SuperHarmÒ

an Electrotek Harmonic Simulation Program

User's Guide

Version 4.3
November 6, 2002

Electrotek Concepts, Inc.
408 North Cedar Bluff Road, Suite 500
Knoxville, Tennessee 37923
Phone (865) 470-9222 / Fax (865) 470-9223
www.electrotek.com

Page Intentionally Left Blank

Legal Notice

Abstract

SuperHarm is a tool designed to assist the electric power engineer in evaluating harmonic concerns for electric power systems. This software allows the user to develop a computer model of the system of interest and explore variations on system loads and configurations along with the resulting impact on system frequency response and distortion levels.

SuperHarm utilizes TOP, The Output Processor to visualize simulation results. The program takes advantage of the Microsoft Windows Graphical User Interface and clipboard to allow the user to easily transfer data to other Windows programs such as Microsoft Excel® and Microsoft Word®.

Credits

The SuperHarm Development Team:

Development:	Erich Gunther / Tom McDermott / Kelley Flatford
Documentation:	Tom Grebe
Marketing:	Sandy Smith

Page Intentionally Left Blank

C O N T E N T S

Page Intentionally Left Blank

INTRODUCTION

GETTING STARTED

This introduction is intended to make the installation process of **SuperHarm**® a trouble-free experience. The following sections provide a step-by-step procedure on how to install and configure SuperHarm. Finally, at the end of this introduction is an overview of the SuperHarm User's Guide and how to receive technical support for the product.

This guide assumes you know the basics of using Microsoft Windows. You should know how to point, click, double-click, and drag. You should also know how to choose comments from menus, select options in dialog boxes, and enter, select, and edit text. See your *Microsoft Windows User's Guide* for details.

In This Chapter

Page Intentionally Left Blank

Overview

SuperHarm is a tool designed to assist the electric power engineer in evaluating harmonic concerns for electric power systems. This software allows the user to develop a computer model of the system of interest and explore variations on system loads and configurations along with the resulting impact on system frequency response and distortion levels.

SuperHarm contains a wide variety of device and source models. The program supports a generic harmonic voltage and current source model, long line corrected pi model, a simple RL branch model, a capacitor model, and a balanced three-phase coupled line model. Also included are models for the modal representation of geometrically-specified lines and cables, advanced load representations, induction and synchronous machines, and three-phase equivalents.

The program can solve both balanced and unbalanced three-phase systems. This is accomplished by using phase domain nodal admittance matrix techniques rather than sequence component solution methods.

The SuperHarm solution engine (Solver) reads a text file created by the user that describes the system to be simulated. SuperHarm's Circuit Description Language (CDL) consists of keywords representing device models as well as control commands. The devices are "connected" together by specifying alphanumeric names for the power system buses.

SuperHarm utilizes TOP, The Output Processor to visualize the simulation results. The program takes advantage of the Microsoft Windows Graphical User Interface and clipboard to allow the user to easily transfer data to other Windows programs such as Microsoft Excel® and Microsoft Word®.

Installing SuperHarm

This section describes the procedure for installing the SuperHarm program on your computer. The program is designed for the Microsoft Windows 95/98, NT 4.0, 2000, or XP operating systems. If your computer has sufficient memory to run Windows, and if the mouse and other peripherals are compatible with Windows, then the program should install and run without problems.

Computer Requirements

SuperHarm Version 4 is a 32-bit application for Windows 95/98, NT 4, 2000 or XP. This means that you must install it on a 486 or later-based computer with at least 8 MB of RAM. At least 2.7 MB of hard disk space will be required to install SuperHarm. You should also have a few MB free for output files. The setup program will update up to 3.3 MB of shared system files, but this results in little or no net increase in disk space used.

Program Installation

To begin the installation, select and double-click the **Setup.exe** file shown in Figure i- 1. This launches the SuperHarm setup program.

