# Export 184 Plug-in for Autodesk® 3ds Max®

**Version 1.7**

# Reference Manual

**Digital Element**

www.digi-element.com



This manual is intended as a user guide for the Export 184 Plug-in for Autodesk® 3ds Max®. Although reasonable care has been taken in preparing the contents of this manual, it may contain omissions, technical inaccuracies or typographical errors. Digital Element, Inc. does not accept responsibility of any damages resulting from the use of this manual or from the Export 184 Plug-in for 3ds max. Product specifications are subject to change without notice.

# Export 184 Plug-In

The Export 184 Plug-in for Autodesk® 3ds Max® supports producing content for mobile devices using the JSR-184 format. JSR-184 is a Java™ standard used to create 3D applications for mobile phones. Once you have exported your scene to JSR-184 format using the Export 184 plug-in, you can preview the animation using the M3G Player shipped with the plug-in.

# Installation

The Export 184 plug-in runs on Autodesk 3ds Max v.6 or better and works best with any system that meets the recommended requirements for Autodesk 3ds Max v.6.

Export 184 is easy to install. Two steps are involved: (1) Installing the software, and (2) registering the product.

If you have purchased a CD, insert it into the drive, and the program should auto-load. If it doesn't start automatically, click Start, then click Run on the desktop taskbar, and type in `X:\autorun.exe`, where X is your CD-ROM drive letter. If you have purchased the downloaded version, go to the directory you downloaded it into and double-click `InstallExport184.exe`.

After you click `InstallExport184.exe`, an installation dialog should appear. Follow the on-screen installation instructions. The serial number for your copy of the product is on the back of the CD case if you purchased the retail version, or it was sent to you in e-mail if you purchased the downloaded version. Enter the serial number, including the dash.

# Using Export 184

**File menu > Export > JSR-184 (*.M3G)**

**File menu > Export Selected > JSR-184 (*.M3G)**

## Procedures

Before actually exporting objects to an `.M3G` file, the Export 184 plug-in prepares the list of included objects, which displays instantly in the JSR-184 Scene tree control.

The plug-in keeps the state of the Export 184 dialog during the current 3ds Max session. If you reopen the dialog, it shows the same Export Settings and JSR-184 Scene tree as when you closed it.

> Please note that some procedures will be explained separately, in "Editing JSR-184 Object Parameters" below.

### To export an M3G file:

- Choose File menu > Export. If you wish to export particular objects in your scene, use File menu > Export Selected.
- Choose JSR–184 (*.M3G) in the Files Of Type list.
- Specify a file name to export, and then click Save.
  This opens the Export 184 dialog box, displaying the original scene hierarchy in the 3ds max Scene tree. By default, elements of your scene are also listed for export in the JSR-184 Scene tree, depending on how the plug-in is called:
    - If you use the Export menu command, all elements are listed.
    - If you use Export Selected - the selected elements are listed, each with its own hierarchy

> Note that only the exportable elements are listed; elements which cannot be exported by the Export 184 plug-in are not included in the JSR-184 Scene tree.

- Edit the JSR-184 Scene tree as described below (delete the objects you do not wish to export, add the desired objects).

> If an object is a descendant within a hierarchy, it will be exported with its hierarchy. The same applies to objects in a group.

- Modify the parameters of objects listed for export to JSR–184. See "Editing JSR-184 Object Parameters" below for more info.
- Set the active camera as described below.
- In the Export 184 dialog, click Export. The export is made as the dialog disappears.
- Preview your exported JSR-184 scene in the M3G Player.

### To add an object from the 3ds Max Scene to the JSR-184 Scene:

- Drag the desired object from the 3ds Max Scene tree and drop it onto the "target" object in the JSR-184 Scene tree. The new entry (with its hierarchy, if any) is added under the object where you dropped it.

> As you drag an item, the mouse cursor changes to indicate where you can release it.

You may want to choose another procedure to add objects:

- Select the "target" object in the JSR-184 Scene tree, and then double-click the object in the 3ds Max Scene tree. The **3d max** object will be linked as a child to the selected JSR-184 object (sometimes double-clicking has no effect - see note below). This feature is useful when you need to add multiple objects from 3ds Max to one JSR-184 object.

> Note that some object types will not allow you to add child objects (for example, in JSR-184 you may not include objects into a Mesh).

**To delete an object from the JSR-184 Scene:**

- Select the object in the JSR-184 Scene tree.
- Click  (the "Remove Object" button). A dialog box is displayed, prompting you to confirm or cancel your delete request.
- In the dialog, set the Delete child objects check box:
    - Set this option to remove the selected object and its child objects.
    - Clear the option to retain the children, so the child objects will be moved up to the parent of the deleted object.
- Click OK to delete the object or Cancel to discard changes.

> Note that:
>
> - Objects of some types cannot be removed (for example, a JSR Data File root object).
> - Some objects can be removed only by removing their parent objects.
>
> The Delete button is disabled, when the selected object in the JSR-184 Scene tree cannot be removed.

**To create a new JSR-184 Scene that includes all objects from the 3ds Max Scene:**

- Click  (the "New JSR-184 Scene" button) to create a new empty JSR-184 scene.
- Click  (the "Add 3ds max Scene" button). Then all objects of the current 3ds Max scene, which can be converted to the JSR-184 file format, will be added to the JSR-184 Scene tree for later exporting to an `.m3g` file. Each object may be initialized with default parameters and gets a unique ID in the JSR-184 scene

> The behavior of the "Add 3ds max Scene" command may vary from above - see "To update the existing JSR-184 Scene from the 3ds max Scene" below.

**To update the existing JSR-184 Scene from the 3ds Max Scene:**

- Click  (the "Add 3ds max Scene" button). Then:
    - New objects (those convertible to the JSR-184 file format) added in 3ds Max since the last update will be added to the JSR-184 Scene.
    - For "old" objects, which have been included into the current JSR-184 scene earlier, their existing JSR parameters will remain unchanged.

> Caution: The JSR data will be lost if you save your 3ds Max scene on a computer where the Export 184 plug-in is not installed.

**To add a new World object to the JSR-184 Scene:**

This command allows you to create JSR-184 scenes with multiple rendering roots:

- Click  (the "Add JSR-184 World Object" button). This adds a new World object under the topmost object of a JSR-184 scene - under the JSR-184 Data File object.

**To set the active camera:**

This command allows you to set the active camera, and so to define the initial position of the viewer in the scene, which the M3G Player needs to render `.m3g` files:

- In the JSR-184 Scene tree, go to the desired World object.
- Under the World object, select its Active Camera sub-object (this sub-object is labeled as "[Active Camera]" in the JSR-184 Scene tree).
  The Parameters list switches to show the only parameter of the Active Camera:
    - Its Name is "Use Camera"
    - Its Value drop-down list box offers a list of all cameras available for the current World object

- In the Value drop-down list, select the desired Camera object. If you do not want to set a camera to this World, then select the default "None" value.

> When you create an image file using the Export menu command, the plug-in automatically includes a World object and its respective cameras into the JSR-184 Scene.
>
> When you edit the JSR-184 Scene tree manually, you are responsible for providing World objects with appropriate cameras.

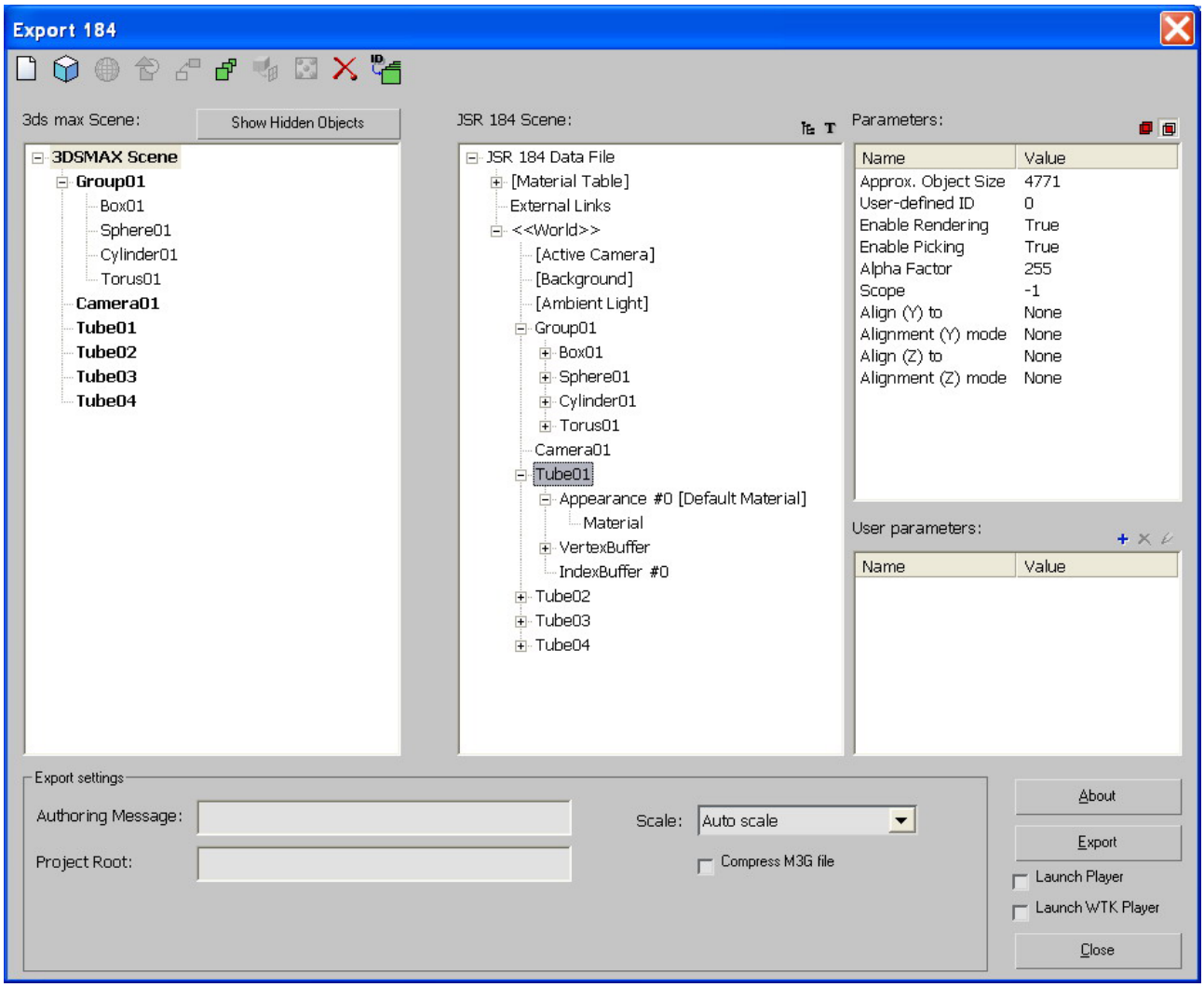
**To add a new Group object to the JSR-184 Scene:**

This command allows you to create groups in JSR-184, used to show the original 3ds Max hierarchy:

- In the JSR-184 Scene tree, select the desired "parent" object.
- Click  (the "Add JSR-184 Group Object" button). This adds a new Group object under the selected parent.

# Interface

## *Export 184 Window*




The original 3ds Max scene hierarchy is shown on the left of the dialog, and the hierarchy of the JSR-184 scene to be exported is displayed in the center of the dialog. Object parameters for the JSR–184 scene are displayed on the right-hand side of the JSR-184 Export dialog. The exported m3g file contains all of the objects and parameters included in the JSR-184 scene.

Many of the parameters for the objects in the JSR-184 scene can be modified prior to exporting the file. See below "Editing JSR-184 Object Parameters" for more details.

### Toolbar

 **New JSR-184 Scene:** Creates an empty JSR-184 Scene tree to collect objects for an `.m3g` file.

 **Add 3ds Max Scene:** Adds all objects with the scene hierarchy from the current 3ds Max scene to the JSR-184 Scene tree. The behavior of this command depends on the history of the scene - new objects will be added to the scene, and for the existing objects their JSR parameters will remain unchanged.

See also about how "To create a new JSR-184 Scene that includes all objects from the 3ds max Scene" and "To update the existing JSR-184 Scene from the 3ds max Scene" above.

 **Add JSR-184 World Object:** Adds a new JSR-184 World object to the JSR-184 Scene tree. A World object can contain a hierarchy of sub-objects. Among other sub-objects, each World object has 3 special sub-objects, which allow setting such properties of the World as its Active Camera, Background, and Ambient Light.

 **Add JSR-184 Group Object**: Adds an empty JSR-184 Group object to the JSR-184 Scene tree. You can use Group objects if you want to show scene hierarchy.

 **Convert Mesh to Sprite3D:** By default, the plug-in exports Plane objects from 3ds Max to JSR-184 Mesh objects. You can further convert such Meshes to JSR-184 Sprite3D objects: select such a Mesh in the JSR-184 Scene tree, and click the Convert Mesh to Sprite3D button.

> This applies to Plane objects initially consisting of two triangles (that is, for such a Plane its Length Segs and Width Segs parameters are set to 1).

 **Add an External Link:** Adds a new External Link entry into an object, selected in the JSR-184 Scene tree (note that only the External Links section, World, and Group objects may contain external links). By editing parameters of the added link, you can further enter the location of the file you want to refer to.

> As well as "intrinsic" objects, an `.M3G` file may include external objects, which are links to `.PNG` files, and even to other `.M3G` files. This allows you to create "object libraries" from sets of files, and use such libraries to optimize your JSR-184 scenes.

 **Add LOD Modifier:** Adds a LOD (Level of Detail) modifier to a Mesh object, selected in the JSR-184 Scene tree. To construct a set of meshes for different levels of detail, drag meshes from the 3ds Max Scene tree and drop them onto the LOD modifier in the JSR-184 Scene tree. See also "Mesh Objects" and "JSR-184 Extension" sections below.

