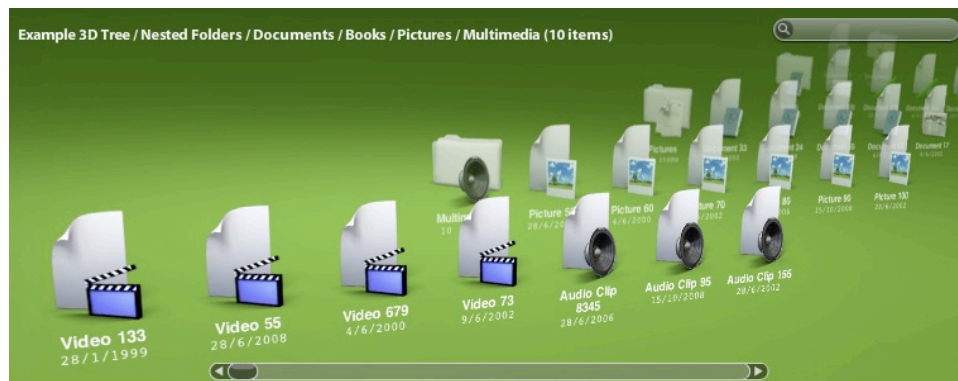


Zerofractal Tree3D v2.0.0 - User Guide

Thank you for choosing the **Zerofractal Tree3D 2**. **Tree3D 2** is a Flash component for ActionScript 3 that allows you to easily create a three dimensional interactive tree UI navigation system. Based on data driven XML, the component can be easily adapted to virtually any data source.



Zerofractal Bogotá

Calle 103 # 21- 60
Piso 1
Bogotá, Colombia
T (031) 2566763

bogota@zerofractal.com
<http://www.zerofractal.com>

Features

- Data driven XML bases
- Customizable Label and Icon styles
- Animateable Icons
- Rows & Columns
- Vertical tree suport
- Breadcrumb navigation
- Customizable number of items per page, item size, item spacing
- Customizable camera rotation
- Customizable scrollbar, adaptable to Adobe's UI Scrollbar

What's New

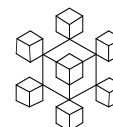
Rows & Columns

In Tree3D 2 we redesigned the way data is displayed. Before, items could only appear on a single horizontal row. Now you can define any number of visible rows and columns. Some great new possibilities with rows and columns are:

- **Vertical Tree** By combining defining 1 Column, and tweaking the levelXSpacing and levelYSpacing parameters you can easily create vertical Trees.
- **3D Wall** By combining Rows and Columns, you can easily create interactive 3D walls.

Alignment

Tree3D can now be aligned to the left, right or centered. This affects views where there are more than 1 row.



Zerofractal Bogotá

Calle 103 # 21- 60

Piso 1

Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>

Improved memory and CPU handling

Tree3D now uses the strictly necessary amount of CPU, reducing considerably the overhead of the component. Memory garbage collection is also handled by the component so that the memory footprint stays as low as possible at all times (*this feature requires flash player 9.0.115 or higher.

Independent Vertical or Horizontal Camera Rotation

Tree3D now has independent camera rotation.

On the fly setting changes

Some settings that previously required a tree refresh may now be initiated on the fly. They will affect the current level and parent/child levels when the user navigates the tree.

Disabling Captions

Caption texts may now be enabled or disabled.

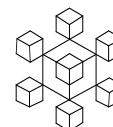
Programatically open children

Developers may not ask the component to open a child folder via actionscript using the childItem method.

Papervision 3D Settings

We added 3 settings to control the performance in tin tree3D's instance of Papervision 3D

- **Smoothing** This setting may disable the bitmap smoothing for the tree items. Smoothing increases the picture quality but decreases the performance of the component. The default value is true.
- **Precise** This affects how Papervision subdivides each item to render it correctly. Precision increases the accuracy of the item's texture but has an impact on performance. The default value is false.
- **Precision** This settings defines the level of precision when precise is set to true. Valid values must be integer numbers 1 or higher. The default value is 8. This setting has no effect if precision is set to false.



Release Notes

This new version of tree3D contains structural changes. For this reason, some properties have been removed and replaced. Other default values have changed. If you are migrating an application developed with tree3D 1.x, please revise the class reference to adjust your code accordingly.