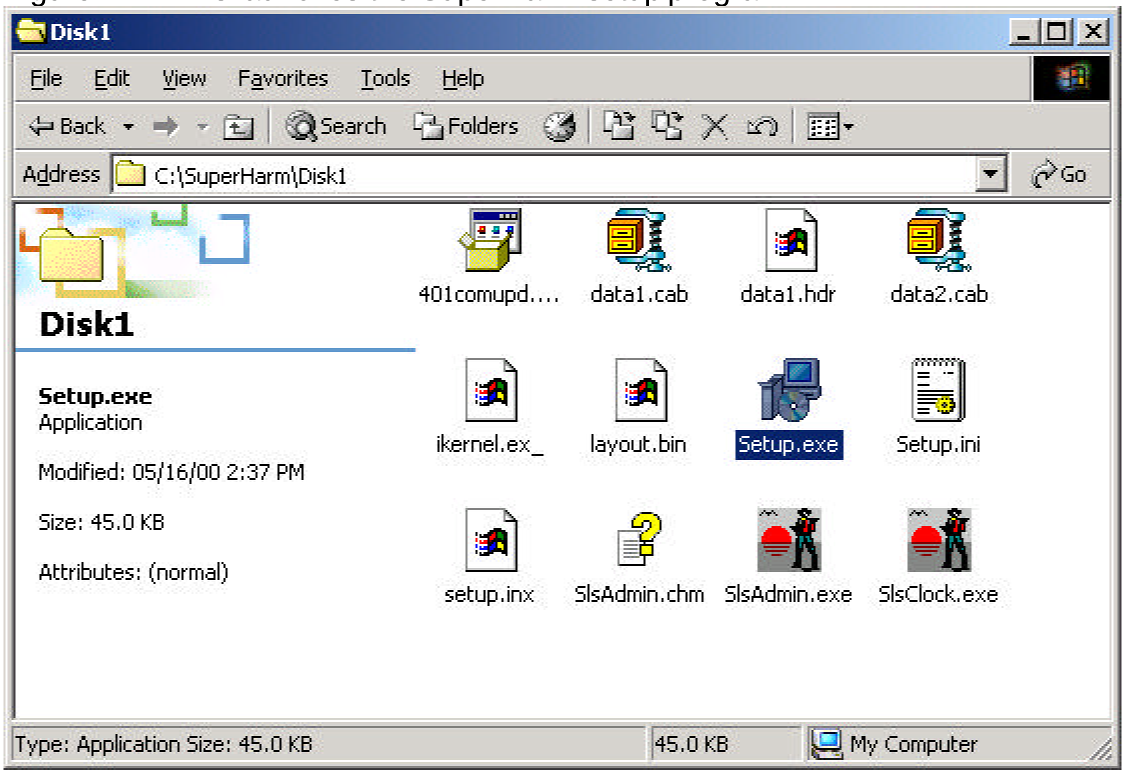


Figure i- 1: Setup.exe file

You may see a message like Figure i-2, if one of your system files must be updated before installing SuperHarm. If you see this message, select and double-click the **401comupd.exe** program that also appears in Figure i- 1. After completing the update, try to run **Setup.exe** again.

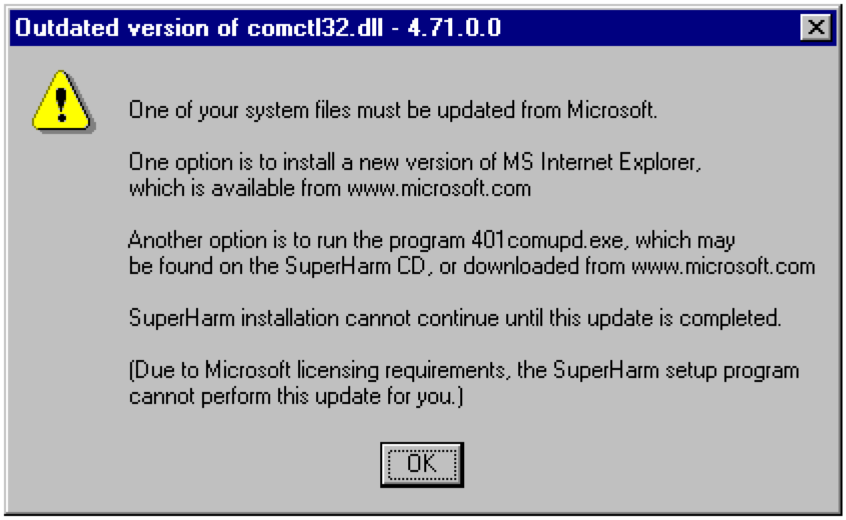


Figure i- 2: : Message for required comctl32.dll update

If you have previously installed SuperHarm, the message in Figure i- 3 may appear. You may choose to Repair or Remove the previous installation of SuperHarm, if you choose the Modify option, no action will be taken. The Repair option will install any updates that are found in the installation to SuperHarm.