 **Texture Tool**: Invokes the Texture Tool dialog box to edit properties of a texture, currently selected in the JSR-184 Scene tree. For details, see "JSR-184 Texture Tool" section below.

 **Remove Object:** Removes the selected item from the JSR-184 Scene tree. Note that some objects can be removed only by removing their parent objects.

 **Assign User ID:** Automatically assigns unique User ID values to all exported objects.

### 3ds max Scene Tree

Shows hierarchy of the original 3ds Max scene objects.

Named Selection sets, if any, are listed immediately below the root `3DSMAX Scene` item in the 3ds Max Scene tree. You can add a Named Selection set as a group to your JSR-184 scene. All items of a Named Selection Set are added with their scene hierarchy.

**Show Hidden Objects** - Shows all the hidden objects in the 3ds Max scene hierarchy.

### JSR-184 Scene Tree

Lists the scene objects to be exported to an `.M3G` file. The JSR-184 Data File is a root object in the JSR-184 Scene tree.

> You can select and deselect multiple objects with the **Ctrl** or **Shift** key.

### Parameters List

Shows parameters, if any, of the object currently selected in the JSR-184 Scene tree. For details, see "Editing JSR-184 Object Parameters" below.

### User Parameters List

Shows the user-defined parameters of the selected object. For details, see "Editing JSR-184 Object Parameters" below.

### Export Settings Group

**Authoring Message**—This message is required by the JSR-184 file format standard and is usually reserved for entering copyright notice information. This message is stored in the header portion of the M3G file.

**Project Root**—Sets the location of the root of the project. If you do not enter a path in this field, all external references within the `M3G` file are loaded from the same location as the current file. If you have entered a string, such as
`http://www.gamesforfun.com/objs`,
then external references are loaded from locations named like
`http://www.gamesforfun.com/objs/texture1.png`.

**Compress M3G File**—Compresses all the data in the exported M3G file.

**Scale Combobox**—Allows to select one of three scaling options:

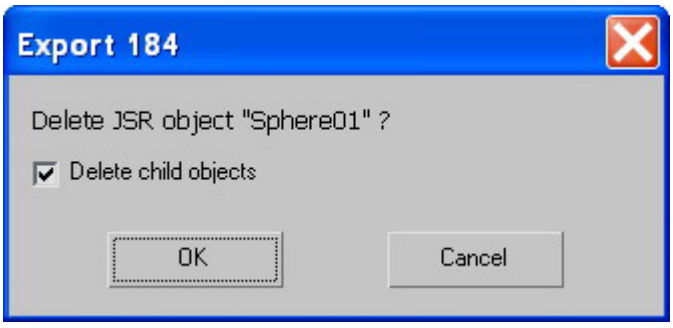
- **No scale**: do not perform any scaling;
- **Auto Scale**: automatically scales all dimensions in the JSR-184 scene within 0 to 32767 range;
- **Smart Scale**: saves all scaling information directly in each object;

### Launch Player and Launch WTK Player Check Boxes

If the Launch Player and/or the Launch WTK Player check box are selected, then respectively the M3G Player and/or M3G WTK Player are launched automatically to preview the exported `.m3g` file when export is completed.

## *Dialogs*

### Remove Object Dialog Box




This dialog box is opened when you click the Remove Object button.

**Delete child objects:** Select this option to remove the object highlighted in the JSR-184 Scene tree, and its child objects. Clear the option to retain the children, so the child objects will be moved up to the parent of the deleted object.

# Editing JSR-184 Object Parameters

**File menu > Export > JSR-184 (*.M3G).**

For the JSR-184 scene shown in the JSR-184 Export dialog, you can adjust many of the parameters of the scene's objects. Object parameters, if any, are shown in the Parameters list on the right side of the JSR-184 Export dialog.

The JSR-184 objects and their editable parameters are detailed in Appendix below.

The JSR-184 objects inherited from Object3D may also have user parameter data. By using such parameters, you can extend capabilities of the Export 184 - see also an example in "JSR-184 Extensions" section in this document. You can add, edit and delete user-defined parameters.

> For further details on the JSR-184 objects and parameters, see also the respective JSR-184 documents, such as "Mobile 3D Graphics API Specification".

## Procedures

When you have the Export 184 dialog opened (see "Exporting JSR-184 Files" above), you can access parameters of the scene's objects.

### To edit a parameter of an object:

- Select the object in the JSR-184 Scene tree
- In the Parameters list, select the parameter by its Name and edit its Value property

> JSR-184 objects and their parameters are described in Appendix below.
>
> You cannot edit certain parameters, for example, *Approx. Object Size, File Size*, and *Approx. File Size*, which show the size of an object with its sub-objects.

### To create a user-defined parameter:

- Select the object in the JSR-184 Scene tree
- Above the User Parameters list, click  to open the Add New User Parameter dialog box
- In this dialog, set the Parameter ID, Type, and Value properties for the new user parameter

### To edit a user-defined parameter:

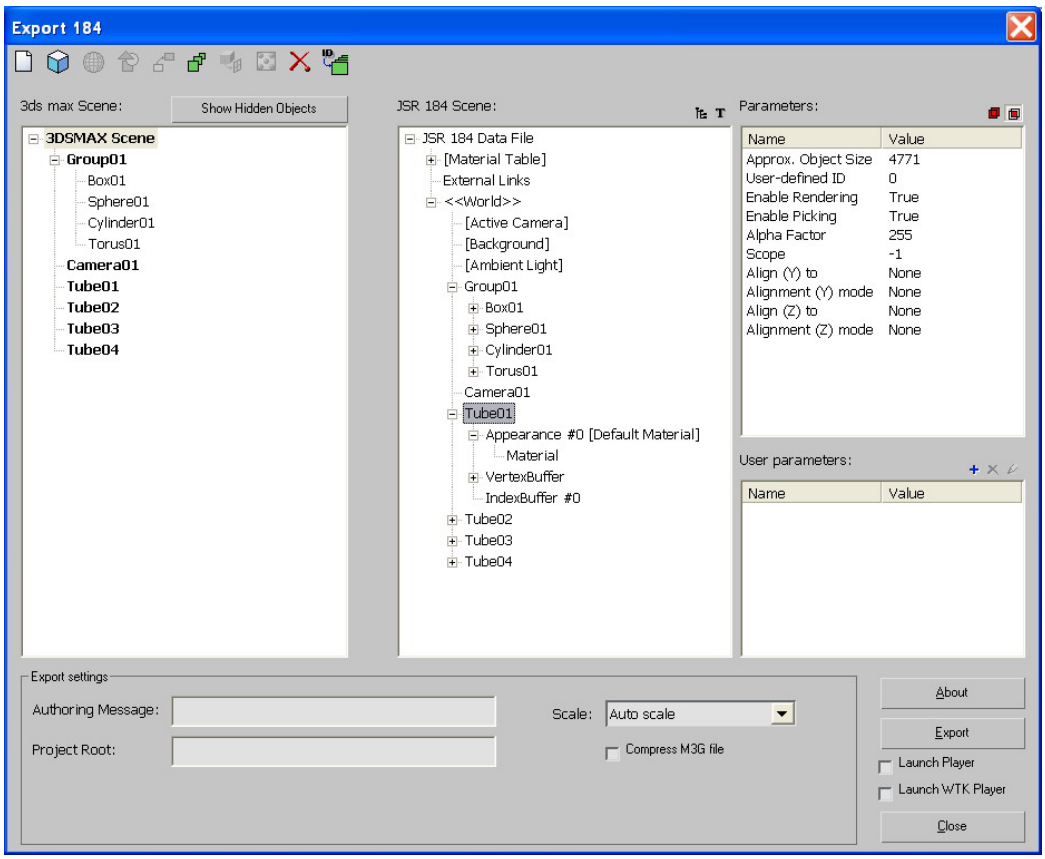
- Select the object in the JSR-184 Scene tree
- In the User Parameters list, select the parameter by its Name (Parameter ID)
- Above the User Parameters list, click 
- In the opened dialog box, edit the Parameter ID, Type, and Value properties

### To delete a user-defined parameter:

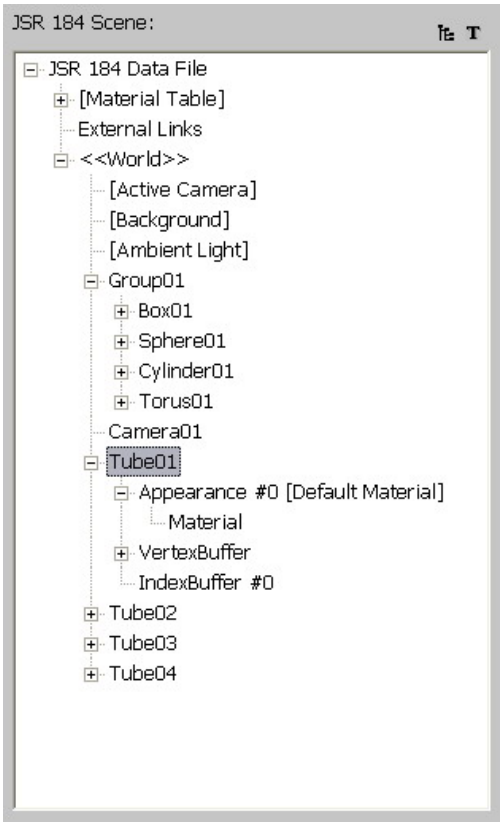
- Select the object in the JSR-184 Scene tree
- In the User Parameters list, select the parameter by its Name (Parameter ID)
- Above the User Parameters list, click  to remove the selected parameter

# Interface

## *Export 184 Main Window*




### JSR-184 Scene Tree



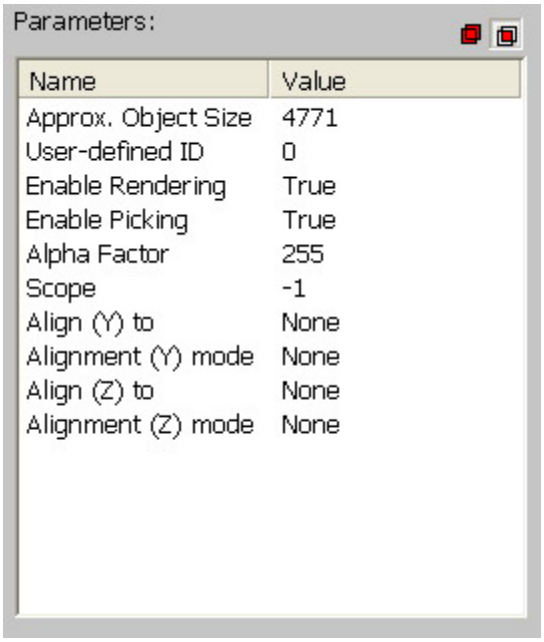

A JSR-184 scene is represented in the center of the JSR-184 Export dialog. The JSR-184 Data File is a root object in the JSR-184 Scene tree.

Two buttons above the tree can help you to select multiple objects:

Recursive Selection - When this option is on, a click on an object in the JSR-184 Scene tree will select this object and all child objects.

Selection by Type - When this option is on, a click on an object in the JSR-184 Scene tree will select this object and also all the other objects of the same type.

### Parameters List




Parameters, if any, of the selected object or multiple objects are shown in the Parameters list on the right side of the JSR-184 Export dialog.

Objects and their parameters are described in Appendix below.

When multiple objects are selected, two buttons above the list can help you to filter the displayed parameters. Hidden parameters disappear from view, making it easier to work with remaining items.

> When only one object is selected, the buttons do not (and cannot) affect the Parameters list.

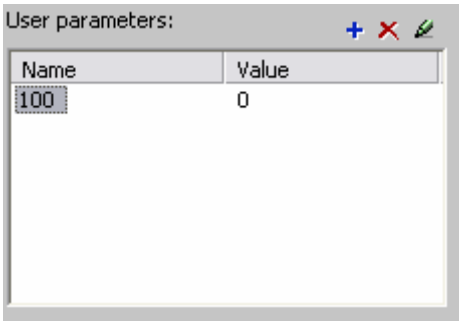
Filtering is based only on the names of the parameters; belonging to an object or parameter values are ignored.

Shows all the parameters of the selected objects.

Shows only the common parameters, that each of the selected objects has.

> Tip: Use the buttons to propagate the same setting values to multiple objects.

### User Parameters List




The User Parameters list on the right side of the JSR-184 Export dialog shows the user-defined parameters of the selected object.

Three buttons above the list allow you to manage the user parameters.

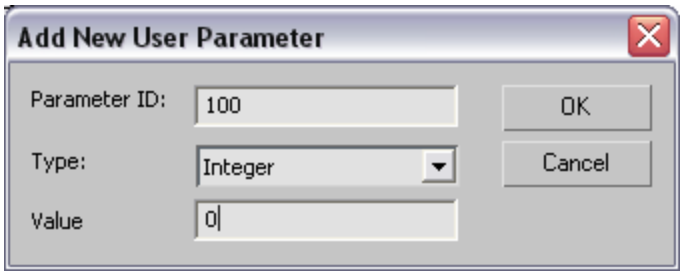
Opens a special Add New User Parameter dialog box (see below) to add a new User Parameter to the currently selected object in the JSR-184 scene tree.

Removes the selected user parameter.

Opens an additional dialog box (see below) to edit the selected user parameter.

## *Dialogs*

### Add New User Parameter Dialog Box




This dialog box is opened when you click buttons at the above list to add or edit a user-defined parameter.

To set its properties, a parameter has the following controls:

**ID:** Shows the ID number (integer) assigned to this parameter. You can edit this field to change the ID number. If you assign two parameters the same ID, the older value is overwritten.

