTH="300" HEIGHT="300" TYPE="application/x−molsoft−icb">
<!−−
plug−in parameters
−−>
</OBJECT>
```

The html code above will create an empty ActiveICM object 300x300 pixels.

Attributes:

- `ID` − name identifier of the control. Might be any string. If you have several controls in the page and plan to refer to them from `JavaScript` the identifier should be unique.
- `TYPE` − this is mandatory attribute which tells the browser to create an ActiveICM object.
- `WIDTH` and `HEIGHT` − object's window dimensions. Can be either in percents or pixels.

In most situations you want to load some data and set various properties after the ActiveICM object is created. It can be done by passing parameters to the object using *PARAM* html tag.

Possible parameters are:

Name	Type	Default	Description
projectFile	string	no default	Path to the file to load. Should be relative to the html file. Possible file formats are: ICB, PDB, MOL
rockView	integer 1/0	0	Set to "1" if you want to enable rocking after loading.

graphicsQuality	integer	5	Change graphics quality
currentSlide	integer	0	Sets initial slide (starts from 0)
bgcolor	string	black (or from slide)	Background color (named color or hex notation #070707)
slideButtonScale	float	1.	Scale slide switching buttons
onDisplayChange	string	no default	JavaScript callback on display state changes
onLoad	string	no default	JavaScript callback on data file load.

Example:

```
<OBJECT ID="ActiveIcmCtl" WIDTH="300" HEIGHT="300" TYPE="application/x-molsoft-icb">
<PARAM NAME="projectFile" VALUE="t1.icb"> <!-- Load t.icb  -->
<PARAM NAME="currentSlide" VALUE="2"> <!-- set initial slide -->
<PARAM NAME="onLoad" VALUE="alert('Done');">  <!-- alert when done -->
</OBJECT>
```

JavaScript interface

JavaScript language can be used to communicate with ActiveICM object after the page is loaded.

To get a reference to the ActiveICM object *getElementById* method of the **document** class can be used.

Example:

```
var acticm = document.getElementById('ActiveIcmCtl');
```

ActiveICM object methods

NextSlide()

Switches to the next slide. Updates the view. If the current slide is the last one or there are no slides nothing will happen.

Arguments : none

Return value: none.

PrevSlide()

Switches to the previous slide. Updates the view. If the current slide is the first one or there are no slides nothing will happen.

Arguments : none

Return value: none.

Example:

```
<HTML>
<p><OBJECT ID="ActiveIcmCtl" WIDTH="300" HEIGHT="300" TYPE="application/x-molsoft-icb">
<PARAM NAME="projectFile" VALUE="t1.icb">
</OBJECT></p>
<a href="javascript:document.getElementById('ActiveIcmCtl').NextSlide();">Next Slide</a>
<a href="javascript:document.getElementById('ActiveIcmCtl').PrevSlide();">Prev Slide</a>
</HTML>
```

RunCommands(s_icmCommands)

Runs a list of semicolon or newline separated list of ICM shell commands. The method is the main gate to communicate with ICM shell. It allows you to run arbitrary set of ICM commands.

Arguments : string containing ICM commands.

Return value: none.

Example:

```
<HTML>
<SCRIPT>
function f()
{
  var ctl = document.getElementById('ActiveIcmCtl');
  ctl.RunCommands( 'color a_/hi* red;' );
}
</SCRIPT>
<p><OBJECT ID="ActiveIcmCtl" WIDTH="300" HEIGHT="300" TYPE="application/x-molsoft-icb">
<PARAM NAME="projectFile" VALUE="t1.icb">
</OBJECT></p>
<a href="javascript:f();">Color</a>
</HTML>
```

IsDisplayed(s_icmObjectName|s_meshName|s_mapName)

Returns the display state of the object, atom selection, mesh or map. Molecular object or atom selection is considered to be displayed if at least one atom is displayed.

Arguments : string containing name of the object.

Return value: logical.

GetShellVar(s_icmName)

Returns ICM shell variable in JSON format.

Arguments : string containing name of the shell variable.

Return value: string.

AddToGroup(s_groupName)

Adds control to specified group for simultaneous rotation/zooming/panning.

Arguments : string containing name of the group.

Return value: none.

Example 1: (two control side−by−side rotating simultaneously)

```
<html>
<script>
function addToGroup(id)
{
  document.getElementById(id).AddToGroup('gr1');
}
</script>
<table>
<tr>

<td>
<object ID="act1" WIDTH="400" HEIGHT="400" type="application/x-molsoft-icb">
<param name="projectFile" value="Scripting_Workshop_ActiveICM.icb" />
<param name="currentSlide" value="1" />
<param name="onLoad" value="addToGroup('act1');" />
</object>
</td>

