

Python Scripting How-To: Generating a Netlist

Summary

This script is a netlisting generation shell that can easily be customized. It will walk through the hierarchy for a specified schematic and print out the elements.

```
class Netlister:
    def __init__(self, awrde: mwo.CMWOoffice, schematic: mwo.CSchematic):
        self.awrde = awrde
        self.schematic = schematic
        self.subcircuits = set()
        self.datafiles = set()

    def output_nodes(self, element):
        for i, n in enumerate(element.Nodes, 1):
            pn = n.PortNumber
            nn = n.NodeNumber
            print(f'pin({i})=node({nn}) ', end='')

    def output_parameters(self, element):
        for p in element.Parameters:
            print(f'{p.Name}={p.ValueAsString} ', end='')

    def output_element(self, elem: mwo.CElement):
        awrde = self.awrde
        if 'SUBCKT' in elem.Name:
            # get the name of the subcircuit, will be double quoted so remove them
            subckt_name = elem.Parameters('NET').ValueAsString.replace('"', '')
            # need to determine the type
            if awrde.Project.Schematics.Exists(subckt_name):
                self.output_sch_subcircuit(elem, subckt_name)
            elif awrde.Project.EMStructures.Exists(subckt_name):
                self.output_em_structure(elem, subckt_name)
            elif awrde.Project.EM3DStructures.Exists(subckt_name):
                self.output_em3d_structure(elem, subckt_name)
            elif awrde.Project.DataFiles.Exists(subckt_name):
                self.output_datafile(elem, subckt_name)
            else:
                pdb.set_trace()
                raise ValueError(f'Cannot find subcircuit named "{subckt_name}"')
        else:
            try:
                (name, _) = elem.Name.split('.')
            except:
                # elements like GND don't have ID's
                name = elem.Name
            print(f'{name} ', end='')
            self.output_nodes(elem)
            self.output_parameters(elem)
            print()

    def output_em_structure(self, elem: mwo.CElement, name: str):
        print(f'EM_Structure ', end='')
        self.output_nodes(elem)
        self.output_parameters(elem)
        print()

    def output_em3d_structure(self, elem: mwo.CElement, name: str):
        print(f'3DEM_Structure ', end='')
        self.output_nodes(elem)
        self.output_parameters(elem)
        print()

    def output_sch_subcircuit(self, elem: mwo.CElement, name: str):
        print(f'Subcircuit ', end='')
        self.output_nodes(elem)
```

```

        self.output_parameters(elem)
        self.subcircuits.add(name)

def output_datafile(self, elem: mwo.CElement, name: str):
    print(f'Datafile ', end='')
    self.output_nodes(elem)
    self.output_parameters(elem)
    self.datafiles.add(name)

def generate(self):
    for element in self.schematic.Elements:
        self.output_element(element)

    # now we have to netlist all the referenced schematics
    while self.subcircuits:
        schematic_name = self.subcircuits.pop()
        # get the schematic by name
        schematic = self.awrde.Project.Schematics(schematic_name)
        print(f'\nDEFINE_SCHEMATIC {schematic_name}')
        for element in schematic.Elements:
            self.output_element(element)
        print(f'END_SCHEMATIC {schematic_name}')
        print()
    while self.datafiles:
        datafile = self.datafiles.pop()
        print(f'INCLUDE "{datafile}"')

```

for convenience this is included in `pyawr.helpers` and can be invoked by creating a netlister object for a schematic and calling `generate()`.

```

from pyawr.helpers import Netlister
import pyawr.mwooffice as mwo

awrde = mwo.CMWOoffice()
# open project you want
s = awrde.Project.Schematics('name of schematic')
netlister = Netlister(awrde, s)
netlister.generate()

```