Rhinoceros for Mac

Although there is not yet plugin support for Rhino for Mac, it is still possible to work with Maxwell and Rhino natively on OSX, through the use of Maxwell Studio, since Maxwell is a fully cross-platform application suite. This workflow is enabled through Maxwell Studio's support for the import of various file formats: it is possible to export various file formats from Rhino, import them into Studio, set up materials, cameras, and such, and then render in Maxwell Render.

Exporting and Importing files

Rhino and Maxwell Studio share support for the following export/import file formats:

Supported geometry import formats	
AutoCAD Drawing Exchange	*.dxf
COLLADA	*.dae
LightWave	*.lwo
MotionBuilder	*.fbx
WaveFront OBJ	*.obj
PLY (Polygon File Format)	*.ply
STL (Stereolithograpy)	*.stl

Please note that different formats have different limitations and export options, so one may work better than another for your particular workflow. There may be better or worse compatibility depending on the file version exported (in Rhino's Export Options for the format – see here for details on Rhino's Export options), and whether the file is exported using text or binary format, both of which are supported for some formats. Once a preferred export/import format, and export options have been found, it is quite easy to encapsulate the action in a script:

Example OBJ Export Command Script

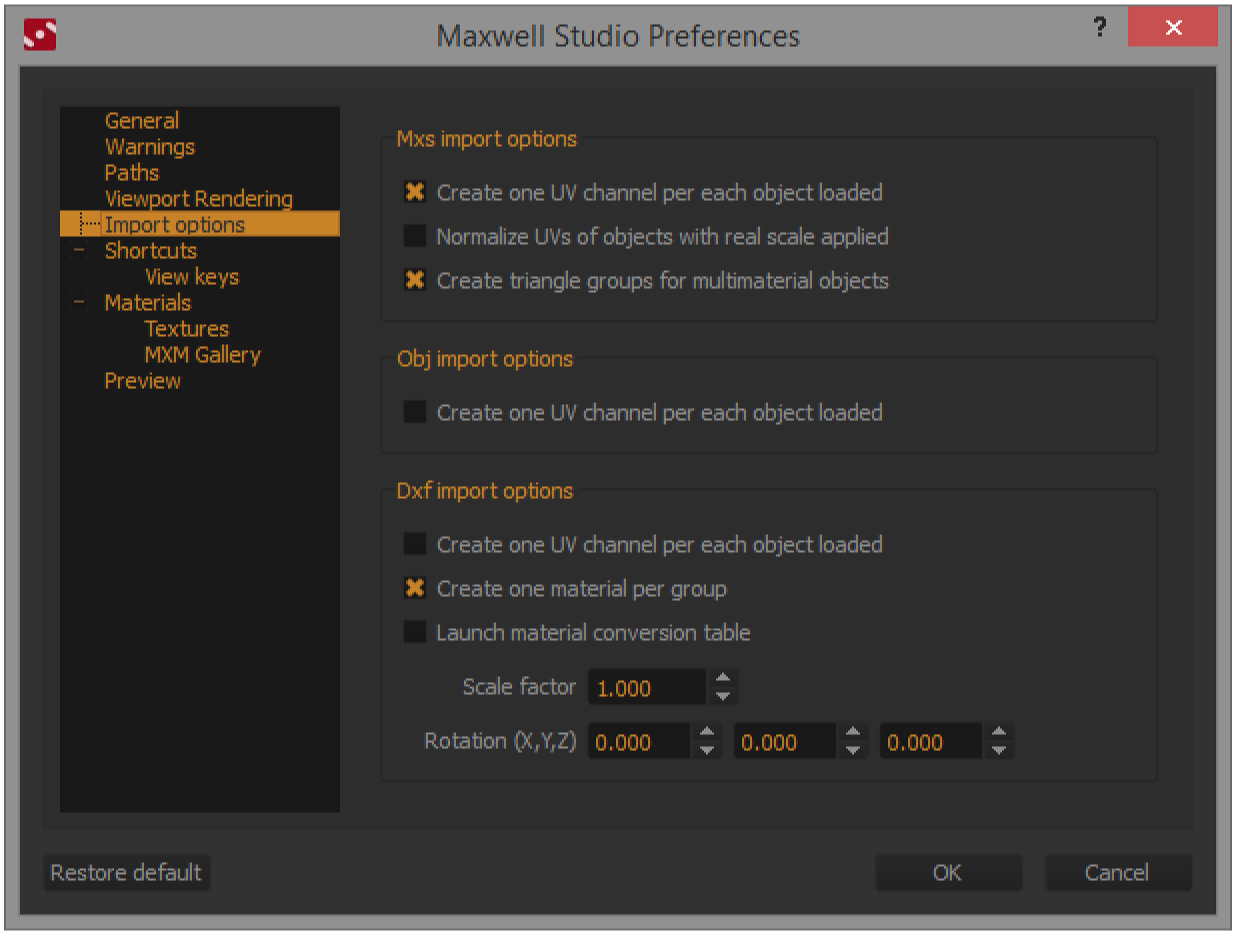
Expand source

```
_NoEcho
_-RunPythonScript (
import rhinoscriptsyntax as rs
arr = rs.SelectedObjects()
if not len(arr):
  # No objects were pre-selected.
  arr = rs.GetObjects('Select objects to export')
if len(arr):
  # Note: .obj will be added to the name.
  name = rs.GetString('Enter a name for the OBJ file')
  if len(name):
    try:
      # Make sure the objects are selected.
      rs.SelectObjects(arr)
      # Disable redraw.
      rs.Command('_noecho _setredrawoff ')
      # Export the OBJ.
      rs.Command('_noecho _-export ' + name + '.obj' +
        ' g m'  + # Geometry=Mesh
        ' e c'  + # EndOfLine=CRLF
        ' x e'  + # ExportRhinoObjectNames=ExportObjectsAsOBJGroups
        ' p=y'  + # ExportMeshTextureCoordinates=Yes
        ' o=y'  + # ExportMeshVertexNormals=Yes
        ' c=n'  + # CreateNGons=No
        ' r=y'  + # ExportMaterialDefinitions=Yes
        ' y=n'  + # YUp=No
        ' w=n'  + # WrapLongLines=No
        ' v u'  + # VertexWelding=Unmodified
        ' i 16' + # WritePrecision=16
        ' _enter _enter')
    finally:
      # Re-enable redraw.
      rs.Command('_noecho _setredrawon ')
)
```