**Type:** Lets you select the type (integer or string type) for the parameter.

**Value:** Lets you enter a custom value for the parameter.

# Using JSR-184 Extensions

The Export 184 plug-in for 3ds Max offers a set of extensions for the JSR-184 file format. All these extensions require additional java code on the mobile device. The following extensions are supported:

- Level of Detail (LOD) Modifier for Mesh objects.
- Locator objects
- Gravity and velocity properties for free fall simulation

See "Programming JSR-184 Extensions" below for more details about implementing this code on the phone.

## LOD Modifier for Mesh Objects

The Level of Detail approach lets you construct an object that alters its geometric complexity, or level of detail, based on its size in the rendered image. You can create several versions of the same Mesh object each with different levels of detail, and group them as one. The LOD modifier defined in the Export 184 plug-in then hides and unhides Mesh objects in the group, depending on the distance to the object in the rendered scene, such as the M3G Player scene.

An object with the LOD modifier includes:

- A primary object used at the closest range, and
- A set of less-detailed sub-objects, each for its specific range.

A sub-object is applied when the distance between it and the camera exceeds a certain positive threshold value set for this level. Levels are automatically ranged by the threshold value.

## Procedures

### Example: To create a Sphere object that includes different levels of detail:

The object should include four sphere meshes with different levels of detail. Depending on the distance, spheres should be selected as follows:

- from 0 to 100 - Sphere01
- from 100 to 200 - Sphere02
- from 200 to 300 - Sphere03
- more than 300 - Sphere04

To create a Sphere object:

- In 3ds Max, select the primary object. In our case, select Sphere01 that should be used at the closest range from 0 to 100.
- Position the primary object in 3D space. Also place objects that should be used as the less-detailed sub-objects of the primary.

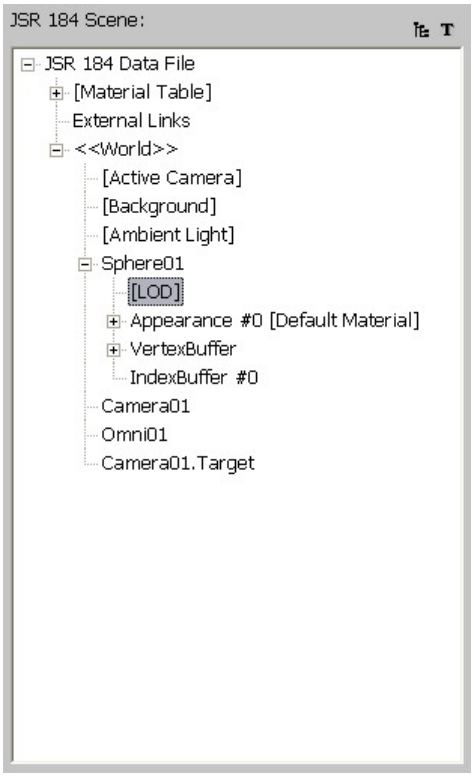
> You must specify an exact origin for the primary object only; you may place the less-detailed sub-objects anywhere in the scene. When switching the level of detail, the LOD modifier will position and unhide each appropriate sub-object at exactly the same coordinates as those of the primary object.
>
> Also, you should only animate the primary object; animations for the less-detailed sub-objects are ignored.

- In 3ds Max, go to the Export 184 plug-in. Create a JSR-184 scene.
  Do not include the less-detailed sub-objects in your JSR-184 scene. Remove these sub-objects, if any.

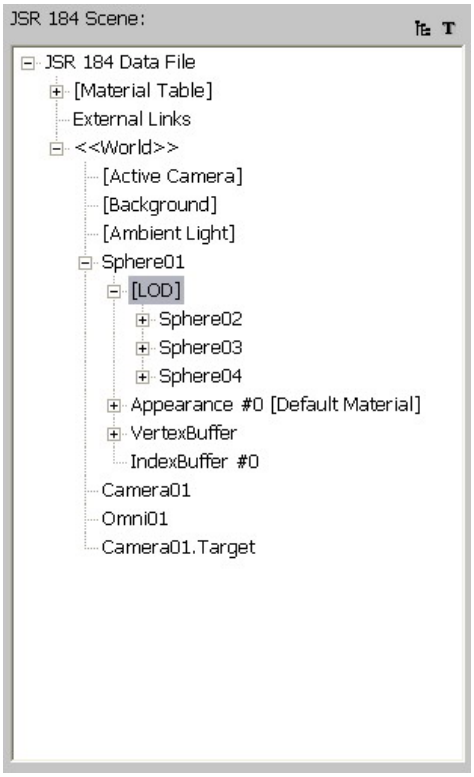
- In the JSR-184 Scene tree, apply the LOD modifier to your primary object:

  select the primary object, and then click  .

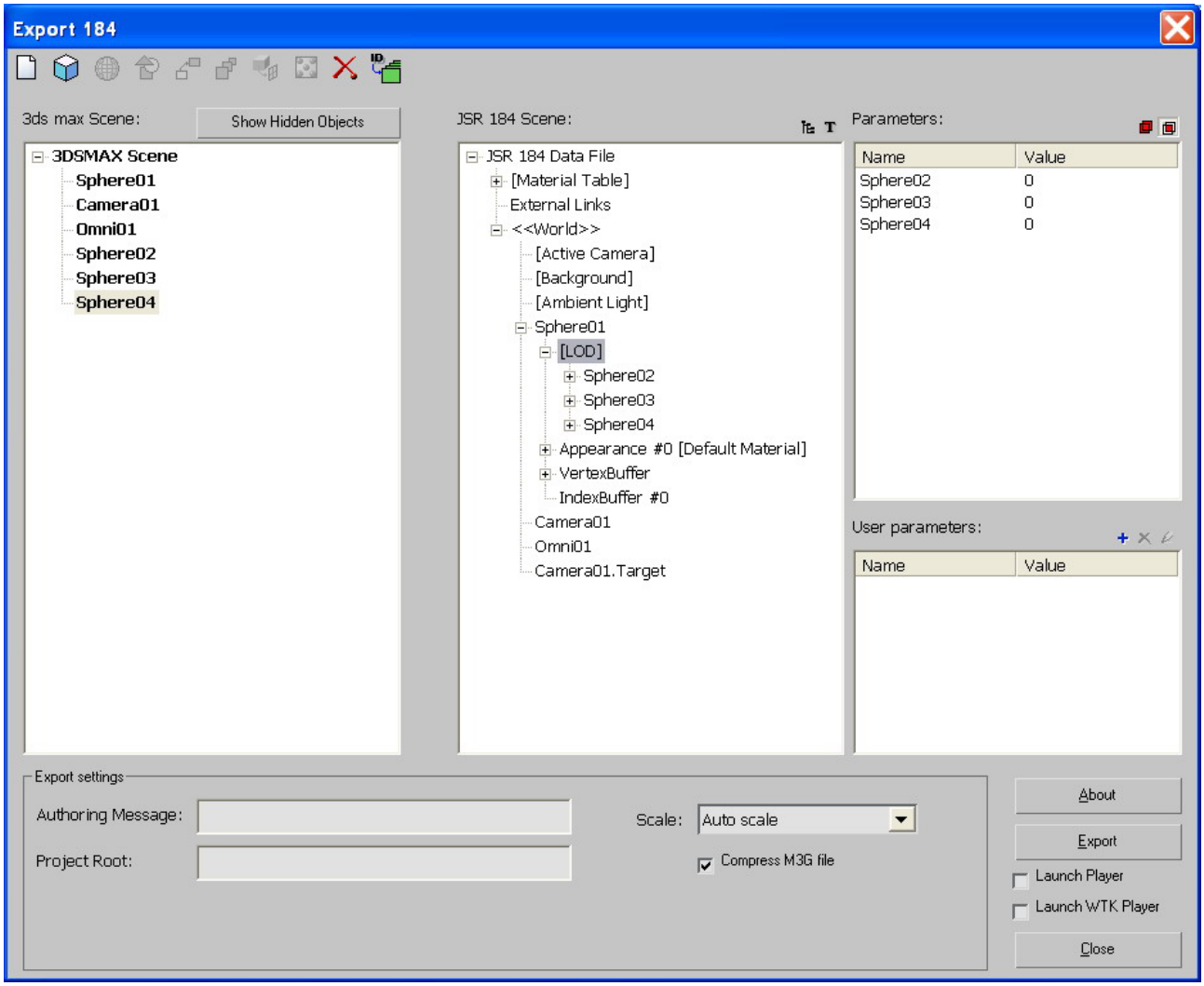
A LOD modifier entry, labeled "[LOD]", appears under the primary object (in our case, under the Sphere01 mesh):



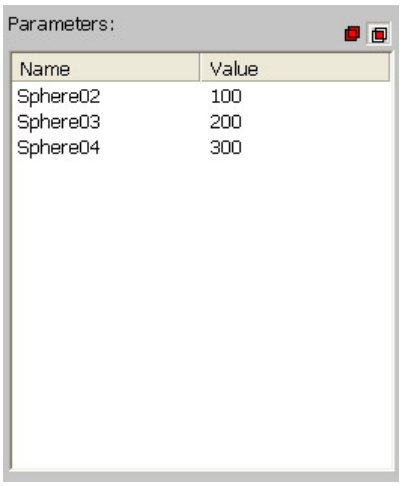

- In 3ds Max scene, select the mesh for the "next level of detail" sub-object (in this case, it is Sphere02). Drag it to the JSR-184 Scene tree, and drop it onto the LOD entry. Repeat the above steps for each less-detailed sub-object (Sphere03, Sphere04):



- For each sub-object added under the LOD modifier, the plug-in assigns their respective LOD parameters. By default, the parameters are set to 0:




- Enter the parameter values specified for this LOD object:




- The Mesh object with the LOD modifier is set up.
- Click the Export button to save the JSR-184 scene to an `.M3G` file.

For details, see the `lod.m3g` and `lod.max` sample files.

# Locator Objects

The Locator extension allows defining locator objects in the 3D space. A Locator Object, among common parameters, has such parameters as:

- A position in 3D space - *locator point*.
- A *radius of action*.

When an active camera (currently set in the JSR-184 scene) comes to the locator point within its radius of action, a special procedure is called. This event can be identified by the user-defined ID assigned to this locator object.

Locators help to handle such event driven actions, as in-game "open this door", "explode this mine", and so on.

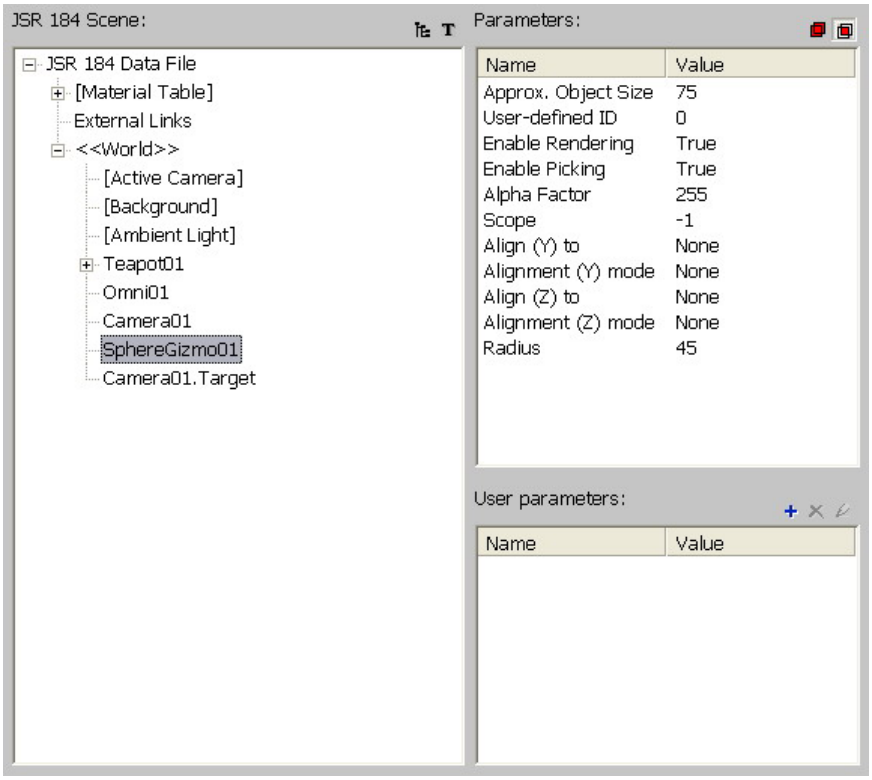
# Procedures

### To create a new Locator object:

- In 3ds Max, create a new SphereGizmo object (by Create > Helpers > Atmospheric Apparatus > SphereGizmo).
- Position the gizmo and adjust its radius. You may also want to rename the gizmo to give it an intelligible name.
- In 3ds Max, go to the Export 184 plug-in. Add the gizmo to the JSR-184 scene.
- Set a unique User-defined ID to identify the locator.

### To change a Locator's radius of action:

- In the JSR-184 Scene tree, select the Locator object.
- In the Parameters list, edit the Radius value.




For details, see the `locator.m3g` and `locator.max` sample files.