The following properties have depreciated:

items per page:int
itemspacing:Number
menuY:Number

The following properties have been added:

columns:int
rows:int
cameraRotationX:Boolean
cameraRotationY:Boolean
columnSpacing:Number
rowSpacing:Number
treeX:Number
treeY:Number
levelXSpacing:Number
levelYSpacing:Number
levelZSpacing:Number
align:String ["left,center,right"]
captions:Boolean
smooth:Boolean
precise:Boolean
precision:int

Zerofractal Bogotá

Calle 103 # 21- 60
Piso 1
Bogotá, Colombia
T (031) 2566763

bogota@zerofractal.com
<http://www.zerofractal.com>

Included in the Product

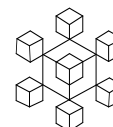
- the tree3D.swc Component in a MXP installer
- Example FLA files walking through every mayor feature
- Real world application example FLA files
- user guide in PDF
- class library in PDF

Updates

The product includes free v2.x updates

Support

For updates and support, please visit <http://tree3d.zerofractal.com> or write to tree3d@zerofractal.com



The Basics

Installation

1. Unpack the distribution zip file.
2. Double click on the tree3D.mxp file
3. read and accept the license agreement (EULA)

Once this steps are completed, the component will be successfully installed

Step By Step Examples

In the examples folder you will find a set of step by step examples that will walk you though the tree3D implementation with the following easy to understand procedures:

4. Loading XML Data
5. Customizing Icons
6. Customizing Labels
7. Customizing Labels
8. Understanding and using Breadcrumbs
9. Handling Events, including item click

Zerofractal Bogotá

Calle 103 # 21- 60

Piso 1

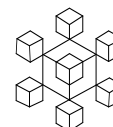
Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>

IMPORTANT: tree3D uses 3D rendering and therefore must be redrawn in every frame. To get a smooth motion from the component you must configure your fla file to at least 24 fps. Higher values yield a smoother menu, for example 60 fps.



Usage - XML DATA

Data Structure

The tree accepts any XML with a hierarchical structure. By default, the component uses the name attribute to assign the label name and the number of child nodes to decide whether an element corresponds to a folder or to an item.

Example:

```
<xml name="My tree 3D">
  <folder name="folder 1">
    <document name="document1">
    <document name="document2">
    <document name="document3">
    <folder name="sub-folder 1">
      <document name="document1">
      <document name="document2">
    </folder>
  </folder>
  <folder name="folder 2" />
  <folder name="folder 3" />
</xml>
```

Zerofractal Bogotá

Calle 103 # 21- 60

Piso 1

Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>

Data describing each item in the tree must be placed as attributes rather than as nodes. Sub-nodes will be treated as children in the tree structure.

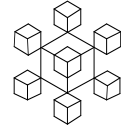
This will not work:

```
<node>
<name>Item 1</name>
<icon>document.png</icon>
<url>http://www.zerofractal.com</url>
</node>
```

Instead you must feed Tree3D this:

```
<node name="Item 1" icon="document.png" url="http://www.zerofractal.com" />
```

When reading xml files formatted with values instead of attributes, you can always easily parse them in ActionScript before feeding it to Tree3D using the AS3 XML Object.



Loading Data

Tree3D supports AS 3's native XML object. This means that data can come from external XML files or XMLs fabricated within flash.

XML Object Example:

```
var xmlData:XML = new XML(<tree name="Breadcrumbs demo">
    <folder name="Folder 1">
        <item name="Document 1"/>
        <item name="Document 2"/>
        <folder name="Sub Folder 1">
            <item name="Document 1"/>
            <item name="Document 2"/>
            <item name="Document 3"/>
            <item name="Document 4"/>
        </folder>
    </folder>
    <item name="Document 1"/>
    <item name="Document 2"/>
    <item name="Document 3"/>
    <item name="Document 4"/>
</tree>);
```

```
tree3D.loadXMLData(xmlData);
```

External XML File:

```
var xmlLoader:URLLoader = new URLLoader();
xmlLoader.addEventListener(Event.COMPLETE, onLoadXML);
xmlLoader.load(new URLRequest("example.xml"));
```

```
function onLoadXML(e:Event):void {
    var xmlData:XML = new XML(e.target.data);

    //Loads the data onto the tree
    tree3D.loadXMLData(xmlData);
}
```

Zerofractal Bogotá

Calle 103 # 21- 60

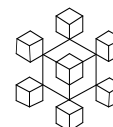
Piso 1

Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>



Usage - Labels

Customizing Labels

You may also, based on the item's XML data create whichever label you want by using the *labelFunction* property as with the custom icons. You may even create new style tags to produce interesting effects.

