

Introduction to the OpenDSS

The Open Distribution System Simulator (*OpenDSS*, or simply, *DSS*) is a comprehensive electrical system simulation tool for electric utility distribution systems. OpenDSS technically refers to the open-source implementation of the DSS. It is implemented as both a stand-alone executable program and a COM DLL designed to be driven from a variety of existing software platforms. The executable version adds a basic user interface on to the solution engine to assist users in developing scripts and viewing solutions.

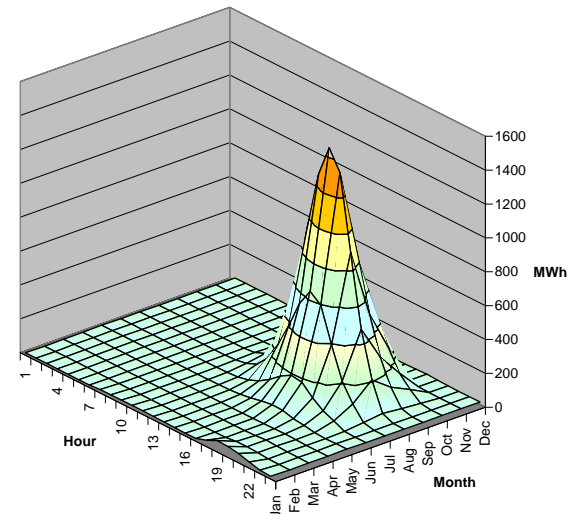
The program basically supports all rms steady-state (i.e., frequency domain) analyses commonly performed for utility distribution systems. In addition, it supports many new types of analyses that are designed to meet future needs, many of which are being dictated by the deregulation of US utilities and the formation of distribution companies worldwide. Many of the features found in the program were originally intended to support distributed generation analysis needs. Other features support energy efficiency analysis of power delivery and harmonics analysis. The DSS is designed to be indefinitely expandable so that it can be easily modified to meet future needs.

The OpenDSS program has been used for:

- Distribution Planning and Analysis
- General Multi-phase AC Circuit Analysis
- Analysis of Distributed Generation Interconnections
- Annual Load and Generation Simulations
- Wind Plant Simulations
- Analysis of Unusual Transformer Configurations
- Harmonics and Interharmonics analysis
- Neutral-to-earth Voltage Simulations
- Development of IEEE Test feeder cases
- And more

The program has several built-in solution modes, including:

- Snapshot Power Flow
- Daily Power Flow
- Yearly Power Flow
- Harmonics
- Dynamics
- Faultstudy
- And others ...



Through the COM interface, the user is able to add other solution modes and features externally and perform the functions of the simulator, including definition of the model data. Thus, the DSS could be implemented entirely independently of any database or fixed text file circuit definition. For example, it can be driven entirely from a MS Office tool through VBA, or from any other 3rd party analysis program that can handle COM. Users commonly drive the OpenDSS with the familiar Mathworks MATLAB program. This provides powerful external analytical capabilities as well as excellent graphics for displaying results.

Many users find the scripting interface available with the stand-alone executable version sufficient for nearly all their work. As users find themselves repeatedly needing a feature for their work, the feature is implemented within the program and connected to the text-based command interface.

The COM interface also provides direct access to the text-based command interface as well as numerous methods and properties for accessing many of the parameters and functions of the simulator's models. Through the command interface, user-written programs can generate scripts to do several desired functions in sequence. The input may be redirected to a text file to accomplish the same effect as macros and also provide some database-like characteristics. Many of the results can be retrieved through the COM interface as well as from various output files. Output files are typically written in Comma-separated Value (CSV) format that imports easily into other tools for post processing.

The experienced software developer has two additional options for using the OpenDSS tool:

1. Downloading the source code and modifying it to suit special needs.
2. Developing DLLs that plug into generic containers the OpenDSS provides [*This interface is being revised as of this writing.*]. This allows developers to concentrate on the model of the device of interest while letting the DSS executive take care of all other aspects of the distribution system model. Such DLLs can be written in most common programming languages.