# Gravity and Velocity Properties for Free Fall Simulation

This extension by the Export 184 plug-in allows you to simulate a free fall motion of objects in your scene. You can set:

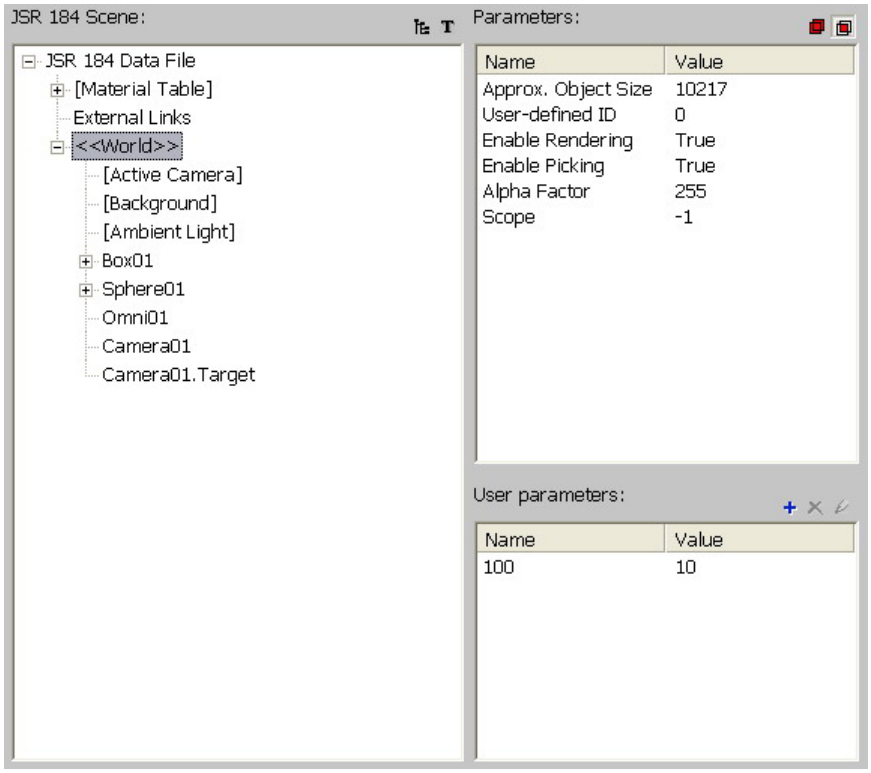
- Acceleration of free fall or Gravitational constant of your World.
- Initial velocities for objects selected in your scene.

In your game scene, each selected object will move with time, so that its 3D position will also depend on its initial velocity vector and Gravitational constant.

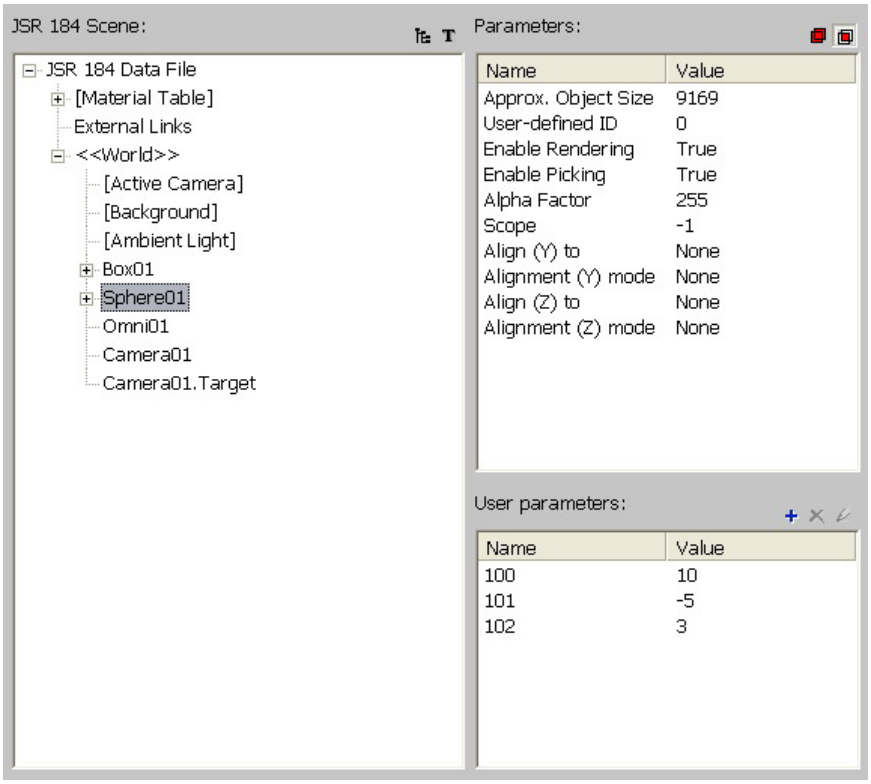
## Procedures

### To start simulation of free fall motion of objects:

- Open an existing 3ds Max scene which you want to add the gravity support feature to.
  For the World Object, define a user parameter:
  - Gravitational constant: set its Name to '100' and its integer Value
    (see figure below, gravity constant set to 10)



- For each Object in the scene you want to apply some initial velocity to, define user parameters:
  - Velocity along X-axis: set Name to '100', set the initial Value
  - Velocity along Y-axis: set Name to '101', set the initial Value
  - Velocity along Z-axis: set Name to '102', set the initial Value
    (see figure below, initial velocity vector is set to (10,-5, 3))




- Export the scene to a JSR-184 file and play the exported M3G file using M3GPlayer.

> Important: The above extensions require additional Java code on your mobile device.
>
> The M3G Player shipped with the plug-in has these additions installed, so it can play extended JSR-184 scenes.

# JSR-184 Texture Tool

**File menu > Export > JSR-184 (*.M3G). > Name your file and click OK. > Select a Bitmap object in the Material Table in your JSR-184 scene. > Click the Texture Tool button to edit the texture properties.**

All the textures applied to the object surfaces are included in the Material Table branch in the JSR-184 Scene tree. The Texture Tool dialog box allows you to edit the parameters and behavior of the textures in your JSR-184 scene. You can modify the parameters described in this topic.

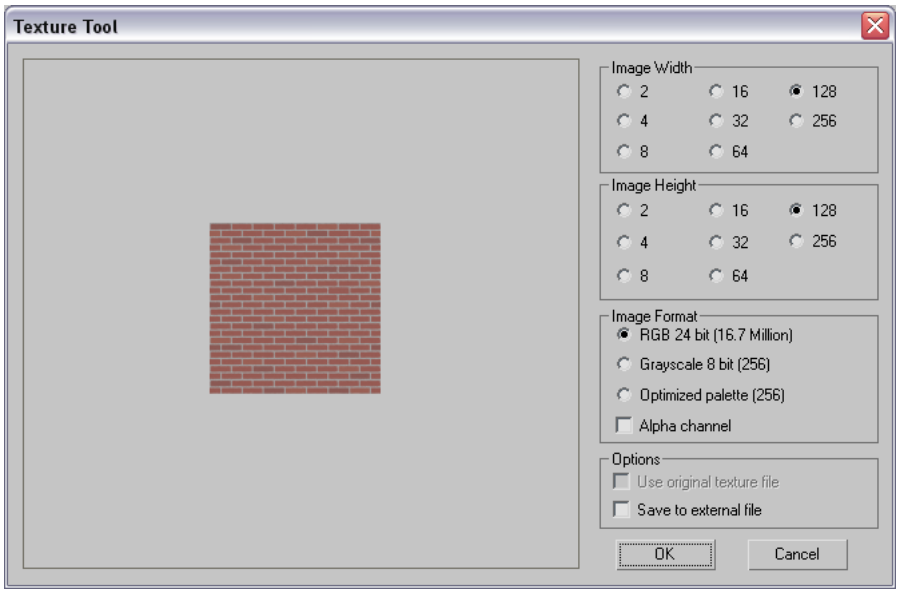
The Texture Tool dialog displays a preview of the texture with its current settings. When you select different options, the preview is updated to reflect the changes.

## Procedures

### To edit a texture:

- In the JSR-184 Export dialog box, go to the Material Table in the JSR-184 Scene tree, and select the texture.
- In the toolbar, click  to open the Texture Tool dialog box.
- In this dialog, set the desired properties for the selected texture.

## Interface




### Image Width Group

The JSR-184 format requires that texture sizes use the "power of two rule", and currently supports texture sizes less than or equal to 256 by 256. You can select 2, 4, 8, 16, 32, 64, 128, or 256.

### Image Height Group

These are the same options as Image Width: you can select 2, 4, 8, 16, 32, 64, 128, or 256.

> For Background textures, you can set any width and height. To enter a width value, the plug-in provides an edit box appearing in the bottom-right corner of the Image Width group. A similar edit box appears in the Image Height group to enter height values.

### Image Format Group

Changes the color model of a texture in your JSR-184 scene.

There are three available options for color depth:

- **RGB 24 bit (16.7 Million)**—Changes the color model to RGB (16.7 Million of colors).
- **Grayscale 8 bit (256)**—Changes the color model to grayscale (256 grayscales).
- **Optimized Palette (256)**—Changes the color model to a limited 256-color adaptive palette.

Also, you can choose to save more color data:

**Alpha Channel**—When on, saves the alpha channel with the texture.

### Options Group

**Use original texture file** — When on, an existing texture file is used "as is", in its original format. Depending on the Save to External File option (see below), the original texture is copied either to a new external file, or immediately into the current JSR-184 scene.

> You cannot edit parameters of a texture added this way.

**Save to External File**—When on, saves the texture as an external file (in `PNG` format) in the same directory as the `M3G` file, and adds a link to this texture file into the JSR-184 Scene tree. If you clear this option, the texture is saved as a part of the `M3G` file.

# Export 184 Log Files

**File menu > Export > JSR-184 (*.M3G).**

When you export an M3G file, the Export 184 plug-in creates a log file. This log file is an HTML file, saved in the same directory as the exported M3G file. Errors, detected during export, cause a message box to appear.

## Interface


**Show export log file**—If set, the log file generated during export opens automatically as you close this message box.

### Log Data

The table contains entries for all the objects exported to the M3G file format.

**Object #**—Indicates the root object value. When multiple root objects exist in the M3G file, this value is used by the JSR-184 Load object to load the correct object.

**Object Name**—Contains the name (if any) of the corresponding 3ds Max object.

**Object Type**—Indicates the type of saved object.

**User ID**—Displays the User ID for the object.

**Object Size**—Shows the object size (in kilobytes) for the object.

**Comments**—Displays additional information such as errors and warnings.

Below is a sample of a simple log file:

| Object # | Object Name | Object Type | User ID | Object Size | Comments |
|---|---|---|---|---|---|
| | Box01 | Vertex Array | 0 | 161 | |
| | Box01 | Vertex Array | 0 | 89 | |
| | Box01 | Vertex Buffer | 0 | 48 | |
| | Box01 | Triangle Strip Array | 0 | 213 | |
| | Box01 | Material | 0 | 30 | |
| | Box01 | Appearance | 0 | 33 | |
| | Box01 | Mesh | 0 | 78 | |

| | | | | | |
|---|---|---|---|---|---|
| | <> | Group | 0 | 66 | |
| | Camera01 | Camera | 0 | 89 | |
| | Omni01 | Light | 0 | 90 | |
| | <> | Group | 0 | 26 | |
| | Light01 | Light | 0 | 50 | |
| | [Background] | Background | 0 | 40 | |
| 0 | <> | World | 0 | 58 | There is no active camera defined. |

# M3G Player

The M3G Player can interpret M3G files (JSR-184 format), so you can preview, how the animation that you have exported from your 3ds Max scenes, is displayed on various mobile screens.

## Installation

- Make sure that Java 1.4.2 or better Runtime Environment is installed on your system.
- The M3G Player is installed by default, when you install the Export 184 plug-in.

To run the M3G Player, from the desktop taskbar (by default) choose

Start > Programs > Digital Element > Export 184 > M3G Player.

# M3G Player Main Window

**Start menu > Digital Element > Export 184 > M3G Player**

This window opens when you start the M3G Player. This allows you to preview how the animations from your 3ds Max scenes are displayed on various mobile screens.

Also you can edit phone profiles (phone models, screen resolutions) used to preview M3G files.

## Procedures

### To preview an M3G animation:

- In the File menu, choose Open. A standard Windows Open dialog box appears.
- In the Open dialog box, enter the path and filename of the M3G file you want to preview. If the file is opened correctly, its filename appears in the window title bar.
- Play the animation using the player control buttons described below in "Player controls".

### To choose phone settings used to preview M3G files:

- In the toolbar, in the drop-down list of phone profiles, click a profile of your choice.

### To add a new phone profile:

- In the Tools menu, choose Handsets. The Handsets dialog box appears.
- In the Handsets dialog box, in its toolbar, click  to open the Handset dialog box.
- In the Handset dialog box, set the properties (such as Vendor, Model, screen width and height, and the color depth) for the new phone profile.

### To edit a phone profile:

- In the Tools menu, choose Handsets to open the Handsets dialog box.
- In this dialog box, select the profile to edit in the list of phone profiles;
  then click  to open the Handset dialog box.
- In the Handset dialog, edit the properties of the selected phone profile.

### To delete a phone profile:

- In the Tools menu, choose Handsets to open the Handsets dialog box.
- In this dialog box, select the profile you want to delete in the list of phone profiles; then click  to remove the selected phone profile from the list.

