FBX to SF Converter v. 1.0.0

Table of contents:

1. What is this?
2. How do I use this?
3. Why would I use this?
4. Detailed instructions
5. Additional remarks
6. Change log

1. What is this?

FBX to SF Converter is a tool with which you can convert 3D models and animations from SpellForce 1 to FBX format and vice-versa, from FBX to 3D models and animations that SpellForce 1 can understand.

2. How do I use this?

There are two ways to do it:

a) Simply **drag and drop** the files onto `SF1_tool_fbx_io.exe`.

b) Use **command line**:

```
SF1_tool_fbx_io.exe file1 file2 ...
```

3. Why would I use this?

It was made so people can use a variety of existing 3D modelling, rigging and animating software to add new content for this game. FBX format is widely supported by nearly all mentioned software. The tool is mainly intended for use by modders, but it's simple enough that anyone can do the conversion of anything.

4. Detailed instructions

Internally, there are 6 main commands that the converter can perform:

a) Export mesh to FBX
b) Export rigged mesh to FBX
c) Export animation to FBX
d) Import mesh from FBX
e) Import rigged mesh from FBX
f) Import animation from FBX

First, the converter sorts all provided files to different bins, depending on the file's extension.
There are bins for the following extensions:
- **.msb** (mesh file)
- **.bor** (bone reference file)
- **.bob** (animation file)
- **.fbx** (FBX file)

Additional files may be created by the converter, but are not used by it:
- **.msb2** (skin file)
- **.bsi** (bone index file)

Then, the converter extracts files from the bins and makes a list of commands to perform. These are the requirements for commands to be added:

I. For **every file** in **.msb** bin:
- If there is **a file with the same name** in **.bor** bin, add command **b)**
- **Otherwise**, add command **a)**

II. If there is **only one file** in **.bor** bin, then for **every file** in **.bob** bin, add command **c)**

III. For **every file** in **.fbx** bin, the converter will **automatically determine** whether to add command **d)**, **e)** or **f)**

What it means in simpler words:

I. If you drag 2 **.msb** files onto the `SF1_tool_fbx_io.exe`, 2 **.fbx** files will be created, each containing a separate mesh.
If you drag 2 **.msb** files and 2 corresponding **.bor** files onto the `SF1_tool_fbx_io.exe`, 2 **.fbx** files will be created, each containing a separate rigged mesh.

II. If you drag 1 **.bor** file, and N number of **.bob** files onto the `SF1_tool_fbx_io.exe`, N **.fbx** files will be created, each containing a skeleton with a separate animation.

III. If you drag N number of **.fbx** files onto the `SF1_tool_fbx_io.exe`, a number of files will be created, depending on the content of the dragged files (see command details below).

For commands **a)**, **b)**, **d)** and **e)**, **textures** used by mesh have to be present in the same folder as the input **.msb** [commands **a)**, **b)**] or the input **.fbx** [commands **d)**, **e)**].

Used by commands **a)**, **b)**, **d)** and **e)**, you can specify additional command line arguments:
`--invert-uv` inverts UV coordinates of all vertices of the mesh
`--invert-normal` inverts normal vectors of all vertices of the mesh

For **all** commands, the converted files will be **created in the same folder** as the **.msb** file [commands **a)**, **b)**], **.bor** file [command **c)**] or **.fbx** file [commands **d)**, **e)**, **f)**]. Additionally, `log.txt` will be **created in the same folder** if you're using drag and drop functionality, otherwise it will be **created in the folder with the converter .exe**. If the conversion fails, the reason will be provided in that file.

Details of every command:

a) Export mesh to FBX
Required files: a **.msb** file
Generated files: an **.fbx** file containing the converted mesh
Additional info: **Textures** referenced by the **.msb** file must be **in the same folder** as the **.msb** file that references them.

b) Export rigged mesh to FBX
Required files: a **.msb** file, a **.bor** file
Generated files: an **.fbx** file containing the converted mesh linked with the converted skeleton, effectively creating a rig
Additional info: **Textures** referenced by the **.msb** file must be **in the same folder** as the **.msb** file that references them. The **.bor** file must exactly **correspond** to the **.msb** file.

c) Export animation to FBX
Required files: a **.bor** file, a **.bob** file
Generated files: an **.fbx** file containing the converted skeleton, animated using the converted animation
Additional info: The **.bob** file must **match** the **.bor** file in the **number of skeleton bones** used by those files.

d) Import mesh from FBX
Required files: an **.fbx** file containing a mesh
Generated files: a **.msb** file containing the converted mesh data
Additional info: **Textures** referenced by the **.fbx** file must be **in the same folder** as the **.fbx** file that references them.

e) Import rigged mesh from FBX
Required files: an **.fbx** file containing a rigged mesh (mesh with linked skeleton)
Generated files: a **.msb** file containing the converted mesh data, a **.bor** file containing the converted skeleton data, a **.msb2** file containing the generated skin data, and a **.bsi** file containing the bone indices used by the **.msb2** file
Additional info: **Textures** referenced by the **.fbx** file must be **in the same folder** as the **.fbx** file that references them.

f) Import animation from FBX
Required files: an **.fbx** file containing a skeleton with animation
Generated files: a **.bob** file containing the converted animation data
Additional info: None.

With regards to SpellForce 1, the generated files should be put in the following directories:
- **.msb** files belong to `mesh/` folder
- **.bor** files and **.bob** files belong to `animation/` folder
- **.msb2** files and **.bsi** files belong to `skinning/b20/` folder
Additionally, **.msb2** files must have their extension **changed** to **.msb** in order to work correctly.

5. Additional remarks

For Blender users:

The .fbx files can be imported and exported using the FBX plugin included with Blender, starting with version **2.8**.

The recommended **import** settings:
- **Include** -> **Custom Normals** **ON**, **Custom Properties** **ON**
- **Transform** -> Scale: **100.00**
- **Animation** -> **Animation Offset**: **0.00**
- **Armature** -> **Ignore Leaf Bones** **OFF**, **Automatic Bone Orientation** **OFF**, **Primary Bone Axis**: **X Axis**, **Secondary Bone Axis**: **Y Axis**

The imported meshes will probably be "laying on the floor", you need to go to **Object Properties** and set Transform **->** Rotation X/Y/Z to **0** degrees

Meshes with **transparent** materials will not appear correct right after loading the FBX to Blender. Go to **Material Properties** of each **transparent** material, and set **Settings** -> **Blend Mode** to **Alpha Hashed**, and **Settings** -> **Shadow Mode** to **Alpha Hashed**. **Transparent** materials are those which have a texture connected to the material's Alpha property in **Surface** tab.

Each material specifies the following **custom properties**:
- **DepthBias** (**0** is default)
- **RenderMode** (**0** is default)
- **TextureFlags** (**7** is default for **opaque** materials, **3** is default for **transparent** materials)
- **UserData** (**0** is default)
- **WrapMode** (**0** is default)

The recommended **export** settings:
- **Include** -> **Limit to Selected Objects** **ON** (will have to only have mesh/skeleton selected), **Custom Properties** **ON**
- **Transform** -> Scale**: 100.00**
- **Geometry** -> **Smoothing**: **Normals Only**
- **Armature** -> **Primary Bone Axis**: **X Axis**, **Secondary Bone Axis**: **Y Axis**, **Armature FBXNode Type**: **Null**, **Add Leaf Bones** **OFF**

All commands were tested using these parameters.

6. Change log

Version 1.0.0

- First released version :)