

# CyberForm for Java3D

Release 1.1

## User's Guide

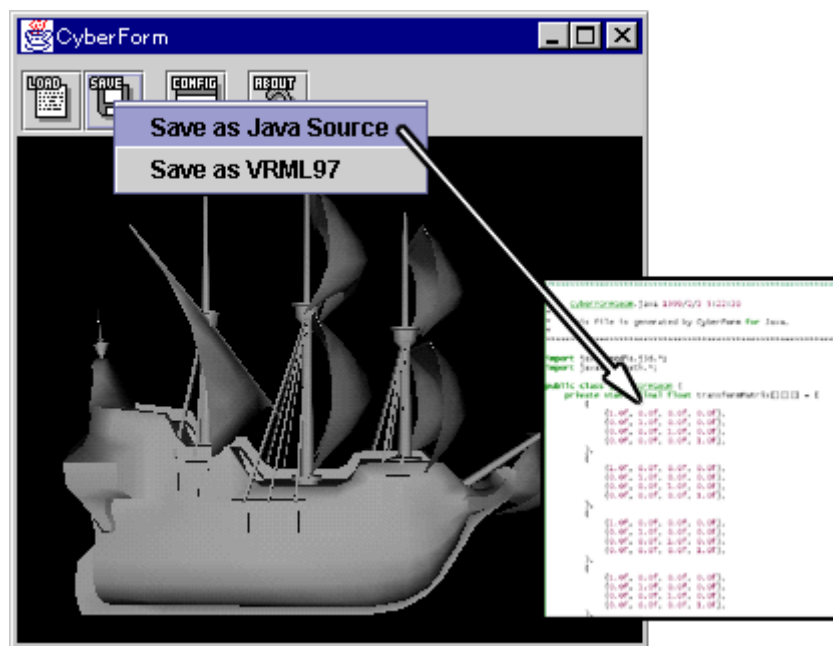
---

<b><i>What is CyberForm for Java3D ? .....</i></b>	<b><i>2</i></b>
<b><i>Installation.....</i></b>	<b><i>3</i></b>
<b><i>Using of CyberForm .....</i></b>	<b><i>4</i></b>
<i>Loadig a geometry file .....</i>	<i>5</i>
<i>Generating a source code for Java.....</i>	<i>6</i>
<i>Generation options .....</i>	<i>7</i>
<b><i>Using of a generated source code .....</i></b>	<b><i>9</i></b>
<b><i>Supported File Formats .....</i></b>	<b><i>11</i></b>
<i>VRML97 .....</i>	<i>11</i>
<i>Autodesk 3DS.....</i>	<i>11</i>
<i>Wavefront OBJ.....</i>	<i>11</i>
<i>LightWave3D LWS.....</i>	<i>12</i>
<i>SENSE8 NFF.....</i>	<i>12</i>

## ***What is CyberForm for Java3D ?***

---

CyberForm for Java3D is a source code generator for Java3D programmers. CyberForm generates a compressed binary file that has a specified geometry information with the source code. To specify the output precision, you can reduce the file size more. Using CyberForm, you can generate small and appropriate geometry files for Java3D applications on the Internet.



Currently, CyberForm can load some geometry file formats, VRML97, Autodesk 3DS, Wavefront OBJ, LightWave3D LWS and SENSE8 NFF, and generate the source code. Using CyberForm, you can add the generated source code to your programs very easily, and create the your programs more simply and small because you don't have to use big loader packages to load the geometry information.

You can use CyberForm as a translator utility for VRML97 because CyberForm can save a current geometry information into a VRML97 file too.

## ***Installation***

---

To run CyberForm for Java3D, you have to install latest Java2 (JDK1.2) and Java3D packages. You can get the packages from Sun's Java site (<http://java.sun.com>),

If you have installed a VRML-Java3D package of the Java3D and VRML Working group (<http://www.vrml.org/WorkingGroups/vrml-java3d/>), you should remove the package to install CyberToolbox easily because the following classes in the VRML-Java3D package conflicts with the VRML-CyberToolbox package.