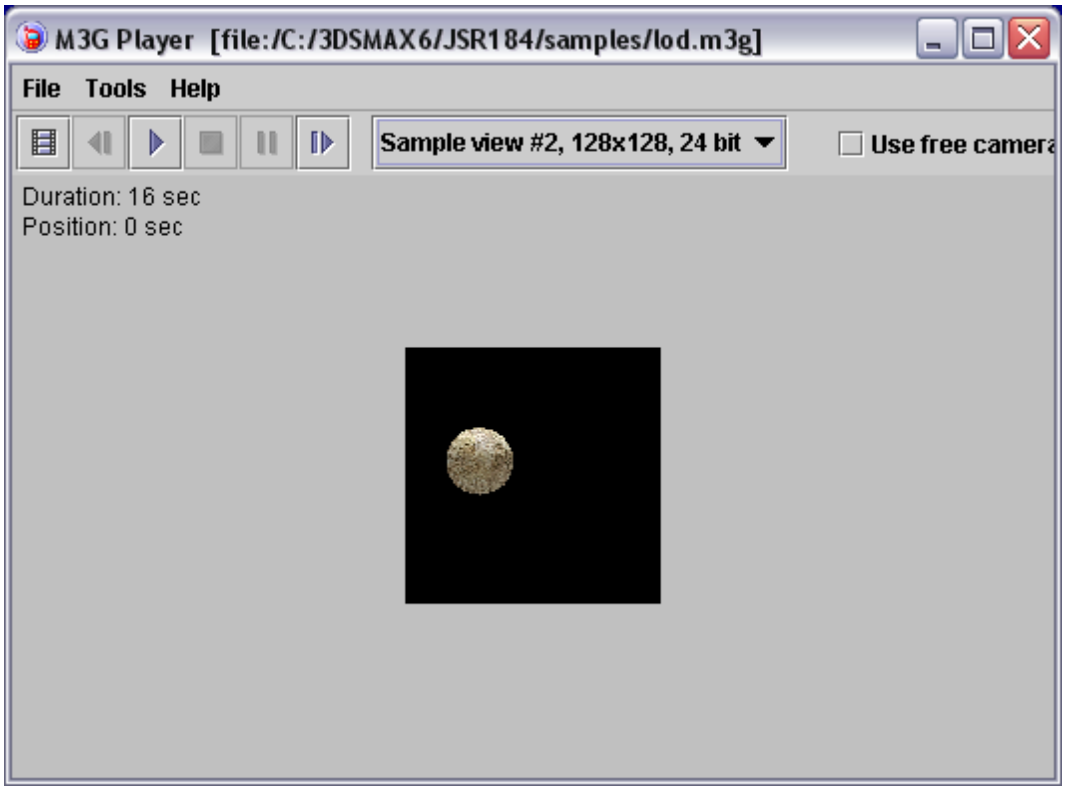
### To control the camera in your scene:

- In the toolbar, select the "Use free camera" check box to enable camera navigation in the JSR-184 scene.
- Move and rotate the camera using the keyboard shortcuts as described below in "Interface".

### To change a rendering root object in the scene:

- In the toolbar, click  to open the "Choose objects for rendering" dialog box.
- In this dialog, in the JSR-184 scene hierarchy, select the World object you want to preview, and click the Set as root button

# Interface




## *Menu Commands*

### File Menu

**Open—** This command allows you to open an M3G file for viewing in the M3G Player.

**Reopen—**Displays a list of recently opened files. The list displays the most recently opened file at the top.

**Exit—**Closes the M3G Player window.

### Tools Menu

**Handsets—**Allows you to add or edit phone profiles. You can modify the vendor, model, screen width, screen height, and screen color depth.

> Note: You can also modify the handset profiles directly in the *terminals.xml* file in the directory, where the M3G Player is installed

## *Toolbar*

The toolbar at the top of the M3G Player window contains:

- A button, which allows you to select the rendering root for your JSR-184 scene.
- Player control buttons, which control playback of the animation.
- Phone profiles, which represent various mobile screens, listed in the drop-down list box.
- A check box, which enables camera navigation in the JSR-184 scene.

### Button to Choose Objects for Rendering

This button is the leftmost on the toolbar:

 Opens the Choose Objects for Rendering dialog box - see below.

This dialog allows you to select a World object in the JSR-184 scene hierarchy when there are multiple world objects in the JSR-184 data file. Then the selected World object and its subordinates are rendered in the M3G Player window.

### Player Controls

The player is controlled by the following buttons on the toolbar:

 "Step backward" - Steps one frame backward through the animation.

 "Play" - Starts playing the animation.

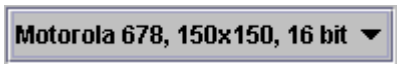
 "Stop" - Stops the animation and returns to the first frame.

 "Pause" - Pauses the animation. Click "Play" to resume.

 "Step forward" - Steps one frame forward in the animation

### Phone Profile List

In this drop-down list box, each phone profile entry represents a mobile screen, available as a target for preview rendering. You can define phone profiles by using the Handsets and Handset dialog boxes - see below.



A phone profile includes the following fields:

- Phone model name (for example, Motorola 678)
- Phone screen width and height (150x150)
- Phone screen color depth (16 bit)

You can choose a phone profile from the drop-down list box at any time.

### Use Free Camera Check Box

When selected, the Use Free Camera check box enables camera navigation in JSR-184 scenes.

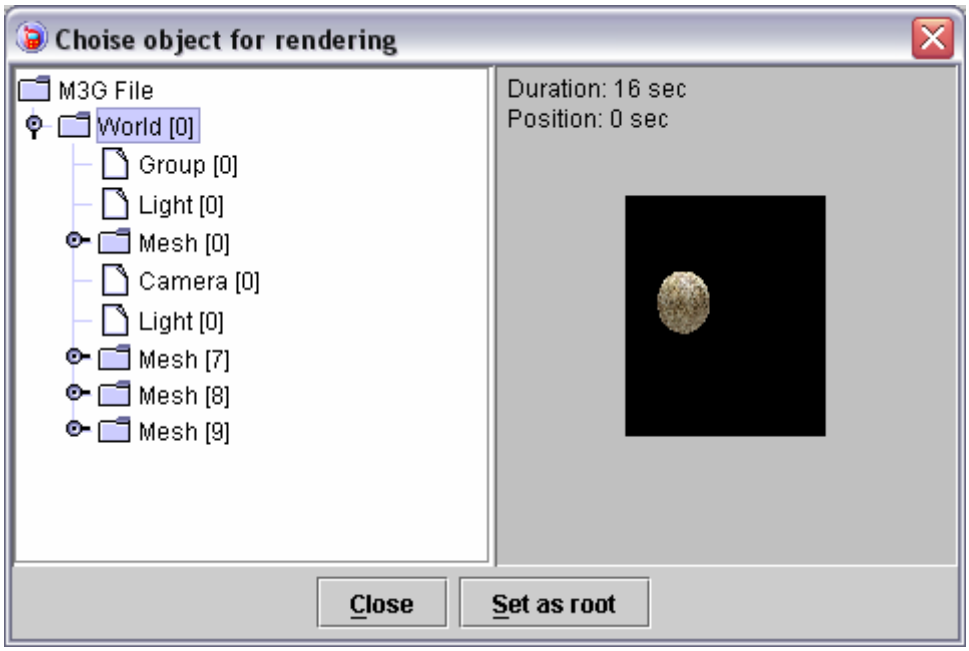
You can control the camera by the following keyboard shortcuts:

| Key | Function |
|---|---|
| Left Arrow | Rotate left |
| Right Arrow | Rotate right |
| Up Arrow | Rotate up |
| Down Arrow | Rotate down |
| A | Move left |
| D | Move right |
| S | Move backward |
| W | Move forward |
| Home | Move camera to default position |

## *Dialogs*

### "Choose objects for rendering" Dialog Box

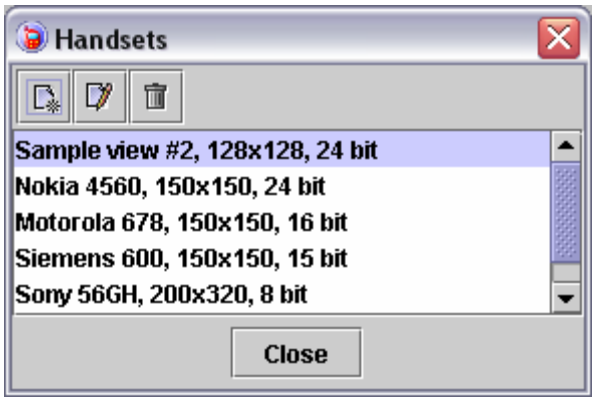
This dialog allows you to select a World object in the JSR-184 scene hierarchy when there are multiple World objects in the JSR-184 data file.




**Set as Root -** Switches the selected World object and its subordinates to be rendered in the M3G Player window.

### Handsets Dialog Box

This dialog allows you to add and edit phone profiles.




In the dialog, mobile screens are listed, currently available as targets for preview rendering in the M3G Player window. You can select an entry in the list to edit or remove the respective phone profile.

The dialog has the following buttons on its toolbar:

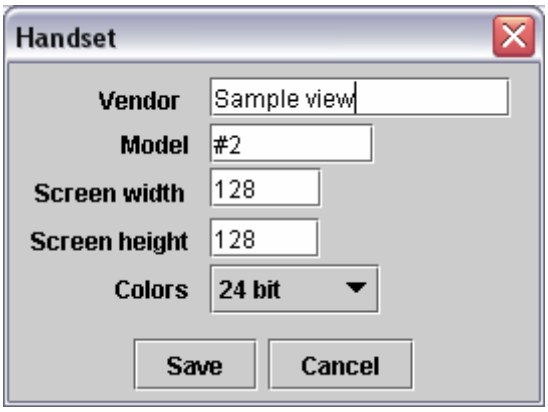
 "New" - Creates a new handset profile, and opens the Handset dialog box (see below) to set properties (such as Vendor, Model, screen width and height, and the color depth) for the new profile.

 "Edit" - Opens the Handset dialog box (see below) to edit properties of the selected profile.

 "Delete" -Removes the selected handset profile

### Handset Dialog Box

This dialog allows you to set properties for a phone profile.




**Vendor**, **Model**, **Screen width** and **Screen height**, and **Colors** (the color depth) - these are the phone properties you can specify here.

**Save** - Accepts your edits.

**Cancel** - Cancels your edits.

# M3G WTK Player

**Start menu > Programs > Digital Element > Export 184 > Export 184 Server,** then
**Start menu > Programs > Digital Element > Export 184 > WTK Player**

or

**Export 184 window > Launch WTK Player check box > Export button**

To preview animations that you export from your 3ds max scenes, the Export 184 plug-in offers another option based on the Sun Java™ Wireless Toolkit (formerly known as Java™ 2 Platform, Micro Edition (J2ME) Wireless Toolkit). Emulators based on the Wireless Toolkit provide an impressive list of features:

- Creating a skin on the toolkit's emulator gives a good approximation of a particular device; you can specify the screen size, color depth, fonts, and input methods of a device.
- The toolkit's emulator provides a robust environment to test applications that don't use vendor-specific APIs.
- The Wireless Toolkit includes helpful tools for the application developer, such as performance simulation, method profiling, and network monitoring.

The M3G WTK Player shipped with the Export 184 plug-in, runs on the Sun Java Wireless Toolkit's emulator. The player can interpret M3G files to display your animations on various emulated mobile screens.

Also, a vendor may write code to emulate the actual processor and other chips on the device. In that case the M3G WTK Player will run on any emulator that conforms to MIDP 2.0 (the Mobile Information Device Profile) and JSR 184 specifications.

# Installation

- Make sure that J2ME Wireless Toolkit 2.2 or better is installed on your system.
- The M3G WTK Player is installed by default, when you install the Export 184 plug-in.

# Launching M3G WTK Player on Emulator

You can start the M3G WTK Player in two ways:

- Start manually using the Windows Start menu
- Start automatically in the Export 184 plug-in

### To start the M3G WTK Player using the Windows Start menu

- In the Start menu, point to Programs, locate the Export 184 Server program, and then click it:

  Start > Programs > Digital Element > Export 184 > Export 184 Server (by default)

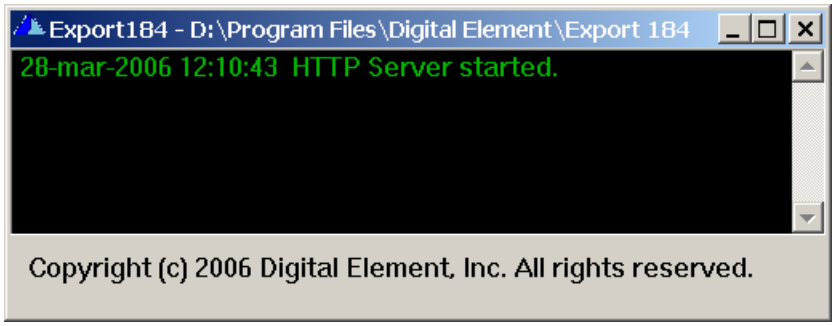
- In the Start menu, point to Programs, locate the WTK Player program, and then click it:

  Start > Programs > Digital Element > Export 184 > WTK Player

### To start the M3G WTK Player from the Export 184 plug-in window

The Export 184 Plug-in for Autodesk 3ds Max can automatically launch the M3G WTK Player for you. If the Launch WTK Player option is selected in the Export 184 plug-in window, the emulation environment and the player will start when you click the Export button.

- In the Export 184 plug-in window, select the Launch WTK Player check box
- Click Export. The exported objects are saved as an `.m3g` file, and also the Export 184 Server starts. As its working directory, the server applies the folder, where the `.m3g` file has been saved.