The DSS structure is illustrated in Figure 1

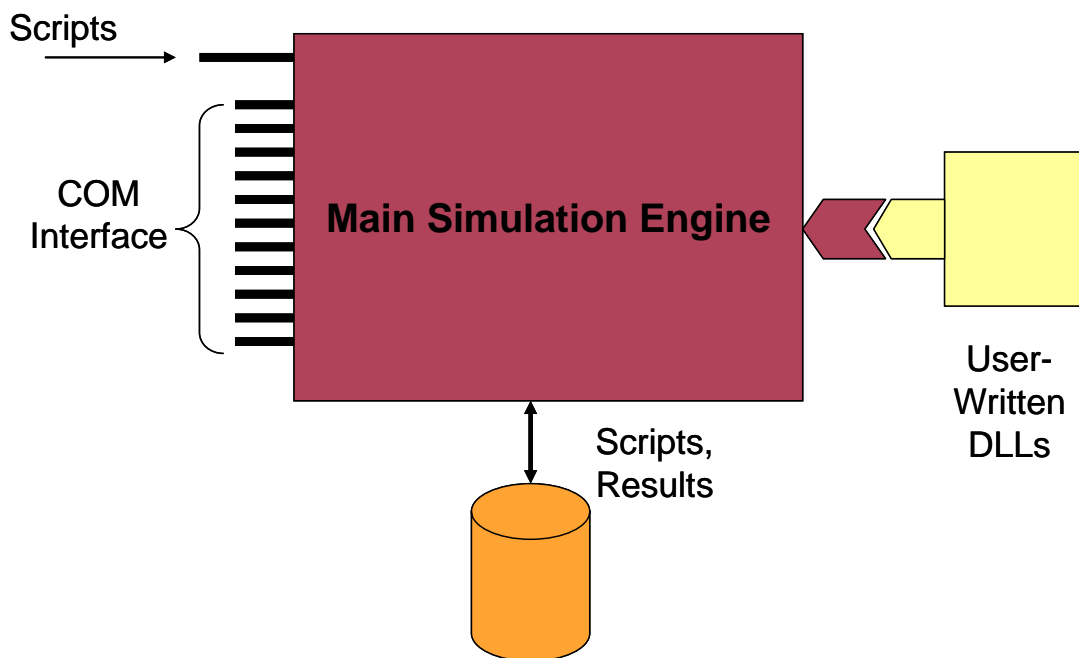


Figure 1. DSS Structure

Brief History and Objectives of the DSS

Development of the DSS began in April 1997 at Electrotek Concepts, Inc. by Roger Dugan and Thomas McDermott. The DSS was acquired by EPRI Solutions in 2004, which was united with EPRI in 2007. In 2008, EPRI released the software under open source to participate with other grid modernization efforts active in the Smart Grid area.

There were two events that triggered the development of the DSS in 1997:

1. EPRI had issued an RFP earlier for software to support the application of distributed generation to distribution systems. While Electrotek was not awarded a contract there had been enough thinking about the project to have arrived at a function spec for a simulator that would support the vast majority of electrical system analysis for evaluating DG for distribution planning purposes.
2. Roger Dugan was serving as Chair of the IEEE PES Software Engineering Working Group and one of the hot topics at that time was object-oriented programming and data representations. The late Mark Enns of Electrocon had issued a challenge to the group for someone to implement some of the principles we had been discussing in a power system analysis program. The distribution system simulator concept seemed the perfect vehicle to experiment with some of those ideas and also provide a useful tool to support the needs of a consulting company.

