



Publish Cult3D in HTML - guide extended with javascript

Using Cult3D Objects in a Web Page

The technique used to embed a Cult3D object in an HTML page in a way that is compatible both with Netscape Navigator, (version 2.0 or later) and Internet Explorer (version 3.0 or later), is a bit complex. Luckily, there is a "recipe" which you can follow, and which is described in the following sections. If you are not interested in the details, just read the sections "Preparing Your Web Server," "The Basic Recipe," and "Cult3D Viewer Plug-in Parameters," and have a look at the Examples in the last section. If you want to know what the various fixed attributes do, you may also want to read the section "Specific OBJECT and EMBED Element Attributes." If you want to understand why you have to do things in such a specific way, you should read the section below entitled "Some Technical Details."

Preparing Your Web Server

In order for the end-user's browser to recognize a Cult3D object, you have two options. The best option is to install the Cult3D MIME type "application/x-cult3d-object" with the file suffix/extension ".co" on the web server on which your object resides. (Your Webmaster should know how to do this.) If for some reason this is not possible, you should instead use the TYPE attribute in the EMBED element (see EMBED element attributes in the section called "Specific OBJECT and EMBED Element Attributes").

The Basic Recipe

To use a Cult3D object in your HTML page, you need to specify some information for the browser. The minimum information requirements include where the object resides, and the width and height of the sub-window showing the object. You may also want to send some optional parameters to the Cult3D Viewer plug-in to further specify the appearance and behavior of your object.

To embed a Cult3D object in your HTML page you can use the following "skeleton":

```
<object
classid="clsid:31B7EB4E-8B4B-11D1-A789-00A0CC6651A8"
codebase="http://www.cult3d.com/download/cult.cab"
width=width height=height>
<param name="SRC" value=source-url>
more object parameters
<embed
type="application/x-cult3d-object "
pluginspage="http://www.cult3d.com/newuser/index.html "
src=source-url
width=width height=height
more embed parameters
>
</embed>
</object>
```

Copy this skeleton into your HTML page and supply the appropriate width, height, source-URL, and if applicable, more parameters. As you see, you need



to specify all parameters twice, once for the OBJECT element and once for the EMBED element. Follow these steps:

Supply the source-URL. This is the location of your Cult3D object. (You need to insert this in two places.)

Supply the width and height. This is the size of your Cult3D object's sub-window in your page, in pixels. (You also need to insert these in two places.)

Determine if you need to specify any further parameters. These are described in the section Cult3D Script Engine. For each parameter param with value value, insert one line in the more object parameters part on the form:

```
<param name="param" value=value>
```

and one line in the more embed parameters part on the form

```
param=value
```

So, for example, let's say you want to insert the object "myobject.co" in your HTML page. You want it to appear in a sub-window that is 240 pixels high and 320 pixels wide. Let's assume your Web server is configured correctly, so you do not need the TYPE attribute. Following the steps above should give you the following:

```
<object
classid="clsid:31B7EB4E-8B4B-11D1-A789-00A0CC6651A8"
codebase="http://www.cult3d.com/download/cult.cab"
width=320 height=240>
<param name="SRC" value="myobject.co">
<embed
plug-inspage="http://www.cult3d.com/newuser/index.html"
src="myobject.co"
width=320 height=240>
</embed>
</object>
```

Specific OBJECT and EMBED Element Attributes

This section describes the various attributes specific to the OBJECT and EMBEDS elements. You do not need to read this to get a Cult3D object onto your Web site, but we still recommend that you at least gloss through this information, to get a better understanding of how it works.

<OBJECT> Element Attributes

```
CLASSID="clsid:31B7EB4E-8B4B-11D1-A789-00A0CC6651A8"
```

The CLASSID and the related clsid: modifier is a confirmation that we are using ActiveX. The number following clsid is the class identifier for the Cult3D Viewer plug-in, and it is important that you do not make any typing errors when putting this into your HTML page.

```
CODEBASE="http://www.cult3d.com/download/cult.cab"
```

If the ActiveX control is not already installed, this parameters tells the browser where it can be found for download. When it is downloaded, your system checks its digital signature and automatically installs it.