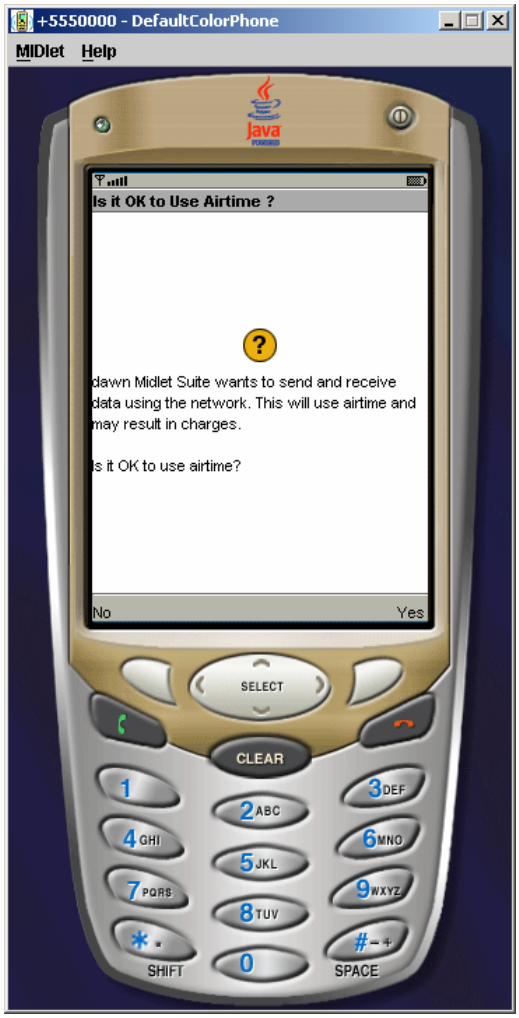



> Note that you must close the Export 184 Server window manually.

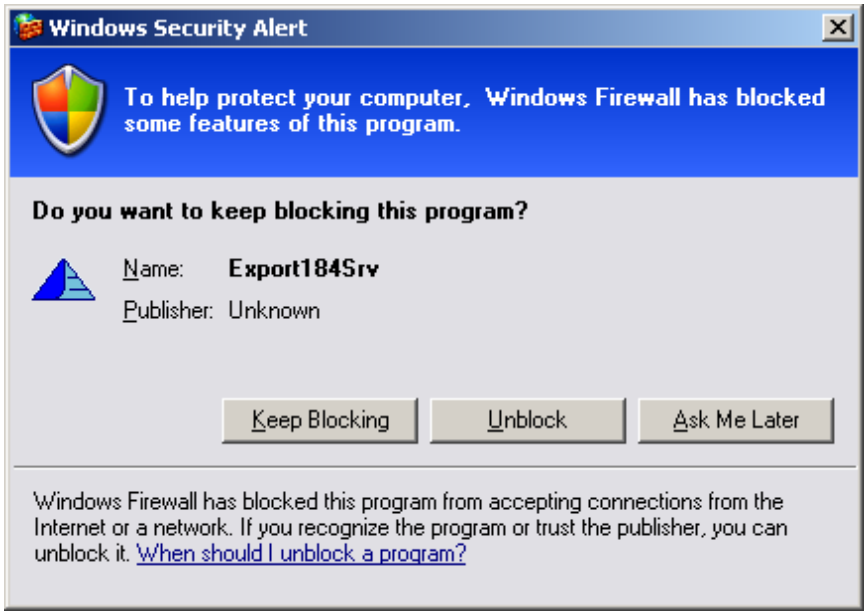
For security reasons, your mobile device cannot open certain folders and files, and the emulator, which hosts the M3G WTK Player, inherits this restriction.

The Export 184 Server grants access for the emulator to all files with the `.m3g` extension on your computer. When the server is starting up, Windows and the WTK emulator may warn that you are opening a potentially dangerous application.

- In the emulator window, choose Yes




- In the Windows Security Alert, click Unblock



# Playing M3G Scenes on Emulator

Once the M3G WTK Player is launched, you can preview `.m3g` scenes on the emulated phone screen. The player provides you access to all files with the `.m3g` extension on your computer.

> If the player is launched automatically (if the Launch WTK Player check box was selected), it starts playing the currently exported scene.

## Overview


Of the features provided by the toolkit's emulator, the M3G WTK Player applies the following.

**5-Way Navigation Pad -** Used to select options in on-screen lists and toolbars:

- Click arrows on the pad to navigate to the desired item
- When the item is highlighted, click Select

**Soft Keys -** Used to perform the functions currently labeled in the bottom line of the phone's screen.

**Input Keys** (**Numeric Keypad**) **-** Used here to duplicate buttons in the on-screen toolbars.

When running under the M3G WTK Player, the emulator may show screens of three types:

- *Open Dialog* - a list of folders and `.m3g` files on your computer; used to select a scene to preview
- *Worlds* screen - a list of the World-type objects in the currently loaded scene; used to set a rendering root
- *Preview* screen - in which you can playback animations and fly through the current scene

Normally the Preview screen is shown on the emulator. To switch screens, you should use the soft keys.

Also, for the Preview screen the player provides on-screen toolbars that control preview. To switch toolbars, you should use the up and down arrows of the 5-way Navigation pad.

## Procedures

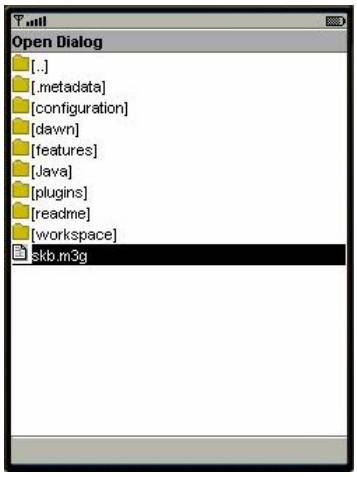
To play an .m3g scene, you should go through the following steps:

- Open the scene (optional step)
- Choose a rendering root (optional step)
- Play the animation, controlling the playback and camera position and rotation

### To open an `.M3G` scene:

You may need to switch to the *Open Dialog* first.

- If not the *Open Dialog* is on:
    - If the *Worlds* screen is on, click the right *Back* soft key to return to the Preview screen
    - When the Preview screen shows, click the left *Exit* soft key.
      The *Open Dialog* is switched on then.

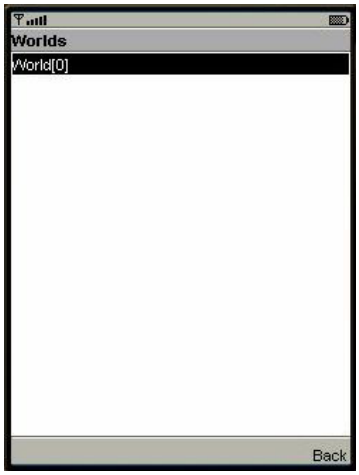



- In the *Open Dialog*, browse to the desired `.m3g` file, using the up and down arrows of the Navigation pad:
    - Selecting a folder opens this folder in the *Open Dialog*
    - Selecting an `.m3g` file will load the scene to play in the Preview screen.

### To change a rendering root:

You may need to switch to the *Worlds* screen first.

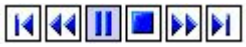
- If not the *Worlds* screen is on:
    - When the Preview screen shows, click the right *Menu* soft key.
      The *Worlds* screen opens then.




- In the *Worlds* screen, in the list of available *Worlds* objects, select a desired one, using the up and down arrows of the Navigation pad:
- To switch back to the Preview screen, click the right *Back* soft key.

**To control playback:**

- Switch to the *Animation Playback* toolbar on the Preview screen, using the up and down arrows of the Navigation pad



- Play the animation using the on-screen buttons described below in "Playback Controls".

**To control the camera position (Truck/Dolly Camera):**

- Switch to the *Camera Move* toolbar on the Preview screen, using the up and down arrows of the Navigation pad

- Move the camera using the on-screen buttons described below in "Camera Move Controls".

**To control the camera rotation (Pan/Roll Camera):**

- Switch to the *Camera Rotation* toolbar on the Preview screen, using the up or down arrows of the Navigation pad

- Rotate the camera using the on-screen buttons described below in "Camera Rotation Controls".

# Interface

## *On-Screen Toolbars*

To select a button in the on-screen toolbar, use the left and right arrows of the Navigation pad, and then click Select.

### Playback Controls

The player is controlled by the following buttons on the *Animation Playback* toolbar:

 - Rewind to the first frame.

 - Step one frame backward through the animation.

 - Start playing the animation.

 - Stop playing the animation and return to the first frame.

 - Pause the animation. Click  to resume.

 - Step one frame forward in the animation.

 - Forward to the last frame.

### Camera Move Controls

To move the camera (and its target) parallel to the view plane (*Truck* Camera), select the following buttons on the *Camera Move* toolbar:

 - Move upward and downward

 - Move to the left and to the right

To move the camera and its target along the camera's main axis, toward or away from what the camera is pointing at (*Dolly* Camera), select the following buttons on the *Camera Move* toolbar:

 - Move into or away from the view

### Camera Rotation Controls

To rotate the view about the camera (*Pan* Camera), use the following buttons on the *Camera Rotation* toolbar:

 - Produce a horizontal pan (about the world vertical axis)

 - Produce a vertical pan (about the world horizontal axis)

To rotate the view about the depth axis that is perpendicular to the screen (*Roll* Camera), use the following buttons on the *Camera Rotation* toolbar:

 - Rotate the camera about its line of sight

## *Keypad*

### 5-Way Navigation Pad

Used to select options in on-screen lists and toolbars:

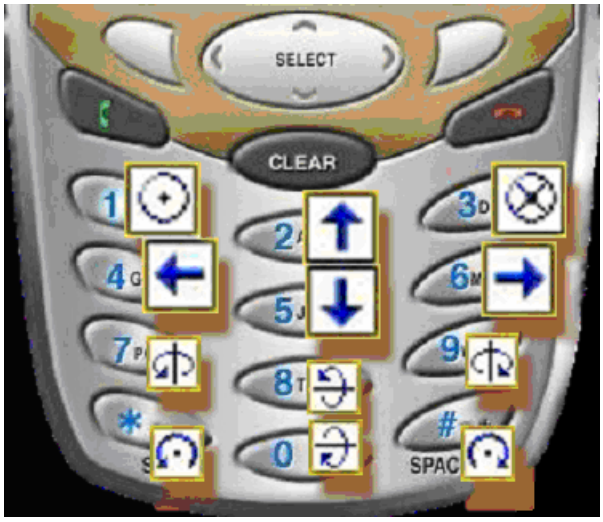
- Click arrows on the pad to navigate to the desired item
- When the item is highlighted, click Select

### Soft Keys

Used to perform the functions currently labeled in the bottom line of the phone's screen.

### Input Keys

Used here to duplicate buttons in the on-screen toolbars.



# Launching M3G WTK Player on Various Emulators

You can run the M3G WTK Player on various emulators.

To do so, you should start the M3G WTK Player manually, in much the same way as you did it for the primary Sun Java Wireless Toolkit:

- Using the Windows Start menu, start the Export 184 Server
- Start a toolkit of your choice (say, Sprint WTK) instead of the Sun Java Wireless Toolkit
- In the dialog box that opens, locate the `.jad` file for the M3G WTK Player, and then open it.

When prompted, allow access like you did it for the J2ME emulator (see "Launching M3G WTK Player on Emulator" above).

So you can work with the M3G WTK Player the same way as before.

# Programming JSR-184 Extensions

The Export 184 plug-in for 3ds Max offers a set of extensions for the JSR-184 file format. All these extensions require additional java code on the mobile device. Here you can see a more detailed explanation on how to implement sample Gravity and velocity properties for free fall simulation.

## Getting Started

Included with the Export 184 plug-in are the source codes of the M3G Player. The source codes illustrate both: (1) the typical steps to read and play JSR-184 scenes on a mobile device, and (2) how to use extensions provided by the Export 184 plug-in. You can find code samples for extensions in the `m3gext_src\m3gext\src\com\digitalelement\m3g\ext\` subdirectory of the directory where the M3G Player source code is installed.

JSR-184 extensions offered by the Export 184 plug-in (LOD Modifier for Mesh objects; Locator object; and Gravity and velocity properties) are supported in the M3G Player software by using a set of specifically designed Java classes. User parameters described in JSR-184 are used to prepare data for the extensions and to save the data in `.M3G` files.

Of the JSR-184 extension classes, the following are the most important:

- `public class GameObject extends GameObjectBase.`
  - a `GameObject` is created for each object being retrieved from an `.M3G` file, which is a descendant of JSR-184 Node class. `GameObject` stores extended object's data, such as its modifiers and so on
- `public class GameScene`
  - a `GameScene` is created for each World object retrieved from an `.M3G` file. It stores a list of `GameObjects` included in this World
- `public class GameObjectParams`
  - `GameObjectParams` provides access to User Parameters based on `Hashtable` objects
- `public abstract class ModifierBase`
  - all the JSR-184 extension modifiers are based on the `ModifierBase` class

The `GameObject, GameScene`, and `ModifierBase` classes allow updating JSR-184 scenes for each frame by propagating `update()` calls through the hierarchy.

Specific extensions shipped with the plug-in are implemented by using the following classes:

- `public class GameLocator extends GameObject`
  - `GameLocator` implements Locator objects.
- `public class ModifierGravity extends ModifierBase`
  - `ModifierGravity` implements Gravity and velocity properties for free fall simulation.
- `public class ModifierLOD extends ModifierBase`
  - `ModifierLOD` implements objects with Level of Detail (LOD).

For further details, see the M3G Player source code located within the `m3gext_src` folder shipped with the plug-in. The two generic sample files are shipped in the `m3gext_src\src\com\digitalelement\m3g\player\` sub-folder; these are: `ExampleGameObject.java` and `ExampleObjectModifier.java`.

# Adding Own Extensions

The JSR-184 extension classes mentioned above allow implementing your own modifiers.

## Example: Gravity and Velocity Properties for Free Fall Simulation

This example (Gravity and Velocity Properties) included with the plug-in helps to learn more about how to create user modifiers based on the JSR-184 extension classes.

The sample `ModifierGravity` modifier is intended to support a simple simulation of free fall motion.

The modifier shall be controlled by parameters of two types:

- Initial velocity of an object in 3D space (set for the objects we want to be affected by Gravity).
- Gravitational constant (is set for a World, and affects the above objects).