<http://www.vrml.org/Specifications/VRML97/part1/java.html#B.9>

The CyberForm is distributed as a zip file. To extract the package, use a jar utility that is included with JDK or WinZip utility. For example, to extract the package using the jar utility,

```
j ar xvf cfj 3d110. zi p
```

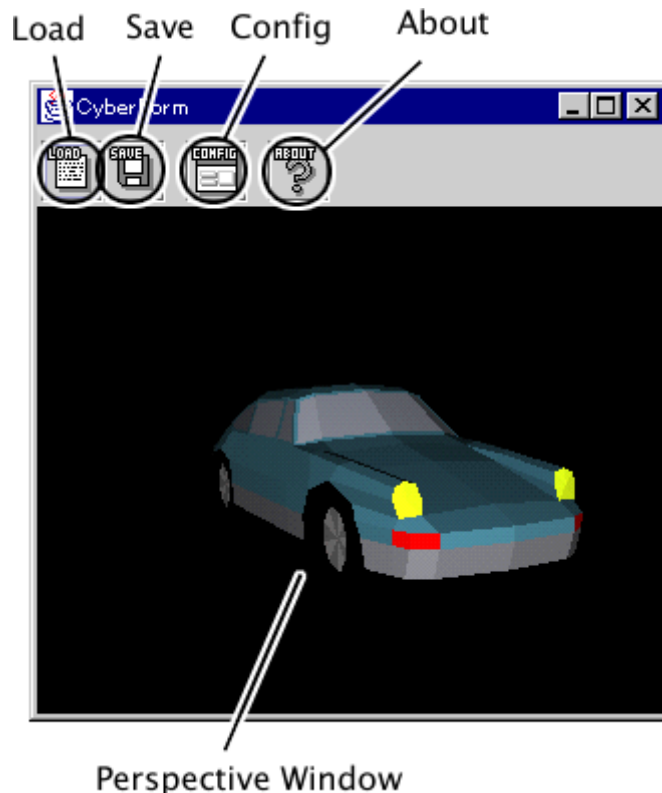
To run the CyberForm, use the following commands.

```
cd cyberform  
run (or run.bat)
```

## Using of CyberForm

---

CyberForm has a perspective window and four operation buttons, Load, Save, Config and About in the tool bar.



**Load**

Click to load a geometry file. Please see the following “Loading a geometry file” in more detail.



**Save**

Click to save a current geometry information into a source code for Java or a VRML97 file. Please see the following “Generating a source code for Java” in more detail.



**Config**

Click to set generation options for creating source codes for Java. Please see the following “Generating a source code for Java” in more detail about the generation options.



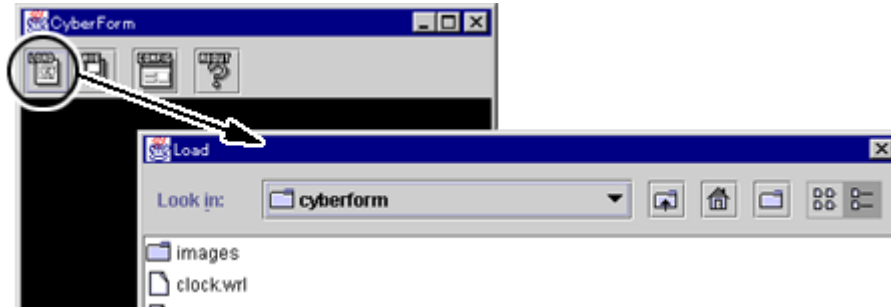
**About**

Click to confirm a current release information.