Prior to 1997, Electrotek had been performing DG studies for distribution planning using conventional distribution system analysis methods that are still employed by many tools today. We were well aware of the limitations of these methods and wanted a tool that was more powerful and flexible. One issue we discovered early on is that no two DG planning studies were exactly alike and we were constantly having to adjust our models. Thus, we needed a tool that would not lock us in to particular types of analysis. Other issues we set out to address were:

1. The distribution system model data we received from utilities came in all sorts of different formats, each with various strengths and various modeling limitations. One design goal was to develop an object-oriented circuit description language that minimized the conversion effort.
2. Window-based programs can be very user friendly, but we noticed some of the distribution system analysis tools actually limited what you can do by limiting the interaction to what is available on the dialog forms. By being fundamentally script-driven, the DSS program gets around much of the limited-dialog issue and allows the user to better adapt the analysis process to the problem at hand.
3. The greater value of DG is often found on the subtransmission system serving a distribution planning area. Many distribution planning tools represent only the radial distribution system. Because of the methods employed by the economists we were working with at the time, we wanted a tool that would allow us to model several substations and the distribution circuits between them simultaneously.
4. A key capability desired for the tool was to capture the *time- and location- dependent* value of DG. The location-dependent value is captured by modeling the DG in its actual location on the circuit. Capturing the time-dependent value requires extraordinary loadshape modeling capability.
5. There are many instances where it becomes necessary to model elements with many phases – not just three. For example, power poles in North America with multiple circuits may have as many as 4 circuits sharing a common neutral. Also, we wanted to be able to model what happens when a 69 kV line falls into the 13.2 kV distribution line. Or to model a communications signal passing from one voltage level to another through the

interwinding capacitance of the transformer – this was not possible in many traditional distribution system analysis tools.

6. We wanted a tool that could seamlessly incorporate harmonics analysis into the power flow analysis without requiring the user to laboriously enter nonlinear device models. We also recognized that we would need at least simple dynamics analysis sufficient for DG interconnection evaluation. Simple models are built in and this feature continues to be developed.
7. Recognizing that distribution automation was going to become increasingly important, we wanted a testbed upon which to evaluate control algorithms and their impact on the operation of the system.
8. Recognizing that it was impossible to anticipate all possible user needs, we wanted a program that would allow users to write their own models or solution procedures commensurate with their software capabilities.
9. We wanted a program that would simulate the behavior of devices on the distribution system as they would actually occur for changing load and system faults and other disturbances. This is important for modeling DG interactions and it also allows the tool to be used for many other things as well, such as energy efficiency analysis.

The present version of the OpenDSS has achieved the majority of these goals and has evolved into an extraordinary tool that has acquired many other features not commonly found in other distribution system analysis tools. While DG analysis continues to be one of its key uses, many other types of analysis have been performed with the tool.

The DSS is a general-purpose frequency-domain simulation engine that has special features for creating models of electric power distribution systems and performing many types of analysis related to distribution planning and power quality. It does not perform electromagnetic transients (time domain) simulations; all types of analysis are currently in the frequency domain (i.e., sinusoidal steady state, but not limited to 60 Hz).

Most electrical engineers learned how to write nodal admittance equations in their early University courses, and this is basically how the DSS represents distribution circuits. Each element of the system is represented by a “primitive” nodal admittance (Y) matrix. This is generally straightforward, although it can be tricky for some power system elements such as transformers. Each primitive Y is then coalesced into one large system Y matrix, and the system of equations representing the distribution system is solved using sparse matrix solvers like one might also find in MATLAB. One trick is that nonlinear behaviors of some devices (e.g., some load models) are modeled by current source injections, which some refer to as “compensation” currents. That is, the current predicted from the linear portion of the model that resides in the system Y matrix is compensated by an external injection to iteratively obtain the correct current. This is a common technique for representing loads in distribution system analysis tools. One advantage of this method is that it allows for quite flexible load models, which is important for performing some types of analysis such as done in energy efficiency studies.

The program's heritage is closer to a harmonic flow analysis program, or even a dynamics program, than a typical power flow program. This may seem a strange place to start designing a tool that will be used a great deal for power flow studies, but it gives the tool great modeling flexibility particularly for accommodating all sorts of load models and unusual circuit configurations.