

Getting Started with OpenDSS

Start your search for OpenDSS files on EPRI's link page:

<http://smartgrid.epri.com/SimulationTool.aspx>

Downloads

The Installer for the latest official release may be found at this link

<http://sourceforge.net/projects/electricdss/files/>

Download both the OpenDSS Installer file and any other files of interest on the download page.

Release version installers are posted at irregular intervals. If you have a Delphi compiler, up-to-date versions may be built from the source code, which contains the latest changes. If you don't have a compiler that can build the program, the latest beta builds are posted at these locations:

<http://svn.code.sf.net/p/electricdss/code/trunk/Distrib/x86/> (32-bit version)

<http://svn.code.sf.net/p/electricdss/code/trunk/Distrib/x64/> (64-bit version)

The latest build is generally just fine and you may need it to access the latest features reported in the Wiki. However, there is a small risk some bug has crept into the code as changes were made. Report any problems as soon as possible.

The Files

Presently, there are 5 program files:

1. OpenDSS.EXE (the standalone exe with a text interface)
2. OpenDSSEngine.DLL (the in-process COM server)
3. KLUSolve.DLL (Sparse matrix solver)
4. DSSView.EXE (DSS graphics viewer)
5. IndMach012.DLL (DLL for the GENERATOR's induction machine model)

Instead of using the Installer, you may simply copy these files to a directory (folder) of your choice. The Installer will automatically register the COM server DLL. If you choose manual installation, and wish to access the COM server, you will have to register the server yourself (see below).

It is recommended that you first use the Installer to make the initial installation. Thereafter, you may update your version(s) of OpenDSS simply by overwriting OpenDSS.EXE and OpenDSSEngine.DLL with the newer version from the beta build sites above.

Registering The COM Server

If you intend to drive OpenDSS from another program such as Matlab, MS Excel, Python, C++, or whatever, you will need to register the in-process COM server, OpenDSSEngine.DLL. You can skip this step if you do not intend to do this.

The Installer will automatically do this step during installation. If you have used the Installer, you may skip this step.

Registering is accomplished by issuing the following command to the DOS (Command) prompt

while in the OpenDSS directory, for example:

```
C:\OpenDSS>regsvr32 opendssengine.DLL
```

Assuming you did everything right, you should get a message that says the registration succeeded.

If you start up the Windows RegEdit tool, you should be able to find "OpenDSSEngine.DSS" in the server. Another way to test this is to go into the VBA macro editor (alt-F11) in any of the Microsoft Office tools (e.g., Excel) and see if "OpenDSS Engine" shows up as an *Available Reference* under the Tools>References menu.

Note that EPRI PS174 members who install programs such as DGScreeener should register the server (the DGScreeener installs an older version if you do not).

Updates

For most updates, unless otherwise noted, you will need to replace only the OpenDSS.EXE and the OpenDSSEngine.DLL files. Generally, you should not have to re-register the OpenDSSEngine.DLL server, but it won't hurt if you do. Only one interface (OpenDSSEngine.DSS) is registered and we seldom change it. The other interfaces are created at run time.

If course if an Installer is available for the updated version, you may use it.

Note that if you have installed a version manually, the Installer may not find it and overwrite it. Windows will keep a link to the old version of OpenDSSEngine.DLL and will invoke it when you use a program that had used it in the past, such as a MATLAB program for an old project. If you find some feature missing that you were expecting to find, this is a likely explanation. Locate and delete the older version.

Accessing the Source Code

The source code repository on Sourceforge.net is managed through the SVN source code management system. On Windows, you might consider a client such as TortoiseSVN, installed as follows:

1. Install a 32- or 64-bit TortoiseSVN client, as appropriate for your computer, from <http://tortoisesvn.net/downloads.html>.
2. From the TortoiseSVN General Settings dialog and click the last check box, to use "_svn" instead of ".svn" for local working directory name.

Then, to grab the files from SourceForge:

- a. create a clean directory such as "c:\opendss"
- b. right-click on it and choose "SVN Checkout..." from the menu
- c. the [http://electricdss.svn.sourceforge.net/svnroot/electricdss repository URL] is

```
https://svn.code.sf.net/p/electricdss/code/trunk
```

Change the checkout directory if it points somewhere other than what you want.

That's basically all there is to it. The client will retrieve all the latest source code from Sourceforge.net. This may take several minutes. This will include the available documentation files, examples, training powerpoint files, and the OpenDSS versions of most of the IEEE Test Feeders and EPRI Test Circuits.

You will notice that the folder icons have changed for the folders that are under source control.

Updating the Source Code

Simply right-click on the main folder on your computer and select SVN Update. You should do this periodically to keep up with the latest changes.

Running the Program

To run the executable, simply open OpenDSS.EXE. You may wish to create a shortcut on your desktop or pin this file to the Start menu.

Starting the COM Server from Another Program

In MATLAB

- DSSObj = actxserver('OpenDSSEngine.DSS');

In VBA

- Public DSSObj As OpenDSSEngine.DSS
- Set DSSObj = New OpenDSSEngine.DSS

In PYTHON

- self.engine = win32com.client.Dispatch("OpenDSSEngine.DSS")

In C#

- (Project >Add Reference ... select OpenDSSEngine)
- ...
- Using OpenDSSEngine;
- Public DSS DSSObj;
- DSSObj = new DSS();

In Delphi

- {Import Type Library}
- Type DSSObj : IDSS;
- ...
- DSSObj := coDSS.Create;

In C++

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    CoInitialize(NULL);
    try {
        OpenDSSEngine::IDSSPtr DSSObj;
        DSSObj.CreateInstance(__uuidof(OpenDSSEngine::DSS));
        if(DSSObj->Start(0)){
            cout<<"openDSS Loaded!"<<endl;
            // Do whatever you want, here we show the version
            cout<<DSSObj->Version<<endl;
        };
    } catch (_com_error e) {
        printf("\n");
        printf("Error: %S\n", e.Description());
        printf("Error: %S\n", e.ErrorMessage());
    }
    cout << "Press anykey to exit.";
    cin.ignore();
    cin.get();
    return 0;
}
```