Example

This code reads the label off an attribute in the xml called title.

```
function menuLabelFunction(item:Object):String {
    return item.data.@title;
}
tree3D.labelFunction = menuLabelFunction;
```

Zerofractal Bogotá

Calle 103 # 21- 60
Piso 1
Bogotá, Colombia
T (031) 2566763

bogota@zerofractal.com
<http://www.zerofractal.com>

About the item:Object: Many functions and events make use of the item Object. This object brings useful information such as the XML node and its attributes, its depth and in some cases even a reference to the *MovieClip* for that item in the tree within the tree component.

Example:

```
function onClick(e:com.zerofractal.tree3D.tree3DItemEvent) {
    if (e.item.data.name == "document") {
        trace("Clicked on document: ", e.item.data.attribute("name"));
    }
}
tree3D.addEventListener(tree3DItemEvent.ITEM_CLICK,onClick);
```

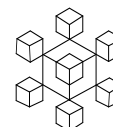
Customizing Label Style

For labels you can customize both its style and the content. To customize the style, simply create a style sheet. Labels are drawn using the "label" tag.

Example:

```
var myLabelStyle:StyleSheet = new StyleSheet();
var labelStyle:Object = new Object();
labelStyle.fontFamily = "_typewriter";
labelStyle.fontSize = 17.5;
labelStyle.color = "#FFFFFF";
myLabelStyle.setStyle("label", labelStyle);

tree3D.labelStyle = myLabelStyle;
```



Usage - Icons

Customizing Icons with symbols from library

Icons and Labels can be easily customized by using the *iconFunction* property. This way you can create a simple handler that is called upon when each item is about to be created.

For icons, you can look at the XML data for that item and decide the name of the symbol from your library you will use for the icon. Icon symbols must be “exported for actionsript”, and be contained within 128x128 pixels. If you want to use an alternate size, set the *iconWidth* and *iconHeight* properties accordingly.

Example:

This example uses the icon attribute for the name of the symbol in the library

```
function treelconFunction(item:Object):String {
    return item.data.@icon;
}
tree3D.iconFunction = treelconFunction;
```

Customizing Icons with external images

You can also use external images for each icon in the tree. To do so, you must assign an empty movie in the *iconFunction* and then make use of the item Added Event. This event is fired each time an icon symbol is actually created and added to the tree. There you can use a loader to load the image. We recommend png over jpg icons due to their transparency. The name of the image may com from an attribute in the XML. As with the symbols, the size of the image must be contained within 128x128 pixels. If you want to use an alternate size, set the *iconWidth* and *iconHeight* properties accordingly.

Example:

This example uses the icon attribute for the name of the external image

```
import com.zerofractal.tree3D.tree3DItemEvent;

function menulconFunction(item:Object):String {
    return "tree3D.icon.empty";
}
tree3D.iconFunction = menulconFunction;

function onAdded(e:tree3DItemEvent) {
    var iconLoader:Loader = new Loader();
    iconLoader.contentLoaderInfo.addEventListener(Event.COMPLETE, onLoadIcon);
    iconLoader.load(new URLRequest(e.item.data.@icon));
    var iconMovie:MovieClip = e.item.movie;
    function onLoadIcon(e:Event):void {
        var iconBitmap:Bitmap = new Bitmap(e.target.content.bitmapData);
        iconMovie.addChild(iconBitmap);
    }
}
tree3D.addEventListener(tree3DItemEvent.ITEM_ADDED,onAdded);
```

Zerofractal Bogotá

Calle 103 # 21- 60

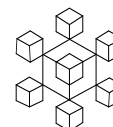
Piso 1

Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>



Customizing Icons with external SWF MovieClips

Likewise, you can use an external swf as an icon. The swf will be rendered, and may be animated, but cannot contain interactive elements, such as buttons, text fields, etc. The interaction is handled by Tree3D events.

Example:

This example uses the icon attribute for the name of the external swf file

```
import com.zerofractal.tree3D.tree3DItemEvent;

function menuIconFunction(item:Object):String {
    return "tree3D.icon.empty";
}
tree3D.iconFunction = menuIconFunction;

function onAdded(e:tree3DItemEvent) {
    var iconLoader:Loader = new Loader();
    e.item.movie.addChild(iconLoader);
    iconLoader.load(new URLRequest(e.item.data.@icon));
}
tree3D.addEventListener(tree3DItemEvent.ITEM_ADDED,onAdded);
```

Zerofractal Bogotá

Calle 103 # 21- 60
Piso 1
Bogotá, Colombia
T (031) 2566763

bogota@zerofractal.com
<http://www.zerofractal.com>

Animating Icons

You may create animateable items using Tree3D by listening to the events from the component. The event object contains a reference to the item's movieclip instace. This way you may use any animation technique (keyframes, tweens, etc).