To implement a simple simulation by the `ModifierGravity` modifier we must do the following:

- Enter control parameters, store the data in an `.M3G` file.
- Define the `ModifierGravity` modifier object by extending the ModifierBase class.
- Code how to read `.M3G` data into the player program.
- Code the "simulating" functionality of the `ModifierGravity` modifier - to affect objects with time (so the modifier will make each object move with time, its 3D position depending on its initial velocity vector and Gravitational constant).

### How to enter control parameters, and store data in an `.M3G` file

We'll start with accepting a kind of "naming convention", which sets the same identification of parameters both in `.M3G` files and in the player program.

Those values should be added to the `GameObjectParams` source code used in the player program:

```
public static final int ID_HASH_GRAVITY = 100;
public static final int ID_HASH_VEL_X   = 100;
public static final int ID_HASH_VEL_Y   = 101;
public static final int ID_HASH_VEL_Z   = 102;
```

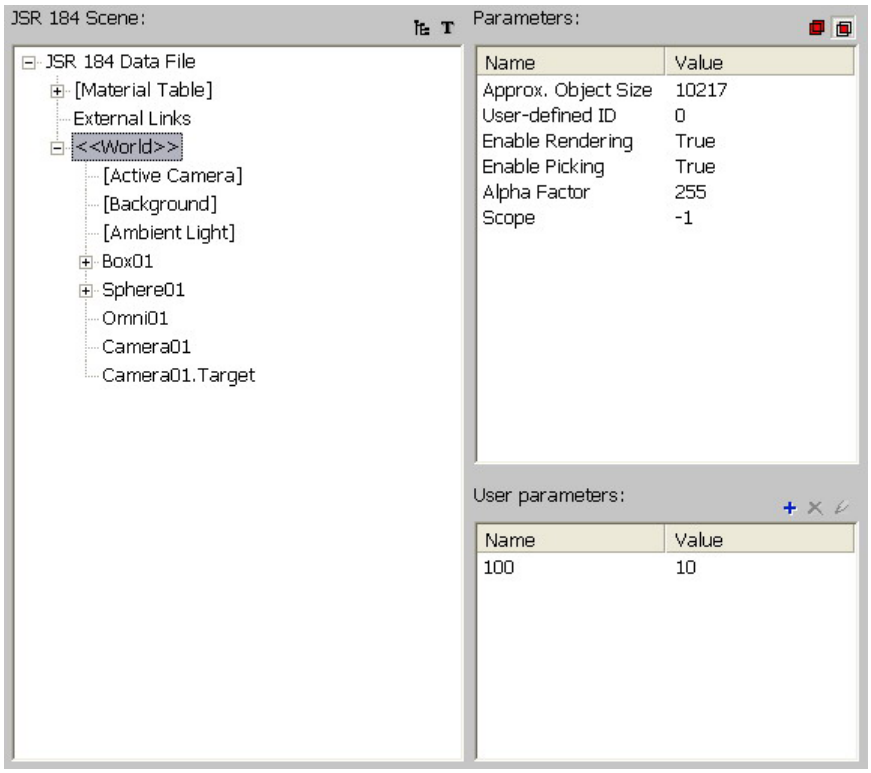
> Note that:
>
> - Here we may use one integer value (100) to identify two parameters, because these parameters belong to two different types of objects - `ID_HASH_GRAVITY` identifies a World's user parameter, and `ID_HASH_VEL_X` identifies a user parameter in another object.
> - We choose to represent a 3D vector of initial velocity by three user parameters - one parameter for each of three components.

You should use the same values when entering the User Parameters by the Export 184 plug-in interface.

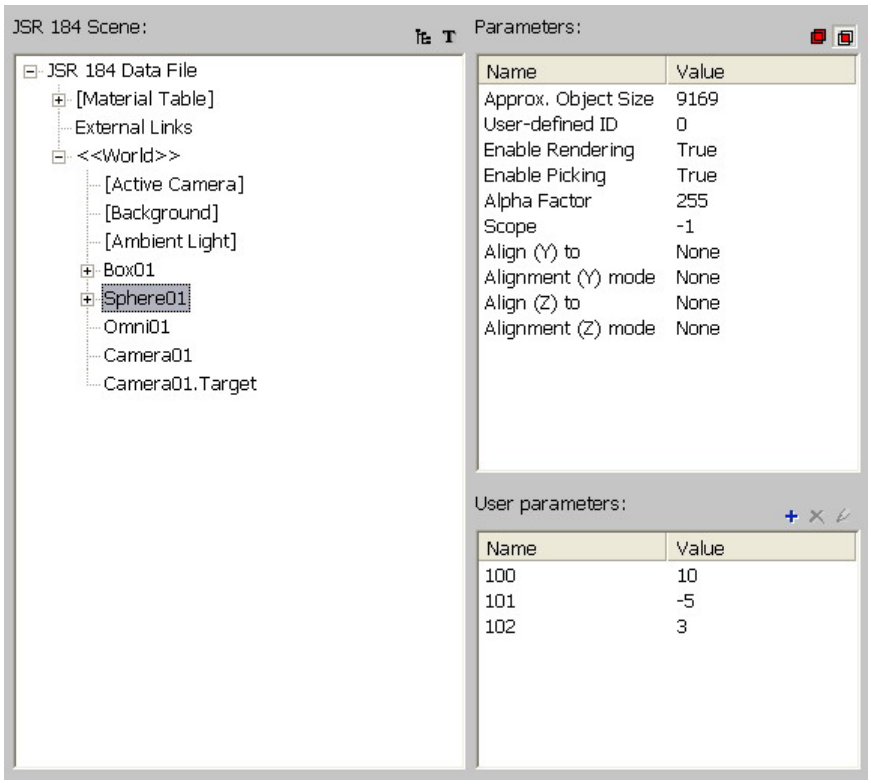
We can create an `.M3G` file, which includes data for the Gravity extension, as described in "Using JSR-184 Extensions". The below items remind you the procedure:

- In 3ds Max, create a scene including a box and a sphere. To make it visible, add also a light and a camera.
- In 3ds Max, go to the Export 184 plug-in. Create a JSR-184 scene by adding all objects from the 3ds Max scene to the JSR-184 scene.

- In the JSR-184 Scene tree, select the World object.
  For the World object, define a User parameter; set its Name to '100' (`ID_HASH_GRAVITY=100`; same as defined in `GameObjectParams`), and set its Value to 10 (10 is an integer value of the Gravitational constant):




- In the JSR-184 Scene tree, select the Sphere01 object.
  For the Sphere01 object, define three User parameters, by setting:
  - their Names to '100', '101', and '102'
    (these are `ID_HASH_VEL_X =100`, `ID_HASH_VEL_Y =101`, and `ID_HASH_VEL_Z =102`; as defined in `GameObjectParams`)
  - their Values to `10`, `-5`, and `3` respectively:



- In the JSR-184 Scene tree, go to the World object and set its Active Camera. Click the Export button to save the current JSR-184 scene to an `.M3G` file

### How to define your modifier object

- Use `ExampleObjectModifier.java` as an example of class description,
  create the `public class ModifierGravity extends ModifierBase` class description (see also `ModifierGravity.java`)

### How to read .M3G data into the player program

To extend program code for reading user parameters from .M3G file:

- In `GameScene` class, in the `loadGameObjects( Object3D node)` function look for reading objects of the World type.
  Add code to load data from the user parameter identified by `ID_HASH_GRAVITY`:

```
if (node instanceof World)
{
  addGravity = false;
  if (ht != null)
  {
    try
    {
      Integer idx       = new Integer(GameObjectParams.ID_HASH_GRAVITY);
      byte[]  byteArray = (byte[]) ht.get(idx);
      int     g         = ByteUtil.getInt(byteArray);
      addGravity = (g > 0);
    }
    catch(Exception e)
    {
    }
  }
  ...
```

  So, if for the current World object, from the user parameter identified by `ID_HASH_GRAVITY`, the Gravitational constant is successfully loaded, `addGravity` will be set to true.

- In the same place (in `GameScene` class, in the `loadGameObjects( Object3D node)` function) look for processing Mesh objects.
  If for the current World object its Gravitational constant was loaded (so `addGravity` is set to `true`), add code to create a new `ModifierGravity` modifier:

```
if (addGravity && (go == null) && (node instanceof Mesh))
{
  go = new GameObject(this, node);
  addObject(go);

  ModifierGravity mg = new ModifierGravity(go, new Integer(0));
  go.addModifier(mg);
}
```

### How to implement the "simulating" functionality

To make objects move with time simulating free fall motion:

- In `ModifierGravity`, overload the public void `update( int time )` function. Add code to this function to calculate how the object's current position and velocity change with time, depending also on the initial velocity vector and Gravitational constant loaded from file.

Finally, build the updated player program, and load the `M3G` file with stored user parameters data to test the result: our sphere moves falling in a parabola.

# Appendices

The following appendices describe:

- The JSR-184 objects and their parameters, which you can edit by the Export 184 plug-in.
- Export limitations
- Tips on creating models for JSR-184

# JSR-184 Object Parameters

For a JSR-184 scene in the Export 184 dialog, you can adjust many of the parameters of the scene's objects.

Objects and their parameters are described here.

> Note that the JSR-184 objects and parameters are mentioned here for your convenience only. For further details, see the respective JSR-184 documents, such as "Mobile 3D Graphics API Specification".

You can preview your exported scene using the M3G Player.

## Common Parameters

In a JSR-184 scene, most objects have two common parameters:

- *Approx. Object Size*. Shows the object size with sub-objects.

> This is a read-only parameter.

- *User-defined ID*. Every object can be assigned a user ID. User IDs are typically used to find different objects in a scene loaded from a data stream.

## Common Parameters of Node Objects

Node objects, such as Cameras, Lights, Sprite3Ds, Meshes, have more common parameters:

- *Enable Rendering*. Sets the rendering enable flag of this node. The effective rendering status for a node is the logical AND of the enable flags on this node and all of its ancestors. This means that this node is disabled if any of its ancestors are disabled. The status of this node has an effect only if all its ancestors are enabled. If the effective status is set to `True`, this node is enabled. If it is `False`, it is disabled.
- *Scope*. Sets an integer mask that allows you to group nodes independent of the scene graph hierarchy. It's just a different form of grouping things. The default scope is -1, implying that all nodes are in the same scope. By default, all objects are therefore visible to all cameras, and are lit by all light sources.
- *Align (Y) to.* Aligns the Y orientation of this node to the Y orientation of a target object, which you can select in the Parameters list (on the right side of the JSR-184 Export dialog box). This means that the aligned node is, upon request, automatically oriented so that its coordinate system matches the reference node's coordinate system in the specified way.
- *Alignment (Y) mode*. Sets the alignment (Y) mode. Options are None, Origin, X axis, Y axis, and Z axis.
- *Align (Z) to.* Aligns the Z orientation of this node to the Z orientation of a target object - by analogy with the *Align (Y) to* parameter, described above.
- *Alignment (Z) mode*. Sets alignment (Z) mode - by analogy with the *Alignment (Y) mode* parameter, described above.

For further details, see JSR-184 documentation.

## The JSR-184 Data File

The JSR-184 Data File is a root object in the JSR-184 Scene tree. Its only parameter is *Approx. File Size*, which shows an approximate size of the exported `m3g` file.

Among other items, the JSR-184 Data File includes two special sub-objects:

- Material Table
- External Links.

## External Links

The parameters of External Links are:

- The *size* of the External links section
- The *links* to textures saved in the external files, if any. These links are represented by filenames of the referred textures. Also other `m3g` files may be referred in the External links section.

Each user-added object in the External links section has two parameters:

- *File size*. Shows the size of the added file
- *File Root*. This parameter works much like the Project Root setting described above in "Export Settings Group", but sets location for this file only. If this parameter is set, then it overwrites the Project Root, otherwise the Project Root location is used.

# Material Table

The Material Table is a sub-object of the JSR-184 Data File. This table contains all materials used in the scene with their textures. In the JSR-184 Scene tree, a Material Table entry is named "[Material Table]".

The Material Table has a two-level structure:

- The first level corresponds to a JSR-184 Appearance object and is named by a 3ds Max material name
- The second level indicates sub-objects of the Appearance object. These items can be:
    - Texture
    - Material
    - Polygon Mode
    - Compositing Mode

## Appearance Objects

A JSR-184 Appearance object, named by the 3ds Max material name, has the following parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Layer*. Sets the rendering layer for a JSR-184 Appearance Object. When rendering World, Group or Mesh nodes, this parameter guarantees that sub-meshes and sprites are rendered in the order of ascending layers.

Also the Appearance object defines the rendering attributes, such as Texture, Material, Polygon Mode, and Compositing Mode.

### *Texture Sub-Objects*

A Texture sub-object represents a JSR-184 Texture2D object, and has the following parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Blending*. Specifies how to combine the filtered texture color with the incoming fragment color in a texturing unit. This is equivalent to the texture environment mode in OpenGL. Options are `Add, Blend, Decal, Modulate,` and `Replace.`
- *WrappingS* and *WrappingT*. The `Repeat` and `Clamp` texture wrapping modes define the treatment of coordinate values that are outside of the [0, 1] range.
- *Level Filter*. Sets the texture filtering mode. The following options are available: `Nearest`, `Linear`, and `Base Level`.
- *Image Filter*. Sets the image filtering. Options are: `Nearest`, `Linear`, and `Base Level`.

Image2D: A Texture object may have an Image2D sub-item with the following parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Width*. Specifies the width of the image, in pixels
- *Height*. Height of image, in pixels
- *Format*. Several image formats are available for Image2D objects: RGB, RGBA, Alpha, Luminance, and Luminance Alpha
- *Mutable*. The `Yes` value indicates the image is a Render Target; the `No` value indicates it is a plain image.

### *Material Sub-Objects*

A Material sub-object represents a JSR-184 Material object; it only has the following common parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.

