



## Technical guide

### Understanding 3D Printing File Types: STL, 3MF, OBJ and G-code Explained

When a customer uploads a file for 3D printing, the first question we consider is not how large the part is or what material it requires. The first question is much simpler: what file format are we working with?

File type plays a significant role in print reliability, geometry accuracy, material settings, and even colour information. Many users treat file formats as interchangeable, but they are not. Each format exists for a reason, and choosing the correct one can improve workflow efficiency and reduce print errors.

In this lesson, we will explore the four most common file types used in 3D printing: **STL, 3MF, OBJ, and G-code**. More importantly, we will examine when to use each one and why.

---

#### Why File Format Matters in 3D Printing

A 3D printer does not understand design software files such as SolidWorks, Fusion, or Blender files directly. Instead, those designs must be exported into a format that slicing software can interpret.

The file you export determines:

- How geometry is stored
- Whether units are preserved
- Whether colour and material data are included
- Whether slicing settings travel with the model
- How easily the file can be repaired

Using the wrong format can introduce scaling issues, lost metadata, corrupted geometry, or unnecessary workflow steps.

Understanding file formats allows you to control your process rather than react to problems later.

---

#### STL – The Industry Standard

The STL file (Standard Tessellation Language or StereoLithography) is the oldest and most widely recognised

format in 3D printing.

STL files describe geometry using triangles. Every curved surface is approximated by a mesh of small triangular faces. The smaller and more numerous the triangles, the smoother the surface appears.

However, STL files contain **only geometry**. They do not store:

- Units (mm or inches)
- Colour
- Material data
- Print settings

This simplicity is both a strength and a limitation.

## Benefits of STL

STL is universally supported. Nearly every slicer, CAD package, and 3D printer workflow accepts STL files. Because the format is simple, files are generally stable and predictable.

For single-material functional parts, STL is often perfectly adequate.

## Limitations of STL

Because STL does not store units, scale errors are common. A part designed in millimetres may open as inches in another system.

STL also cannot store multiple bodies intelligently, colour data, or advanced metadata.

For simple prints, STL works well. For more complex workflows, it begins to show its age.

---

## 3MF – The Modern Alternative

3MF (3D Manufacturing Format) was developed to solve the limitations of STL.

Unlike STL, 3MF files store:

- Units
- Multiple bodies
- Colour information
- Material assignments
- Print settings (when exported from slicers)

This makes 3MF far more powerful for modern workflows.

## Why 3MF Is Often Better for 3D Printing

Because 3MF preserves units, scaling errors are eliminated. What you design is what opens in the slicer.

3MF also allows multi-material and multi-colour information to travel with the file. If you are using a multi-material printer, this is critical.

Additionally, some slicers store support placement and slicing profiles inside the 3MF file. This makes collaboration easier because another user can open the file exactly as intended.

## When to Choose 3MF

If you are working in a modern slicer and want to preserve configuration, or if you are printing multi-material parts, 3MF is usually the better choice.

For professional workflows, 3MF is increasingly becoming the preferred format.

---

## OBJ – Geometry with Visual Data

OBJ files are commonly used in animation and 3D modelling rather than manufacturing, but they also have applications in 3D printing.

Like STL, OBJ files describe geometry through mesh data. However, OBJ files can also store:

- Colour data
- Texture references
- Surface information

This makes OBJ useful for full-colour printing systems.

## When OBJ Makes Sense

If you are printing artistic models, figurines, or textured parts that require colour mapping, OBJ can be valuable.

However, OBJ files can be larger and more complex than STL. They may also include unnecessary data for simple functional prints.

For most engineering parts, OBJ is not required.

---

## G-code – The Machine Instruction File

G-code is very different from STL, 3MF, or OBJ.

While those formats describe geometry, G-code describes **movement instructions** for the printer.

Once a file is sliced, the slicer converts the model into G-code. This file contains:

- Exact nozzle movements
- Extrusion rates
- Temperature commands
- Fan speeds
- Layer heights

G-code is printer-specific. It is generated for a particular machine, with specific settings.

## Why You Don't Share G-code as a Design File

Because G-code contains machine instructions, it is not portable in the same way geometry files are.

A G-code file created for one printer may not work correctly on another due to:

- Different bed sizes
- Firmware differences
- Extruder configurations
- Temperature limits

For this reason, when sending a file to a professional 3D printing service, you should always send geometry files (STL or 3MF), not G-code.

---

## Choosing the Right File Type

So how do you decide?

If you are printing a simple single-material part and need universal compatibility, STL is safe and reliable.

If you want to preserve units, materials, and advanced data, 3MF is the superior modern option.

If you are working with colour textures or artistic models, OBJ may be appropriate.

If you are printing directly from your own slicer to your own machine, G-code is the final production file — but not something to share externally.

---

## Professional Workflow Recommendation

At BritForge3D, we recommend:

- **3MF for most modern workflows**
- **STL for compatibility and simple parts**
- **Avoid sending G-code unless specifically requested**

We re-slice all files to ensure compatibility with calibrated machines and optimised material profiles.

Sending geometry files instead of machine code allows us to ensure print strength, dimensional accuracy, and material suitability.

---

## Common Mistakes to Avoid

One of the most common mistakes is exporting STL with extremely low mesh resolution. This results in visible flat facets on curved surfaces.

Another mistake is forgetting to check units before export.

Finally, some users send G-code assuming it guarantees identical results. In reality, it can introduce compatibility risks.

---

## Final Thoughts

Understanding file formats is not just about software preference. It is about workflow control.

Choosing the correct file format improves reliability, reduces errors, and supports better collaboration between designers and print services.

As 3D printing evolves, 3MF is increasingly replacing STL for professional workflows. However, STL remains deeply embedded in the ecosystem.

In the next guide, we will explore how to prepare files properly before upload — including mesh repair, manifold checks, and design validation.