<EMBED> Element Attributes

```
PLUG-INSPAGE="http://www.cult3d.com/newuser/index.html"
```



This attribute indicates where the user should go to get the Cult3D Viewer plug-in. You should include it with every Cult3D object, exactly as written above. This will make life much easier for end-users who do not already have the Cult3D Viewer plug-in, as it will prompt their browsers to help them get the latest version, with a minimum amount of hassle, straight from the original source. It will greatly enhance the user-friendliness of your site. For the sake of your end-users who use Netscape you should consider this attribute mandatory!

TYPE="application/x-cult3d-object"

If the Web server on which your site resides does not have a registered MIME type for Cult3D objects, you should set it yourself with this attribute.

Some Technical Details

The reason for the need of two different HTML tags, is because Internet Explorer and Netscape use different mechanisms to extend their run-time behavior with plug-ins.

Netscape started supporting plug-ins long before there was a standard for it. They did so by introducing the EMBED element and defining their own plug-in format. The EMBED element has never been part of the official HTML standard, but it is supported, more or less, by many browsers on the market.

When this kind of functionality was later standardized in HTML 3.2, it was done through the introduction of the OBJECT element. Internet Explorer uses this to identify an ActiveX control to be used with the plug-in data.

So, what we have is two different mechanisms by which the behavior of a browser can be extended: the EMBED element for Netscape plug-ins, and the OBJECT element for (among other things) ActiveX controls. Internet Explorer does support the former, to some extent (but not well enough to be compatible with Cult3D). Netscape pre-4.0 does not support the latter at all, and though Netscape 4.0 does, it does not know how to deal with ActiveX controls.

Fortunately, the OBJECT element is designed to deal with this kind of problems. "Old-time" browsers (such as Netscape pre-4.0) that are not aware of the OBJECT element will just ignore it, and all enclosed PARAM elements, as it does not know what to do with them; they will, however, try to interpret anything else enclosed in the OBJECT, such as an EMBED element. Newer browsers, that are aware of the OBJECT element, but that do not support its use with ActiveX controls (such as Netscape 4.0) will do the same. And browsers that support the use of the OBJECT element for specifying ActiveX controls (such as Internet Explorer 3.0 or later) will interpret it, together with any enclosed PARAM elements, and ignore anything else enclosed in the OBJECT, such as an EMBED element. So, if we use the "skeleton" described above, we have a piece of HTML code that will cause Netscape to see an EMBED element, and Internet Explorer to see an OBJECT element specifying an ActiveX control—which is what we wanted.

Example

Below is an example of how an embedded Cult3D object might actually look in an HTML page.

```
<object  
classid="clsid:31B7EB4E-8B4B-11D1-A789-00A0CC6651A8"  
codebase="http://www.cult3d.com/download/cult.cab"
```



```
width=300 height=250>
<param name="src" value="cycore.co">
<param name="PBCOLOR"="000000">
<embed
plug-inspage="http://www.Cult3D.com/newuser/index.html"
width=300 height=250
PBCOLOR="000000"
src="cycore.co">
</embed>
</object>
```

Cult3D feature set

This document describes the scripting features for Cult3D and how to use them.

Cult3D's interface has been extended to add interaction between a Cult3D object and a web page. They can for instance, start an animation when the user selects the correct answer or change an object so it has a custom color. Some of these new features must be handled differently depending on if they are running under Netscape Communicator or Microsoft Internet Explorer.

Cult3D events

An event is triggered when a condition is met. There are four different events: the scene is loaded, the complete object is loaded, the object is invalid or missing, or custom messages created by the designer. When one of these four events is triggered, it can activate scripts in the web page. If the web page contains no handling for a specified event, it will silently ignore the event and continue.