<td>
<object ID="act2" WIDTH="400" HEIGHT="400" type="application/x-molsoft-icb">
<param name="projectFile" value="Scripting_Workshop_ActiveICM.icb" />
<param name="currentSlide" value="2" />
<param name="onLoad" value="addToGroup('act2');" />
```

```
</object>
</td>

</tr>
</table>
</html>
```

Example 2: (interactively toggles simultaneous rotation)

```
<html>
<script>
function toogleGroup(id,actid)
{
  if (document.getElementById(id).checked)
    document.getElementById(actid).AddToGroup('mygroup');
  else
    document.getElementById(actid).RemoveFromGroup();
}
</script>
<table>
<tr>
<td><OBJECT ID="act1" WIDTH=300 HEIGHT=300 type="application/x-molsoft-icb">
<PARAM name="projectFile"  value="pep.icb">
<PARAM name="currentSlide" value="0">
</OBJECT></td>
<td><OBJECT ID="act2" WIDTH=300 HEIGHT=300 type="application/x-molsoft-icb">
<PARAM name="projectFile" value="pep.icb">
<PARAM name="currentSlide" value="1"></td>
</OBJECT></td>
<td><OBJECT ID="act3" WIDTH=300 HEIGHT=300 type="application/x-molsoft-icb">
<PARAM name="projectFile" value="pep.icb">
<PARAM name="currentSlide" value="2"></td>
</OBJECT></td>
</tr>
<TD align="center"><input id="c1" type=checkbox onclick="toogleGroup('c1','act1');"></TD>
<TD align="center"><input id="c2" type=checkbox onclick="toogleGroup('c2','act2');"></TD>
<TD align="center"><input id="c3" type=checkbox onclick="toogleGroup('c3','act3');"></TD>
<tr>
</tr>
</table>
</OBJECT>
</html>
```

RemoveFromGroup([s_groupName])

Adds control from the specified group or from all groups if nothing is provided.

Arguments : string containing name of the group.

Return value: none.

ActiveICM object properties

rockView

Logical property. Toggles rocking mode. (Does not supported on Linux)

Example:

```
<HTML>
<p><OBJECT ID="ActiveIcmCtl" WIDTH="300" HEIGHT="300" TYPE="application/x-molsoft-icb">
<PARAM NAME="projectFile" VALUE="t1.icb">
</OBJECT></p>
<input type="checkbox" id="rock" onclick="document.getElementById('ActiveIcmCtl').rockView=doc
```

currentSlide

Integer property. Holds current slide number. Can be used in PARAM tag to set initial slide after ActiveICM object creation.

Example:

```
<HTML>
<p><OBJECT ID="ActiveIcmCtl" WIDTH="300" HEIGHT="300" TYPE="application/x-molsoft-icb">
<PARAM NAME="projectFile" VALUE="t1.icb">
</OBJECT></p>
<input type="checkbox" id="hq" onclick="document.getElementById('ActiveIcmCtl').graphicsQualit
```

nofSlides

read−only **integer** property. Returns number of slides in the project.

pluginVersion

read only **integer** property. Returns plug−in version number.

projectFile

String property. Holds currently loaded project file name. The property can be changed dynamically to load new project file. New data file will be loaded form the location relative to the html document which contains ActiveICM control.

The following file formats are supported:

- ICB (Molsoft binary format)
- PDB (Protein Data Bank format)
- MOL (MDL mol format)

Example:

```
<html>
<object ID="act1" WIDTH="300" HEIGHT="300" type="application/x-molsoft-icb"> </object>
<hr>
<input type="button" value="Load Project 1" onClick="document.getElementById('act1').projectFi
<input type="button" value="Load Project 2" onClick="document.getElementById('act1').projectFi
</html>
```

bgcolor

String property. Hold background color of 3D window. Default: empty (use default from ICB file or black)

Example:

```
<object id="3dmol" type="application/x-molsoft-icb" width="180" height="180" visible="true" >
<param name="projectFile" value="mol3d.mol" />
<param name="bgcolor" value="lightgreen" />
</object>
```

slideButtonScale

Real property. Holds scale coefficient for slides button. Default: 1.

onLoad

String property. Holds javascript code which is executed when document is loaded. Default: empty

onDisplayChange

String property. Holds javascript code which is executed when display state was changed. (slide transition for example) Default: empty

Example:

```
<script>function js_updatecheckboxes() {
parent.frames['a'].document.getElementById('cb1_1').checked = document.getElementById('ActiveI
parent.frames['a'].document.getElementById('cb1_2').checked = document.getElementById('ActiveI
parent.frames['a'].document.getElementById('cb1_3').checked = document.getElementById('ActiveI
}
</script>
<OBJECT ID="ActiveIcmCtl" WIDTH="100%" HEIGHT="100%" type="application/x-molsoft-icb" >
<param name="projectFile" value="checkbox_export_bin.icb">
<param name="onDisplayChange" value="js_updatecheckboxes();">
</OBJECT>
```

Index