## **Loadig a geometry file**

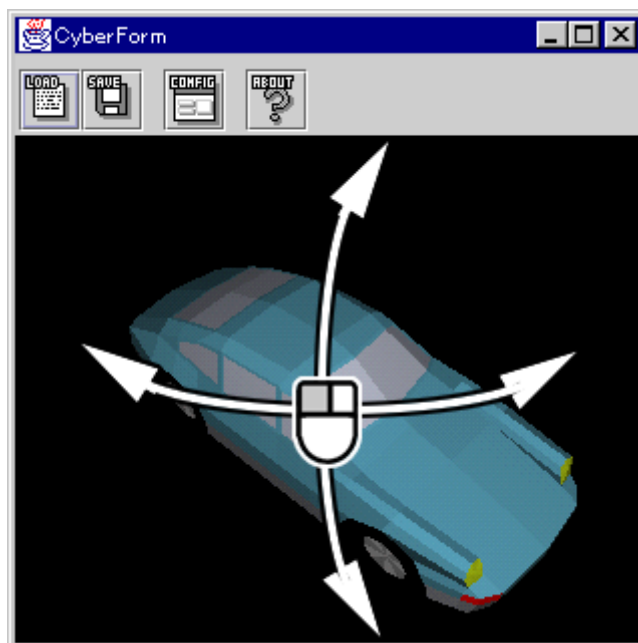
CyberForm can load a geometry file to generate the source code for Java or save into a VRML97 file.

To load a geometry file, click Load button and select the geometry file.



Currently, CyberForm supports for loading VRML97, Autodesk 3DS, Wavefront OBJ, LightWave3D LWS and SENSE8 NFF file format. Please see a description of “Supported File Format” about the file formats in more detail.

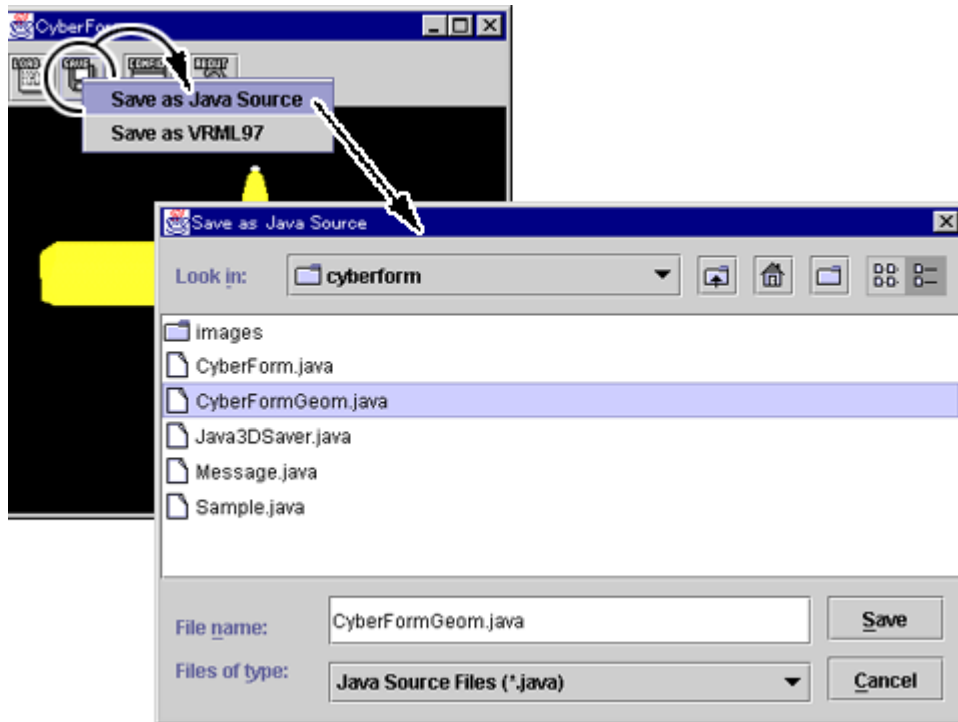
When CyberForm can load the specified geometry file normally, the geometry is centered in the perspective window automatically so that you can see all the geometry. To rotate the geometry, drag the geometry with a mouse. The rotation angle is initialized when you release the mouse button.