### *PolygonMode Sub-Objects*

PolygonMode sub-objects correspond to the PolygonMode rendering attribute in JSR-184. Such objects can have the following parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Culling*. Determines which side of a polygon is removed from processing prior to rasterization: the back face (Back), the front face (Front), or neither (None).
- *Winding*. Specifies what is "front" and what is "back". *Winding* can be set to either clockwise (CW) or counter-clockwise (CCW). (If polygon vertices are defined in clockwise order, and the winding mode is set to be clockwise, the "front" side polygons are the ones which can be drawn to the screen in clockwise order. If the winding mode is set to counterclockwise, the polygon vertex order check is reversed.)
- *Shading*. Two options are available: Smooth and Flat. See the JSR-184 documentation for more details.
- *Two-sided Lighting*. Options are True and False. See the JSR-184 documentation for more details.
- *Local Camera Lighting*. Options are True and False. If *local camera lighting* is disabled, the direction from camera to a vertex being lit is approximated with the (0, 0, -1) vector. If *local camera lighting* is enabled, the direction is computed based on the true camera position. This method produces more accurate specular highlights.
- *Perspective correction* (options are True or False) defines whether textured polygons should use perspective correction algorithms to avoid texture stretching effect on large polygons, especially on those that are close to the camera's clipping plane. Perspective correction is a generic term for techniques that eliminate artifacts caused by the screen-space interpolation of texture coordinates, colors and fog. The lack of perspective correction is especially evident on large textured polygons: the texture is distorted and seems to "crawl" on the surface as the viewing angle changes.

> Note: The PolygonMode and Material objects can only be created if the original 3ds Max material is two-sided.

### *CompositingMode Sub-Objects*

A CompositingMode sub-object represents a JSR-184 Compositing Mode object, which defines different ways of combining polygons with the underlying pixels. A CompositingMode sub-object has the following parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Depth Test*. Enables or disables depth testing. Options are True and False.
- *Depth Write*. Enables or disables writing of fragment depth values into the depth buffer. Options are True and False.
- *Color Write*. Enables or disables writing of fragment color values into the color buffer. Options are True and False.
- *Alpha Write*. Enables or disables writing of fragment alpha values into the color buffer. Options are True and False.
- *Blending*. Selects a method of combining the pixel to be rendered with the pixel already in the frame buffer. Options are Alpha, Alpha Add, Modulate, Modulate X2 and Replace.
- *Alpha Threshold*. Sets the threshold value for alpha testing. Possible values: 0..255.

For further details, see JSR-184 documentation.

## World Objects

A World object is a special Group node that is a top-level container for scene graphs.

Among other sub-objects, each World object has 3 special sub-objects:

- Background
- Active Camera
- Ambient Light.

A World node is named "<<World>>" in the JSR-184 scene tree. World objects have the following parameters.

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Enable Rendering*. Enables rendering of the World sub-tree; options are `True` and `False`.
- *Enable Picking*. If `True`, the World sub-tree is enabled for picking; if `False`, it is disabled

- *Alpha factor*. The alpha factor allows (groups of) Mesh objects to be faded in and out in a convenient way, provided that certain preconditions related to their Appearance are met. The alpha factor is defined for each Node, and its value is between [0, 255].
- *Scope*. Sets an integer mask that allows scene graph nodes to form conceptual groups independent of the scene graph hierarchy. The default scope is -1, implying that all nodes are in the same scope. By default, all objects are therefore visible to all cameras, and are lit by all light sources.

For further details, see JSR-184 documentation.

### Active Camera Object

The only parameter of the Active Camera allows selecting an active camera from sub-objects of the current World object.

### Background Object

A Background object has the following parameters.

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Image Mode X* and *Image Mode Y*. Sets the repeat modes for the background image along the X-axis and Y-axis. Each image mode can be set to either `Border` (specifies that the imaginary pixels outside of the source image boundaries in X or Y direction are considered to have the background color), or `Repeat` (specifies that the imaginary pixels outside of the source image boundaries in X or Y direction are considered to have the same color as the pixel in the corresponding position in the source image).
- *Depth Clear Enabled*. Enables or disables the depth buffer clearing. If enabled, the portion of the depth buffer that corresponds to the viewport is cleared to the maximum depth value. Set this parameter to `True` to enable clearing, or set to `False` to disable clearing.
- *Color Clear Enabled*. Enables or disables the color buffer clearing. If enabled, the portion of the color buffer that corresponds to the viewport is cleared with the background image and/or the background color. Set this parameter to `True` to enable clearing, or set to `False` to disable clearing.

### Ambient light Object

An Ambient light object represents settings of the ambient light from the original 3ds Max environment. Parameters available for Ambient light objects are the same as parameters for Light objects described below.

## Group Objects

A Group object is a scene graph node that stores an unordered set of nodes as its children. In the JSR-184 scene tree, a Group object is named "<<Group>>".

Parameters available for Group objects are the same as parameters for World objects (see above).

> Note: In JSR-184, most objects cannot contain any sub-objects. As a result, to show the original 3ds Max hierarchy, Group nodes are used. A Group object is assigned a name like "<<ObjectName Group>>", where ObjectName is the name of a certain object that has sub-objects in the original 3ds Max scene.

## Camera Objects

A Camera object is a scene graph node that defines the position of the viewer in the scene; it transforms the coordinates from 3D space coordinates to 2D screen coordinates (camera space to clip space). A Camera object is assigned the same name as its counterpart in the original 3ds Max scene.

Camera objects have the following parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Scope*. Sets an integer mask to form node groups, as described above in "Common Parameters of Node Objects".
- *Align (Y) to, Alignment (Y) mode*, *Align (Z) to,* and *Alignment (Z) mode* - these parameters perform alignment, as described above in "Common Parameters of Node Objects".
- *Projection Type.* Sets the projection mode for the Camera. Options are `Parallel` and `Perspective`.

## Light Objects

Light objects are scene graph nodes that represent different kinds of light sources. A Light object is assigned the same name as its counterpart in the original 3ds Max scene.

Parameters of Light object are listed below.

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Enable Rendering*. Switches the Light on or off. Options are `True` and `False`.
- *Scope*. Sets an integer mask to form node groups, as described above in "Common Parameters of Node Objects".
- *Align (Y) to, Alignment (Y) mode*, *Align (Z) to,* and *Alignment (Z) mode* - these parameters perform alignment, as described above in "Common Parameters of Node Objects".

## Sprite3D Objects

Sprite3D is a scene graph node that represents a 2-dimensional image with a 3D position. The only way to create a Sprite3D object is to convert a two-polygon 3ds Max mesh object. A Sprite3D object is named "Sprite, ObjectName", where ObjectName is the name of the corresponding 3ds Max mesh object.

Sprite3D objects have the following parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Enable Rendering*. Sets the rendering enable flag of this node, as described above in "Common Parameters of Node Objects".
- *Scope*. Sets an integer mask to form node groups, as described above in "Common Parameters of Node Objects".
- *Align (Y) to, Alignment (Y) mode*, *Align (Z) to,* and *Alignment (Z) mode* - these parameters perform alignment, as described above in "Common Parameters of Node Objects".

## Mesh Objects

Mesh is a scene graph node that represents a 3D object defined as a polygonal surface. A Mesh object in JSR-184 Scene is assigned the same name as its counterpart in the original 3ds Max scene.

Parameters of a Mesh object are listed below.

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Enable Rendering*. Sets the rendering enable flag of this node, as described above in "Common Parameters of Node Objects".
- *Enable Picking.* Sets the picking enable flag of this node. The effective picking enable status for this node is the logical AND of the enable flags on this node and all its ancestors. This means that the node is disabled if any of its ancestors are disabled. Its own *Enable Picking* status has an effect only if all the ancestors are enabled. If the effective status is `True`, this node is enabled for picking; if is `False`, it is disabled
- *Alpha factor*. The alpha factor allows (groups of) Mesh objects to be faded in and out in a convenient way, provided that certain preconditions related to their Appearance are met. The alpha factor is defined for each Node, and its value is between [0, 255].
- *Scope*. Sets an integer mask to form node groups, as described above in "Common Parameters of Node Objects".
- *Align (Y) to, Alignment (Y) mode*, *Align (Z) to,* and *Alignment (Z) mode* - these parameters perform alignment, as described above in "Common Parameters of Node Objects".

For further details, see JSR-184 documentation.

Also, Mesh objects have the following sub-objects:

- Appearance
- Vertex Buffer
- Index Buffer

### Appearance Sub-Object

The Appearance sub-object is corresponding to the 3ds Max mesh material. See the description of Appearance objects in the Materials Table section.

### Vertex Buffer Sub-Object

A Vertex buffer has two parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.

A Vertex buffer may also include the following sub-objects:

- *Vertex Array [Positions]*. Array of vertex positions.
- *Vertex Array [Normals]*. Array of vertex normals.
- *Vertex Array [TexCoords]*. Array of vertex texture coordinates.
- *Vertex Array [Colors]*. Array of vertex colors.

Every vertex array has the common parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.
- *Component Size* (a read-only value). Displays the number of bytes per component of a vertex; must be 1 or 2. Not used for arrays of vertex texture coordinates.

### Index Buffer Sub-Object

The number of Index Buffers is always the same as the number of Appearance objects for the every Mesh object. Each Index Buffer has common parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.
- *User-defined ID*. Displays the user ID for the object.

### Level of Detail (LOD) Modifier

A Mesh object in the JSR-184 scene may also have the Level of detail (LOD) modifier. A LOD modifier has a set of meshes as sub-objects for every level. LOD modifiers have the following parameters:

- *Approx. Object Size* (a read-only value). Shows the object size with sub-objects.

Each LOD mesh sub-object has:

- *Mesh name.*
- *Threshold distance*.

Meshes are ranged by the *Threshold distance* value, so an appropriate mesh can be provided for each range of distances.

> For details, see "JSR-184 Extensions".

## Morphing Mesh Objects

A Morphing Mesh object is a scene graph node that represents a vertex morphing polygon mesh. A Morphing Mesh object in JSR-184 Scene is assigned the same name as its counterpart in the original 3ds Max scene.

Parameters available for Morphing Mesh objects are the same as parameters for Mesh objects (see above).

> Note: Morphing animation is not supported. Morphing Meshes are exported with morph targets. If you want to animate a Morphing Mesh, you should set animated weights to morph targets during playback.

## Skinned Mesh Objects

A Skinned Mesh object is a scene graph node that represents a skeletally animated polygon mesh. A Skinned Mesh object in JSR-184 Scene is assigned the same name as its counterpart in the original 3ds Max scene.

Parameters available for Skinned Mesh objects are the same as parameters for Mesh objects (see above).

A Skinned Mesh object in the JSR-184 Scene Tree has a special sub-object:

- [Skeleton]

This sub-object contains skeleton data for the Skinned Mesh.

# Export Limitations

Some 3ds Max entities can only be exported with limitations listed below:

| Category | Feature | Limitations |
|---|---|---|
| **Animations** | | |
| | Float controllers (Bezier & Linear) | |
| | Linear position controllers (Bezier & Linear) | |
| | Rotation controllers (Bezier & Linear) | |
| | Scale controllers (Bezier & Linear) | Only uniform scaling supported. |
| | Color controllers (Bezier, & Linear) | |
| **Cameras** | | |
| | Free camera | Depth of field and orthographic projection not supported |
| | Target camera | Depth of field and orthographic projection not supported |
| **Geometry** | | |
| | Any triangle-convertible geometry | |
| | Multiple UV's per vertex | Hardware limitations |
| | Multiple normals per vertex | |
| | Vertex colors | |
| | Vertex weights | |
| **Hierarchy** | | |
| | Full matrix hierarchy is converted | All scales except animated scales are pushed down the tree to the geometry. |
| **Lights** | | |
| | Omni light | Only Far attenuation supported. |
| | Spot Light | Only the falloff angle is supported (not hotspot) |
| | Directional Lights | |
| | Ambient Lights | |
| | Light intensity | |
| | Light on/off switch | |

| | | |
|---|---|---|
| | Animated light position, direction | |
| | Animated light color | |
| | Nonlinear Distance Attenuation | |
| **Materials** | | |
| | Multi/Sub Object-materials | |
| | Self-Illumination | |
| | Ambient material attributes with animation | |
| | Diffuse material attributes with animation | |
| | Specular material attributes with animation | |
| | Emissive material attributes with animation | |
| | Diffuse texture maps | Hardware limitations |
| | Opacity texture maps | Hardware limitations |
| | Texture clamp modes | |
| | Two-sided attribute | Disables backface culling for these objects, so they are only lit from one side |
| | Vertex color mapping | |
| | Opacity and key-framed opacity information | |
| | Animated UV offset, tile and rotation | |
| **Morphing** | | |
| | Morpher modifier | |
| **Skin** | | |
| | Skin modifier | Gizmos are not supported |
| **Environment** | | |
| | Fog | Only linear fog supported |
| | Environment Map | |
| | Background Color | |

# JSR-184 Modeling Tips

Designing models for JSR-184 is similar to designing in any other real-time computer game technology, and we can say briefly the models should be as simple and compact as possible.

Autodesk 3ds Max covers a very broad application domain, and it accordingly provides a rich set of tools. Some danger for designers of game applications is here - creating a complicated data structure can go unnoticed. In general, you can obtain the best optimization for your model, when its structure is as close as possible to some low level data structure in 3ds Max (for example, a mesh object or a vertex buffer structure).

The following tips will help you make your models look as good as possible:

- Use Reset XForm Utility for objects to normalize objects' transformation.
- Instead of using Reset XForm, you may want to set the Auto Scale option from the Export Settings group to normalize objects.
- Do not apply both skinned mesh and morphing mesh to the same mesh.
- Do not use texture coordinates out of 0 to 1 range.
- Use only the following component of the material - one texture map, which must be in the Diffuse channel.