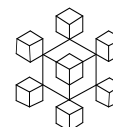
Example:

```
import com.zerofractal.tree3D.tree3DItemEvent;

function onAdded(e:com.zerofractal.tree3D.tree3DItemEvent) {
    e.item.movie["iconType"].gotoAndStop(e.item.data.attribute("type"));
}
tree3D.addEventListener(tree3DItemEvent.ITEM_ADDED,onAdded);

function onOver(e:com.zerofractal.tree3D.tree3DItemEvent) {
    if(e.item.data.children().length() > 0){
        e.item.movie.gotoAndPlay("folderOpen");
    }
}
tree3D.addEventListener(tree3DItemEvent.ITEM_OVER,onOver);

function onOut(e:com.zerofractal.tree3D.tree3DItemEvent) {
    if(e.item.data.children().length() > 0){
        e.item.movie.gotoAndPlay("folderClose");
    }
}
tree3D.addEventListener(tree3DItemEvent.ITEM_OUT,onOut);
```



Breadcrumbs

Using Breadcrumbs

Breadcrumbs are a navigate-able text representation of the tree's hierarchy. Due to the nature of the 3D Tree system, breadcrumbs become an essential aid in terms of user experience.

The developer may link any *TextField* object anywhere in the stage as a breadcrumb. To do so simply assign the breadcrumb property in the *Tree3D* instance.

Customizing Breadcrumbs

By default breadcrumbs print the current tree hierarchy based on the name attribute of each node and shows the total amount of descendants of the current node like this:

```
node1/subnode2/subnode3 (10 items)
```

If you want to customize either the text of the breadcrumb or the descendant summary, you can interface to the *breadcrumbFunction* and the *breadcrumbSuffixFunction*. Here you can specify which field or even what to write at the end of the breadcrumb.

Example:

```
function breadcrumbFunction(item:Object):String {
    return item.data.attribute("Title");
}
tree3D.breadcrumbFunction = breadcrumbFunction;

function breadcrumbSuffixFunction(item:Object):String {
    return "(" + item.data.descendants().length() + " Sub Pages)";
}
tree3D.breadcrumbSuffixFunction = breadcrumbSuffixFunction;
```

Zerofractal Bogotá

Calle 103 # 21- 60

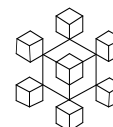
Piso 1

Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>



Events

Detecting Item clicks

You can easily handle item clicks by listening to the import *item_click* event.

Example:

This code makes the tree launch a website whose url is stored in the url attribute of the XML

```
import com.zerofractal.tree3D.tree3DItemEvent;

function onClick(e:com.zerofractal.tree3D.tree3DItemEvent) {
    navigateToURL(new URLRequest(e.item.data.@url), '_blank');
}
tree3D.addEventListener(tree3DItemEvent.ITEM_CLICK,onClick);
```

Zerofractal Bogotá

Calle 103 # 21- 60

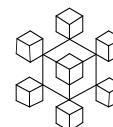
Piso 1

Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>



Others

Using Flash UIScrollbar instead of tree3D's embedded scrollbar

You can easily change the scrollbar component to a UIScrollbar by interfacing to Tree3D's *item_change* event and UIScrollbar's *scroll* event.

Example:

```
function onChange(e:tree3DItemEvent) {
    if (tree3D.itemsPerPage < e.item.data.children().length()){
        scrollbar.enabled = true;
        scrollbar.maxScrollPosition = e.item.data.children().length()-menu01.itemsPerPage;
        if (e.item.startIdx != int(scrollBar01.scrollPosition)) {
            scrollbar.scrollPosition = e.item.startIdx;
        }
    } else {
        scrollbar.enabled = false;
    }
}

function onScroll(e:ScrollEvent) {
    tree3D.scrollTo(e.position);
}

tree3D.addEventListener(tree3DItemEvent.ITEM_CHANGE,onChange);
scrollBar01.addEventListener(ScrollEvent.SCROLL,onScroll);

tree3D.enableScrollbar = false;
```

Zerofractal Bogotá

Calle 103 # 21- 60

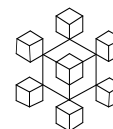
Piso 1

Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>



Credits

- **Tree3D** is based on an idea by Alejandro Gonzalez
- **Tree3D** uses the great opensource **papervision3D** engine for AS3. You are encouraged to visit the PV3D official site at <http://blog.papervision3d.org/>
- **Tree3D** also uses opensource **tweener**. <http://code.google.com/p/tweener/>

Zerofractal Bogotá

Calle 103 # 21- 60

Piso 1

Bogotá, Colombia

T (031) 2566763

bogota@zerofractal.com

<http://www.zerofractal.com>