If you want to load a new geometry file, click Load button again. When CyberForm can load the new geometry file normally, a current geometry that is displayed in the perspective window is removed, and only the new geometry is displayed.

## **Generating a source code for Java**

CyberForm can generate a source file for Java from a geometry information that is displayed in the perspective window currently. To generate the source code, click Save button, then select “Save as Java Source “ in the pop-up menu and input a file name that you want.



CyberForm saves the geometry information into the specified source file with a compressed binary file. The binary file has compressed information of vertices, polygons and textures of the geometry.

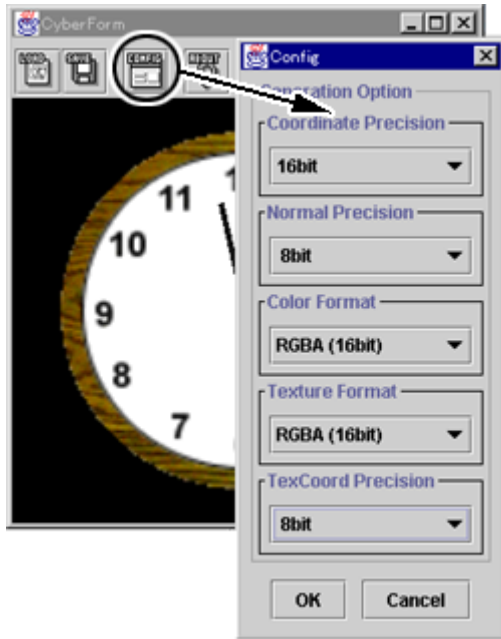
The generated source code reads the compressed binary using CyberFormGeometryLoader class when you create the geometry. If the source file of CyberFormGeometryLoader class doesn't exist in your specified directory, the source file is generated too.

For example, the following three files are generated into you specified directory when you set “CyberFormGeom.java” into the file name. The “CyberFormGeom.bin” is the compressed binary file.

CyberFormGeom. j ava  
CyberFormGeom. bi n  
CyberFormGeometryLoader. j ava

## Generation options

Using the generation options, you can control the precision and the size of the compressed binary file. To set the options, click Config button to display a following dialog..



### Coordinate Precision

Use this option to select a coordinate precision from 32 bit, 24 bit, or 16 bit. In general, you should select the 32bit option if your geometry has many vertices and the extent is large.

### Normal Precision

Use this option to select a coordinate precision from 32 bit, 24 bit, 16 bit, or 8 bit. In general, you can assure enough precision using the 8bit option.

### Color Format

Use this option to select a color format from the following types. For example, select “RGBA (16bit)” if you want to create the smaller binary file at the price of the color depth.

Type	Binary Format
RGB (24bit)	RRRRRRRRRGGGGGGGGBBBBBBBB
RGBA (32bit)	RRRRRRRRRGGGGGGGGBBBBBBBBAAAAAAAA
RGBA (16bit)	ARRRRRGGGGGGBBBBBB



## Texture Format

Use the option to select a texture format from the following types. If you don't want to save any texture information, select "No Texture".

Type	Binary Format
No Texture	-
RGB (24bit)	RRRRRRRRRGGGGGGGGBBBBBBBB
RGBA (32bit)	RRRRRRRRRGGGGGGGGBBBBBBBBAAAAAAAA
RGBA (16bit)	ARRRRRGGGGGGBBBB

## TexCoord Precision

Use this option to select a texture coordinate precision from 32 bit, 24 bit, 16 bit, or 8 bit. In general, you can assure enough precision using the 8bit or 16bit option.

