

TICRA Gerber Export Procedure (TICRA Tools 23.1)

TICRA Setup: Commands > Add > Export Gerber Files > Target: [Select Surface] > Submit Job

Locate the output folder, where two files will be generated as follows: [File Name]_GP.gbr and [File Name]_elements.gbr, Verify the design is valid with Altium Designer Cam Viewer. The common file format is MLA4.4, but the file format from TICRA is a MTI6.6 Format which may not work with some fabricators. In this case, use a gerber viewer (Gerbv.exe/GerberEditorExe) to check that it can be loaded.

Python Script File Conversion: GerberTicraConvert.ipynb

```
##### Ticra Gerber to Altium/Gerv.exe Compatible Gerber #####
import re
from pathlib import Path

def float_to_fixed_44(x):
    """Convert float to 4.4 format integer string."""
    return str(int(round(x * 10000)))

input_path = Path("ticragerber.gbr")

with open(input_path, "r") as f:
    for line in f:
        if line.strip() == "G36*": #Strip Ticra Header
            break
    nsi_content = f.readlines() # Read and Store Rest of Data

# Regex to match lines with coordinates
coord_pattern = re.compile(r'X([-\.d.]+)\s*Y([-\.d.]+)\s*D(\d+)\s*')

converted_lines = []
for line in nsi_content:
    match = coord_pattern.search(line)
    if match:
        x_val = float(match.group(1))
        y_val = float(match.group(2))
        d_code = match.group(3)
        new_line = f"X{float_to_fixed_44(x_val)}Y{float_to_fixed_44(y_val)}D{d_code}\n"
        converted_lines.append(new_line)
    else:
        converted_lines.append(line)

mla_header = [
    "G04 #@! TF.GenerationSoftware,Altium Limited,Altium Designer,25.0.2 (28)*\n",
    "G04 Layer_Physical_Order=1*\n",
    "G04 Layer_Color=255*\n",
    "%FSLAX44Y44*\n",
    "%MOMM*\n", # Units Mode = mm
    "G71*\n", # RS-274-x Oleder Version Units Mode = mm
    "G04 #@! TF.FilePolarity,Positive*\n", # Positive = Drawn Shapes Add Copper
    "G01*\n", # Linear interpolation mode
    "G75*\n", # Circular interpolation with multi-quadrant mode
    "G36*\n", # Polygon Fill
    "%ADD11C,1.00000*\n" # Drill Hole Sizing Diameter = 1 mm
    "%ADD12C,0.62000*\n" # Diameter = 0.62000 mm
]

final_output = mla_header + converted_lines
with open("output_mla44.gbr", "w") as f:
    f.writelines(final_output)

Path("output_mla44.gbr").name
```