Example code for how to connect an event using JavaScript for Internet Explorer:

```
<SCRIPT LANGUAGE="javascript">
<!--
function OnLoadFinished()
{
// Insert your code here
}
//-->
</SCRIPT>
<SCRIPT LANGUAGE=javascript FOR=Cult3D EVENT=OnLoadFinished>
<!--
// Call the actual method
OnLoadFinished()
//-->
</SCRIPT>
<OBJECT id=Cult3D
classid="clsid:31B7EB4E-8B4B-11D1-A789-00A0CC6651A8"
width=192 height=192>
<PARAM NAME="SRC" VALUE="Test.co">
</OBJECT>
```

Example code for how to connect an event using VBScript for Internet Explorer:

```
<SCRIPT LANGUAGE="vbscript">
<!--
Sub Cult3D_OnLoadFinished()
```



CYCORE

www.cycore.com

phone +46 (0)18 65 65 60, fax +46 (0)18 65 65 66, info@cycore.com

```
` Insert your code here

End Sub
-->
</SCRIPT>

<OBJECT id=Cult3D
classid="clsid:31B7EB4E-8B4B-11D1-A789-00A0CC6651A8"
width=192 height=192>
<PARAM NAME="SRC" VALUE="Test.co">
</OBJECT>
```

Even though Microsoft Internet Explorer supports VBScript, we recommend you to use JavaScript to get a coherent interface.

Example code for how to connect an event with JavaScript for Netscape Communicator:

```
<SCRIPT LANGUAGE="javascript">
<!--
function OnLoadFinished()
{

    // Insert your code here

}
//-->
</SCRIPT>
<EMBED NAME=Cult3D
PLUGINSPAGE="http://www.cult3d.com/newuser/index.html"
SRC="Test.co"
WIDTH="192" HEIGHT="192"
ONLOADFINISHED="OnLoadFinished()">
</EMBED>
```

Event: OnSceneLoaded

Cult3D will trigger this event when enough information has been loaded to start the scene. Textures and cursors might not have been loaded yet.

Netscape example:

```
<EMBED NAME="<name to be used to access the object>"
PLUGINSPAGE="http://www.cult3d.com/newuser/index.html"
SRC="yourobject.co"
WIDTH="400" HEIGHT="300"
ONSCENELOADED="alert('OnSceneLoaded was triggered!')">
</EMBED>
```

Event: OnLoadFinished

Cult3D will trigger this event when the whole file has been loaded and decompressed.

Netscape example:

```
<EMBED NAME=Cult3D
PLUGINSPAGE="http://www.cult3d.com/newuser/index.html"
```



```
SRC="yourobject.co"  
WIDTH="400" HEIGHT="300"  
ONSCENELOADED='document.embeds["Cult3D"].triggerEvent("EventName")'>  
</EMBED>
```

Event: OnLoadAborted

Cult3D will trigger this event when the file is invalid or the file doesn't exist.

Event: OnSendMessage

Cult3D will trigger this event whenever it receives a "Send Message to Host" call activated by the designer or by a java function. Retrieve the message by calling the method "getLastMessage()".

Cult3D parameters

These parameters can only be set by HTML tags, e.g.:

```
<EMBED NAME=Cult3D  
PLUGINSPPAGE="http://www.cult3d.com/newuser/index.html"  
SRC="yourobject.co"  
WIDTH="400" HEIGHT="300"  
PBCOLOR="00FF00">  
</EMBED>
```

This Cult3D object has a width and size of 400x300 pixels and the progress bar color (PBCOLOR) is set to green. The progress color can only be set by using a HTML tag.

DISABLEHW

This parameter controls hardware acceleration. Default value is "0" which enables hardware acceleration if available. Set the value to "1" to disables all hardware acceleration.

FRAMESKIP

This parameter controls automatic frame skipping to compensate for slower computers. Default value is "1" which enables frame skipping. Set the value to "0" to disable frame skipping.

VIEWFINISHED

This parameter controls the rendering of the Cult3D object until the object has been completely loaded (see FINISHEVENT). Default value is "0" which allows Cult3D to render the object while it is loading the object. Set to any non-zero value to force Cult3D to not render anything until the object is completely loaded.

ANTIALIASING

This parameter controls only the software antialiasing. It can be set to three (3) different values:

0 - automatic antialiasing, if antialiasing is turned on when nothing has happened in the scene after the time period set by "ANTIALIASINGDELAY". When anything moves, antialiasing is turned off to improve speed. This is the default value.

