OpenDSS is able to **export** distribution network models in a candidate "CIM100" version, which is based on **extensions** of CIM17, which is to be standardized in early 2020. These extensions are important in modeling North American distribution systems and will be tested in July 2020 at CIM interoperability tests that are sponsored by the CIM Users Group[1].

The CIM extensions and model export functions have been maintained under the U. S. Department of Energy's GridAPPS-D project [1-4]. In particular, [1] describes the overall project and the CIM transformer model, [2] describes the CIM unbalanced line model and the advantages of a triple-store database for CIM, [3] describes the relevant CIM UML while [4] contains a Java tool for **importing** CIM files. This Java tool uses the Blazegraph triple-store; it supersedes the legacy OpenDSS *cdpsm_import* code that is archived on SourceForge, but no longer maintained. The tools at [4] support CIM import, OpenDSS export from CIM, and GridLAB-D export from CIM. They also include OpenDSS converters from two commercial tools, but these have not been comprehensively tested. Instead, one goal of the GridAPPS-D project is to encourage CIM adoption by many tool vendors, which would lower the burden of model conversion and lower other costs of integration.

The common distribution power system model (CDPSM) is a CIM profile standardized in IEC 61968-13. To comply with the standard, OpenDSS can export a model into six different sub-profiles:

- Functional (FUN) – defines nearly all components with names, phasing, grounding, base voltage, feeder containment, terminals, operational limits, DER production source and operating status
- Electrical Properties (EP) – voltage and power ratings, control settings, impedances, etc.
- Topological (TOPO) – adds ConnectivityNodes, i.e., buses. There is a one-to-one correspondence of ConnectivityNodes and TopologicalNodes; OpenDSS doesn't use TopologicalNodes
- Catalog (CAT) – impedances and ratings defined in xfmrcode, linecode, spacing, wiredata, etc.
- Geographic (GEO) – locations, coordinate system and coordinates
- Steady-State Hypothesis (SSH) – load and DER p and q values, source voltages, switch states

A valid power flow model requires all six, with a possible exception of GEO. For convenience, OpenDSS can export all six into a single file. An open-source tool to combine them will also be provided at [5].

In order to test the CIM export function, use the *cim_test.dss* file under the Test directory installed with OpenDSSCmd:

```
redirect IEEE13_CDPSM.dss
solve
uuids file=IEEE13Nodeckt_Base_UUIDS.dat
export cim100 substation=subname geo=region subgeo=subregion file=ieee13cdpsm.xml
export cim100fragments substation=subname geo=region subgeo=subregion file=ieee13cdpsm
export UUIDS

redirect IEEE13_Assets.dss
solve
uuids file=IEEE13NodecktAssets_Base_UUIDS.dat
export cim100 substation=subname geo=region subgeo=subregion file=ieee13assets.xml
export cim100frag substation=subname geo=region subgeo=subregion file=ieee13assets
export UUIDS
```

[1] Timing is uncertain due to the Coronavirus pandemic.

Line 1 reads a local copy of the IEEE 13-bus model, with PV, storage, a service transformer, and various switches added. Line 8 reads a local copy of the IEEE 13-bus model, with transformer codes and line spacings used. In combination, these two files test most of the CIM classes that can be exported.

Lines 2 and 9 solve the models, which is a prerequisite for CIM export.

Lines 3 and 10 read CIM mRID attributes that have been saved from a previous export. Without this, the mRID attributes will be re-generated randomly each time a model is exported. The intent of the mRID is to maintain persistent object identification, so once established, these values need to be reused. (Note: during the very first CIM export, Lines 3 and 10 would be commented out because there are no previous mRID values available. In that case, random mRID values will be generated in Lines 4-5 and 11-12.) The mRID is implemented as a Universally Unique Identifier (UUID) version 4, according to IETF RFC 4122. Every named object in OpenDSS can have a mRID assigned or generated randomly when needed. During CIM export, there are other CIM identified objects that must have mRIDs, even if they are not identified with names in OpenDSS. The CIM Terminal is one example, and the individual Wires Phase objects provide other examples. The *uuids* command calls in Lines 3 and 10 take care of these extra mRIDs.

Lines 4-5 and 11-12 create the CIM XML files. The CIM feeder name will be the same as the OpenDSS circuit name. In addition, the CIM models must have substation, geographic region and subgeographic region objects, which are optional named attributes on lines 4-5 and 11-12. If you run these export commands repeatedly on an OpenDSS file that didn't change, the CIM XML files should not change either, because of the *uuids* commands invoked on Lines 3 and 10.

Lines 4 and 11 create single files that contain all six CDPSM profiles within *ieee13cdspm.xml* and *ieee13assets.xml*, respectively. Lines 5 and 12 export the same information into six separate files for each circuit, using *ieee13cdpsm* and *ieee13assets* respectively as the root file names. The rest of the file name will be *_FUN.XML*, *_EP.XML*, *_TOPO.XML*, *_CAT.XML, _GEO.XML* or *_SSH.XML*, indicating the profile contained within.

Lines 6 and 13 write all the mRID, aka UUID, values to the default file names. If there have been changes to the OpenDSS model, copy those UUID files to the file names used in Lines 3 and 10. That way, mRID values will be properly maintained.

The mRID lists are created and freed for each new CIM export, if you follow either of these patterns, <u>after defining and solving the circuit</u>:

- First export of CIM: *export cim100* and then *export uuids*
- Subsequently: *uuids* followed by *export cim100* and then finally *export uuids*

References

[1] R. B. Melton *et al.*, "Leveraging Standards to Create an Open Platform for the Development of Advanced Distribution Applications," *IEEE Access,* vol. 6, pp. 37361-37370, 2018, doi: 10.1109/ACCESS.2018.2851186.

[2] T. E. McDermott, E. G. Stephan, and T. D. Gibson, "Alternative Database Designs for the Distribution Common Information Model," in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, 16-19 April 2018 2018, pp. 1-9, doi: 10.1109/TDC.2018.8440470.

[3] Pacific Northwest National Laboratory. "GridAPPS-D." http://gridappsd.readthedocs.io/en/latest (accessed 2020).
[4] Pacific Northwest National Laboratory. "CIM Importer and Test Files." https://github.com/GRIDAPPSD/Powergrid-Models (accessed 2020).
[5] Pacific Northwest National Laboratory. "CIMHub." https://github.com/GRIDAPPSD/CIMHub/tree/develop (accessed 2020).