## Using of a generated source code

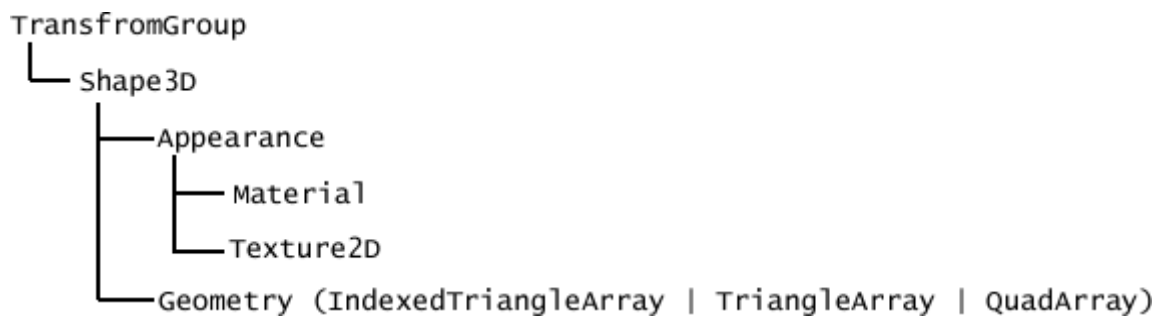
---

CyberForm generates a source code for Java2 and Java3D from your specified geometry file. This chapter describes how to use the generated source code into your program.

Regardless of the generation option, the generated source code has the two constructors for standalone or applet programs and the following public methods to get the shape information.

Method	Description
int getShapeCount()	Returns a number of shapes in the class
TransformGroup getShape(int n)	Returns a specified shape
TransformGroup createShape(int n)	Create a copy of a specified shape and returns the copy

The getShape() and createShape() returns a TransformGroup node that has the shape information, and it consists of the following Java3D nodes. To get the shape information normally, you have to copy the compressed binary file into the same directory or URL as the source code files.



The TransformGroup node has a connected transform information of the shape, the Appearance has the material and texture information, the Geometry node has the coordinate, the normal, the color and the texture coordinate information.

The following example shows how to add all shapes that are generated by CyberForm into a BranchGroup node in your program. In this example, CyberFormGeom is a class name of a generated source code by CyberForm.

```
BranchGroup root = new BranchGroup();
CyberFormGeom geom = new CyberFormGeom ();
for (int n=0; n<geom.getShapeCount(); n++)
    root.addChild(geom.getShape(n));
```

To get the shape information from your applet programs, you have to set the base URL into the constructor to access to the compressed binary file normally. For example,

```
public class Sample extends Applet {  
    .....  
    public BranchGroup createSceneGraph() {  
        BranchGroup root = new BranchGroup();  
        CyberFormGeom geom = new CyberFormGeom (getCodeBase());  
        for (int n=0; n<geom.getShapeCount(); n++)  
            root.addChild(geom.getShape(n));  
        return root;  
    }  
    .....  
}
```

## **Supported File Formats**

---

CyberForm supports for loading the following geometry file formats, and save the geometry information into a source file for Java or a VRML97 file.

### **VRML97**

CyberForm can get all information of VRML97 nodes in the specified file. However CyberForm saves only geometry information of Box, Cone, Cylinder, Sphere and IndexedFaceSet nodes when you want to generate the source file for Java. The geometry information is saved with the appearance and the transform information.

### **Autodesk 3DS**

CyberForm gets only the following information from the specified file, and ignore the other information.

Chunk ID	Description
0xA010	Material Ambient Color
0xA020	Material Diffuse Color
0xA030	Material Specular Color
0xA040	Material Shininess
0x4100	Triangle Set
0x4110	Triangle Point Set
0x4120	Triangle Face Set

### **Wavefront OBJ**

CyberForm gets only the following information from the specified file, and ignore the other information. CyberForm doesn't read the map files and the material files

ID	Description
v	Vertex Position
vn	Vertex Normal
f	Face Index

### **LightWave3D LWS**

CyberForm uses a utility class of Java3D package, `com.sun.j3d.loaders.lw3d` , for loading the files. Please see the document of the package about the loader in more detail. CyberForm converts from the Java3D nodes into the VRML97 nodes, then add the VRML97 nodes into a current scene graph.

### **SENSE8 NFF**

CyberForm gets only the vertex positions and the polygon indices with the color from the specified file, and ignore the other information.