1 - disable antialiasing

2 - enable antialiasing

ANTIALIASINGDELAY



This parameter controls the delay before the automatic antialiasing for the software render is turned on. The value must be positive and default value is 1000 ms, which is equal to one second.

PBCOLOR

This parameter controls the progress bar color. If no value is given or the value is invalid, it will automatically be set to white (FFFFFF). The color must be a hex value.

FF0000 - red progress bar
00FF00 - green progress bar
0000FF - blue progress bar

DISABLEPB

This parameter controls the visibility of the progress bar. Default value is "0" which enables the progress bar. Set the parameter to "1" to disable the progress bar.

JavaScript methods

The object has a couple of new methods to allow the web page control the object:

Netscape:

```
object.setAntiAliasing(mode)  
object.getAntiAliasing()
```

Internet Explorer:

```
object.AntiAliasing
```

This property allows the web page to set and get the antialiasing mode. Antialiasing can only be changed when rendering using software. It can be set to three (3) different values:

0 - automatic antialiasing, if antialiasing is turned on when nothing has happened in the scene after the time period set by "ANTIALIASINGDELAY". When anything moves, antialiasing is turned off to improve speed. This is the default value.

1 - disable antialiasing

2 - enable antialiasing

See the "ANTIALIASING" tag description for more information.

Netscape:

```
object.setAntiAliasingDelay(delay)  
object.getAntiAliasingDelay()
```

Internet Explorer:

```
object.AntiAliasingDelay
```

This property allows the web page to set and get the antialiasing delay for automatic antialiasing. Antialiasing delay can only be changed when rendering using software. See the "ANTIALIASINGDELAY" tag description for information.

Netscape:

```
object.triggerEvent(name)
```

Internet Explorer:

```
object.TriggerEvent(name)
```



This method triggers a named event and returns a boolean value, which is true if Cult3D found the trigger.

Netscape example:

```
var Cult3D = document.embeds["Cult"];
var triggered = Cult3D.triggerEvent("MoveClock");

if (!triggered)

    alert("Event not found!");
```

Netscape:

object.getVersion()

Internet Explorer:

object.Version

This method returns the version of the installed Cult3D Player.

Netscape example:

```
var Cult3D = document.embeds["Cult"];

var CultVersion = Cult3D.getVersion();

window.alert("Cult3D version " + CultVersion);
```

Netscape:

object.getLastMessage()

Internet Explorer:

object.LastMessage

This method returns the last message sent by "Send Message to Host". It is set to null if no messages has been sent.

Netscape:

object.getFrame()

Internet Explorer:

object.Frame

This method returns the frame counter for the Cult3D object. It increases by one every time Cult3D has rendered a new frame.

Netscape:

object.setFocus()

Internet Explorer:

object.Focus

This method sets the keyboard focus to the Cult3D object. This can be used to manually change the focus from a web control back to the Cult3D object so the user doesn't have to click on the Cult3D object to set the keyboard focus.

Netscape:

object.getActionStatus(name)



Internet Explorer:

```
object.GetActionStatus(name)
```

This method returns the action status of a named event. It can be used to retrieve if certain actions have occurred and trigger different events accordingly. The function can return one of the following values:

- 1 - Event name not found
- 0 - Event action status is clear
- 1 - Event action status is set

Netscape example:

```
var Cult3D = document.embeds["Cult"];
var status = Cult3D.getActionStatus("OpenDoor");

if (status == -1)
    alert("Event not found!");
else if (status == 1)
    Cult3D.triggerEvent("CloseDoor");
else
    Cult3D.triggerEvent("OpenDoor");
```

Netscape:

```
object.triggerAction(name,action)
```

Internet Explorer:

```
object.TriggerAction(name,action)
```

This method is used to send a message from an application to a Cult3D java event. It is slightly different from the "triggerEvent" method, which doesn't allow the application to send any parameters. The action parameter must be a string and the function will return true if Cult3D found the event. It is up to the java method to parse the string and do any appropriate action with it.

