# Software Requirements

Objectives: Support EPRI development of advanced distribution analysis tools via enhancements to the OpenDSS platform. Re-architect and document OpenDSS for better cross-platform capability, making it more accessible to a wider pool of software developers, and to prepare it for exploiting high-performance desktop computing and advanced user interfaces.

1. Software shall run on both Windows (Vista or later) and Linux (Ubuntu) operating systems.
2. Software shall run in both 32-bit and 64-bit versions on all supported operating systems.
3. The software shall not use any commercial third-party components.
4. Development tool requirements are:
   1. Delphi XE Professional shall be used for Windows platform development.
   2. Free Pascal and Lazarus shall be used for Linux platform development.
   3. gnuplot[[1]](#footnote-1) shall be used for graphical output on all operating systems.
   4. KLUSolve shall be used for sparse matrix solutions on all operating systems.
   5. Microsoft Visual Studio 2008 shall be used to produce a 64-bit version of KLUSolve for Windows.
   6. GNU C/C++ compilers shall be used to produce 32-bit and 64-bit versions of KLUSolve for Linux.
5. The software shall be tested on all published IEEE test feeders on all supported platforms, comparing new solutions to the existing OpenDSS solutions, using the following criteria:
   1. Injected P and Q at the source, independently by phase. For wye systems, the phases are line-to-neutral and for delta systems, the phases are line-to-line. This comprises 6 values.
   2. Summation of I2R and I2X losses in power delivery elements (lines and transformers), summing the series (load) and shunt (no-load) losses over all phases. This comprises 4 values.
   3. Minimum and maximum voltage by phase, for each voltage level. This comprises 6 values for the 13-bus test feeder.
   4. If applicable, the tap setting of each regulating control. This comprises 3 values for the 13-bus test feeder.
   5. If applicable, the on/off status of each switch and capacitor bank. Assuming ganged operation, this comprises 3 values for the 13-bus test feeder, all defined as “ON”.
   6. If applicable, the mechanical speed and air-gap torque of each rotating machine.
   7. The sum-of-squares difference between voltage profiles at every node (i.e. by phase), normalized by the number of nodes. This comprises 3 values.
6. The OpenDSS shall be re-factored into three modules:
   1. Kernel – with simulation and file input/output functions only
   2. Script Interface – with non-graphical input and output for execution in batch, scripted, or service mode
   3. GUI – with graphical input and output for interactive execution
7. The Kernel shall support code-level function calls from other software, through implementation as a dynamic link library (DLL) on Windows and a shared object library (SO) on Linux.
8. The Script Interface shall support in-process Component Object Model (COM) on Windows.
9. The Script Interface shall support a Web service protocol, such as SOAP or REST, on both Linux and Windows. The Web service interface shall be the same on both platforms.
10. The Script Interface shall not have any GUI components.
11. Script Interface test cases shall include:
    1. Reconfiguration example from EPRI’s report “Example Assessments of Distribution Automation Using OpenDSS”. (COM and Web Service)
    2. Volt/var control example from EPRI’s report “Example Assessments of Distribution Automation Using OpenDSS”. (COM and Web Service)
    3. SampleDSSDriver.xls example on SourceForge (Windows COM only)
12. The new GUI module on Windows shall comprise a re-factored conversion of the existing GUI on Windows.
13. The new GUI module on Linux shall be based on a best-effort port from the new Windows GUI module to Lazarus on Linux, constrained by the available project budget. This requirement has the lowest priority.
14. The GUI module shall be tested using all example scripts (\*.dss) posted on SourceForge.
15. Installers shall be provided for both platforms.
16. The software documentation shall include:
    1. Updated software requirements
    2. Design documentation with UML package diagram, UML sequence diagrams for the Scripting Interface, and supporting narrative
    3. Build instructions and make files
    4. Change log, which is derived from Subversion file check-in comments
    5. Updated license file, release notes, and user manual as needed
    6. Test case document, including instructions to run the test cases and expected results

1. gnuplot uses lower-case “g” because it’s not actually part of the GNU project. It uses a non-GNU open source license that permits copy, modification, and re-distribution of the source code. [↑](#footnote-ref-1)