



# modo 302

modo 302 was released in April, 2008 as a free, downloadable upgrade for registered modo 301 users.

The update is available for both Mac and PC.

## modo File I/O Software Development Kit

Document revised on: March 31, 2008

Version: 302/24870

Date published: April 3, 2008

Compatible with this build of modo: 302

Availability: included with permanent licenses of modo 302

### What is the File I/O SDK:

The modo File I/O SDK provides an application programming interface (API) and source code samples that enables developers to read and write modo 302 scene and image data into files for use by other applications. The SDK only reads or writes to named files; it doesn't explicitly support a 'live' data connection. It is COM-based and designed for the creation of plug-ins for both the Mac and PC versions of modo 302.

### What is included:

Header files and source to build libraries

We provide sample projects for MSVC++ 6 and up, and Xcode version 3.

Note: Samples are entirely C++. The headers are a mix of C and C++.

Samples are provided in source code form (in C) with liberal licensing that lets developers use this code in their own applications/products.

## modo File I/O Software Development Kit, continued

### Samples provided include:

- > Collada (saver)
- > Autodesk FBX (loader and saver)
- > HPGL (Saver only. This is an example of how to create a preference UI using forms. The form sets user values which the plug-in can read. The config can be added automatically to the user's settings, but they will need to integrate the form into their UI.)
- > GEO (loader and saver)
- > OBJ (2 provided; one is the version that ships with modo 302 and contains some unsupported interfaces. The new one is rewritten with only supported interfaces.)
- > freeimage (loader, multiple savers)
- > iraw (loader, saver)

### What you can do with the File I/O SDK:

You can create loaders or savers for geometry or image data.

Loaders consist of a recognition method which can scan the file to see if it's of a known type, and a load method which allows the loader to import data from the file into an object (scene or image) provided by modo.

Savers have a verify method which can display information or warnings about the chosen file format, and a save method. Saving reads the state of a scene or image object and writes that to the file. With calls to the SDK you can access any parameter known to modo about the base mesh. Surface normals are supported, both per-vertex and per-polygon. Developers have access to topology information (e.g. what edges are connected to given edge). If a new image saver is defined, then that format will appear in the "save as" choice for saving renders. You can also define your own layered image saver. The renderer can generate a depth buffer which can be saved as an image or as part of a layered image

You can import vertices and polygons. Also, polygons includes faces, subdivision surfaces, curves of various types, patches and text polygons.

You can use modo forms to build your user interface.

Plug-ins can have preference settings which can be read to alter their behavior.

## modo File I/O Software Development Kit, continued

### Applications and Benefits:

Direct access to modo data is provided to developers who need a customizable way to interface modo to other 3D and 2D applications.

- › Export data from modo into the rest of your pipeline or tool suite. You control how modo data looks to your application for seamless downstream use in your pipeline.
- › Import data from another application into modo. Read a proprietary in-house format or tackle and industry standard format.

### Limitations

Errors encountered during loading or saving can be sent to a log, however loaders and savers cannot open dialogs, fire commands or otherwise interact with the user.

Plug-ins cannot read out GL tessellation or displacement — to save those you'd have to freeze the mesh first.

Currently it is possible only to determine the selection state of vertex maps (only), as this is used by formats that support only one UV map to find the one the user wants to save.

You can “pipe” data to another application if your saver writes to a pipe, but this wouldn't be a live connection.