Netscape example:

```
var Cult3D = document.embeds["Cult"];

var triggered = Cult3D.triggerAction("SetClock", "" + new
Date().getTime());

if (!triggered)
    alert("Event not found!");
```

Netscape example with JavaScript

```
<HTML>
<HEAD>
<TITLE>Netscape example</TITLE>
<SCRIPT LANGUAGE="JavaScript"><!--
function OnLoad() {
    document.frm.txt.value = "Object loaded!";
```



```
        document.embeds["Cult3D"].triggerEvent("Start");
    }

function OnAbort() {

document.frm.txt.value = "Missing or invalid file!";
}

function OnScene() {
    document.frm.txt.value = "Loading, please wait...";
}

!--></SCRIPT>
<BODY>
<EMBED NAME=Cult3D
PLUGINSPPAGE="http://www.cult3d.com/newuser/index.html"
SRC="Test.co"
WIDTH="400" HEIGHT="300"
ONSCENELOADED="OnScene()"
ONLOADABORTED="OnAbort()"
ONLOADFINISHED="OnLoad()">
</EMBED>
<FORM ID=frm NAME=frm>
Result: <INPUT NAME=txt TYPE="TEXT" SIZE=30>
</FORM>
</BODY>
</HTML>
```

Resource for java and vb script.

Below is source code for java script functions that can be used with Cult3D. It is meant to simplify the scripting for Netscape and Internet Explorer. The code can also be found under the name "Cult3D.js" in [Cult3D Designer installation dir]/HTML/

```
<!--
// Cult3D support file for Netscape Navigator and Internet Explorer

// AntiAliasing
var AA_AUTOMATIC = 0; // Automatic antialiasing when using software
var AA_DISABLED = 1; // Disable antialiasing when using software
var AA_ENABLED = 2; // Enable antialiasing when using software

// Action status
var AS_NOTFOUND = -1; // Action status not found
var AS_CLEAR = 0; // Action status is clear
var AS_SET = 1; // Action status is set

// Netscape support functions

function NS_getVersion()
{
    return this.Cult3D.getVersion();
}

function NS_triggerEvent(name)
{
```



CYCORE

www.cycore.com

phone +46 (0)18 65 65 60, fax +46 (0)18 65 65 66, info@cycore.com

```
    return this.Cult3D.triggerEvent(name);
}

function NS_getLastMessage()
{
    return this.Cult3D.getLastMessage();
}

function NS_setAntiAliasing(aa_mode)
{
    this.Cult3D.setAntiAliasing(aa_mode);
}

function NS_getAntiAliasing()
{
    return this.Cult3D.getAntiAliasing();
}

function NS_setAntiAliasingDelay(aa_delay)
{
    this.Cult3D.setAntiAliasingDelay(aa_delay);
}

function NS_getAntiAliasingDelay()
{
    return this.Cult3D.getAntiAliasingDelay();
}

function NS_getFrame()
{
    return this.Cult3D.getFrame();
}

function NS_setFocus()
{
    this.Cult3D.setFocus();
}

function NS_getActionStatus(name)
{
    return this.Cult3D.getActionStatus(name);
}

function NS_triggerAction(name,action)
{
    return this.Cult3D.triggerAction(name,action);
}

// This are available in version 5.2.x.x and later
function NS_setStereroMode(mode)
{
    this.Cult3D.setStereroMode(mode);
}

function NS_getStereoMode()
{
    return this.Cult3D.getStereoMode();
}
```