This is a python script, wrapped in a *-RunPythonScript* command block, such that it can be placed into the Command of a toolbar button and run with a single click. This particular script prompts for an output filename (with the file being exported into the same directory as the 3DM), but this could easily be changed such that it would always write to the same name, and so forth, depending on the workflow you prefer to establish.

Import Options

Also note that Maxwell Studio has various options (under *Import Options* in Maxwell Studio Preferences) specifically related to some of these formats:



Mxs import options

- [✓] Create one UV channel per each object loaded
- [ ] Normalize UVs of objects with real scale applied
- [✓] Create triangle groups for multimaterial objects

Obj import options

- [ ] Create one UV channel per each object loaded

Dxf import options

- [ ] Create one UV channel per each object loaded
- [✓] Create one material per group
- [ ] Launch material conversion table

Scale factor 1.000

Rotation (X,Y,Z) 0.000 0.000 0.000

Import Options (OBJ & DXF)	
Create one UV channel per each object loaded	If an imported mesh has no UV coordinates, a set will be automatically generated for that mesh.
Import Options (DXF)	
Create one material per group	Create one Maxwell material for each group found in the DXF, and associate it with corresponding geometry.
Launch material conversion table	Show the DXF Material Conversion table, which allows associating materials and MXM files with imported geometry.
Scale Factor	A global scale factor to be applied to imported DXF geometry.
Rotation	A global rotation to be applied to imported DXF geometry.

Working with Imported Geometry

Once your geometry has been imported into Maxwell Studio, you can proceed to use Maxwell FIRE to interactively set up cameras, materials, env ironment, advanced texture mapping of objects, and so forth, and also make use of some of Maxwell's more advanced features such as Maxwell Grass, Maxwell Scatter, etc.

In the future

Watch this space in the near future, for information on a new feature called *AssetReference*, originally conceived with the specific purpose of improving the workflow between Rhino for Mac and Maxwell Studio, which will debut in the early builds for the upcoming Maxwell 3.2. This is a Geometry Loader Extension, which will be able to be used to establish an active link to non-Maxwell files, where whenever the referenced file is changed, its geometry will be automatically updated in real time in Maxwell Studio (in the OpenGL viewport and in Maxwell FIRE). It will support mapping of different axis orientations to Maxwell (since different formats consider different directions to be "up"), and it will also support instancing.

In addition to its realtime geometry-update capability, the AssetReference extension will also automatically map material names found in its referenced file either to materials currently existing in the scene, or to MXM files found in the referenced file's directory. In the event that neither can be found, it will also attempt a rudimentary translation of the material in the file, though this can be quite limited; however, it will also support an option whereby translated materials will automatically be written to disk as MXM files (since it can be difficult to know what they may end up being named in the exported file).

This extension is basically a wrapper around the Open Asset Import Library, and as such, will support a wide variety of 3D file formats (list here, though experience shows that even more formats are supported, than claimed there).