CYCORE

www.cycore.com

phone +46 (0)18 65 65 60, fax +46 (0)18 65 65 66, info@cycore.com

```
function NS_setDisparity(disparity)
{
  this.Cult3D.setDisparity(disparity);
}

function NS_getDisparity()
{
  return this.Cult3D.getDisparity();
}

// Internet Explorer support functions

function IE_getVersion()
{
  return this.Cult3D.Version;
}

function IE_triggerEvent(name)
{
  return this.Cult3D.TriggerEvent(name);
}

function IE_getLastMessage()
{
  return this.Cult3D.LastMessage;
}

function IE_setAntiAliasing(aa_mode)
{
  this.Cult3D.AntiAliasing=aa_mode;
}

function IE_getAntiAliasing()
{
  return this.Cult3D.AntiAliasing;
}

function IE_setAntiAliasingDelay(aa_delay)
{
  this.Cult3D.AntiAliasingDelay=aa_delay;
}

function IE_getAntiAliasingDelay()
{
  return this.Cult3D.AntiAliasingDelay;
}

function IE_getFrame()
{
  return this.Cult3D.Frame;
}

function IE_setFocus()
{
  this.Cult3D.Focus();
}
```



```
function IE_getActionStatus(name)
{
    return this.Cult3D.GetActionStatus(name);
}

function IE_triggerAction(name,action)
{
    return this.Cult3D.TriggerAction(name,action);
}

// This are available in version 5.2.x.x and later
function IE_setStereoMode(mode)
{
    this.Cult3D.StereroMode=mode;
}

function IE_getStereoMode()
{
    return this.Cult3D.StereoMode;
}

function IE_setDisparity(disparity)
{
    this.Cult3D.Disparity=disparity;
}

function IE_getDisparity()
{
    return this.Cult3D.Disparity;
}

// Dummy function

function None_Dummy()
{
}

// Identify the browser, wait for a small period of time to make sure
// that Netscape is able to start the Java engine. Provide the
// same interface for both Internet Explorer and Netscape
// Communicator.

function Cult3D_Object(name)
{
    if (typeof(this.Type)!="undefined")
        return;
    if (typeof(document.embeds[name])=="object") {
        this.Type="NS";
        this.Cult3D=document.embeds[name];
        this.triggerEvent=NS_triggerEvent;
        this.getVersion=NS_getVersion;
        this.getLastMessage=NS_getLastMessage;
        this.getAntiAliasing=NS_getAntiAliasing;
        this.getAntiAliasingDelay=NS_getAntiAliasingDelay;
        this.setAntiAliasing=NS_setAntiAliasing;
        this.setAntiAliasingDelay=NS_setAntiAliasingDelay;
    }
}
```



```
    this.getFrame=NS_getFrame;
    this.setFocus=NS_setFocus;
    this.getActionStatus=NS_getActionStatus;
    this.triggerAction=NS_triggerAction;
    this.getStereoMode=NS_getStereoMode;
    this.setStereoMode=NS_setStereoMode;
    this.getDisparity=NS_getDisparity;
    this.setDisparity=NS_setDisparity;
    return;
}
if (typeof(eval(name))=="object" &&
typeof(eval(name).Version)=="string") {
    this.Type="IE";
    this.Cult3D=eval(name);
    this.triggerEvent=IE_triggerEvent;
    this.getVersion=IE_getVersion;
    this.getLastMessage=IE_getLastMessage;
    this.getAntiAliasing=IE_getAntiAliasing;
    this.getAntiAliasingDelay=IE_getAntiAliasingDelay;
    this.setAntiAliasing=IE_setAntiAliasing;
    this.setAntiAliasingDelay=IE_setAntiAliasingDelay;
    this.getFrame=IE_getFrame;
    this.setFocus=IE_setFocus;
    this.getActionStatus=IE_getActionStatus;
    this.triggerAction=IE_triggerAction;
    this.getStereoMode=IE_getStereoMode;
    this.setStereoMode=IE_setStereoMode;
    this.getDisparity=IE_getDisparity;
    this.setDisparity=IE_setDisparity;
    return;
}
this.Type="";
this.Cult3D=null;
this.getVersion=None_Dummy;
this.getLastMessage=None_Dummy;
this.getAntiAliasing=None_Dummy;
this.getAntiAliasingDelay=None_Dummy;
this.setAntiAliasing=None_Dummy;
this.setAntiAliasingDelay=None_Dummy;
this.getFrame=None_Dummy;
this.setFocus=None_Dummy;
this.getActionStatus=None_Dummy;
this.triggerAction=None_Dummy;
this.getStereoMode=None_Dummy;
this.setStereoMode=None_Dummy;
this.getDisparity=None_Dummy;
this.setDisparity=None_Dummy;
}

function isIE()
{
    return navigator.appName.indexOf('Microsoft') != -1;
}

//-->
```