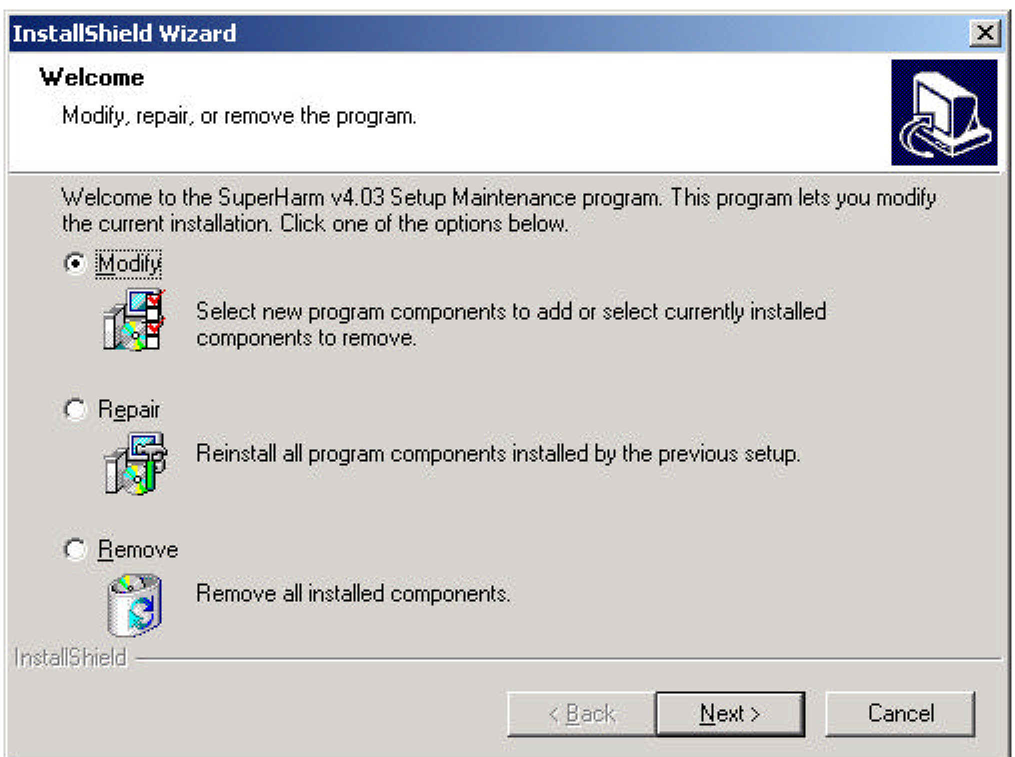


Figure i- 3: Previous Installation

The opening screen of the SuperHarm Setup welcomes you to the installation process(Figure i- 4). Select **Next** to continue to the next screen.

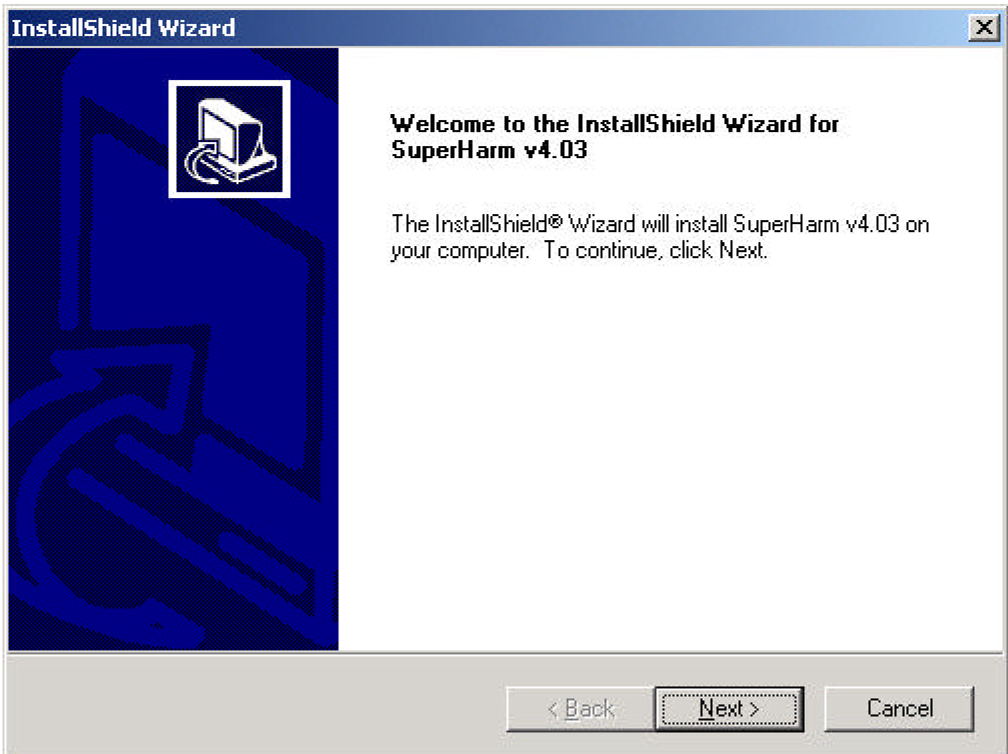


Figure i- 4: Welcome Window

The next screen you will see is the license agreement in Figure i- 5. Click **Yes** to continue.

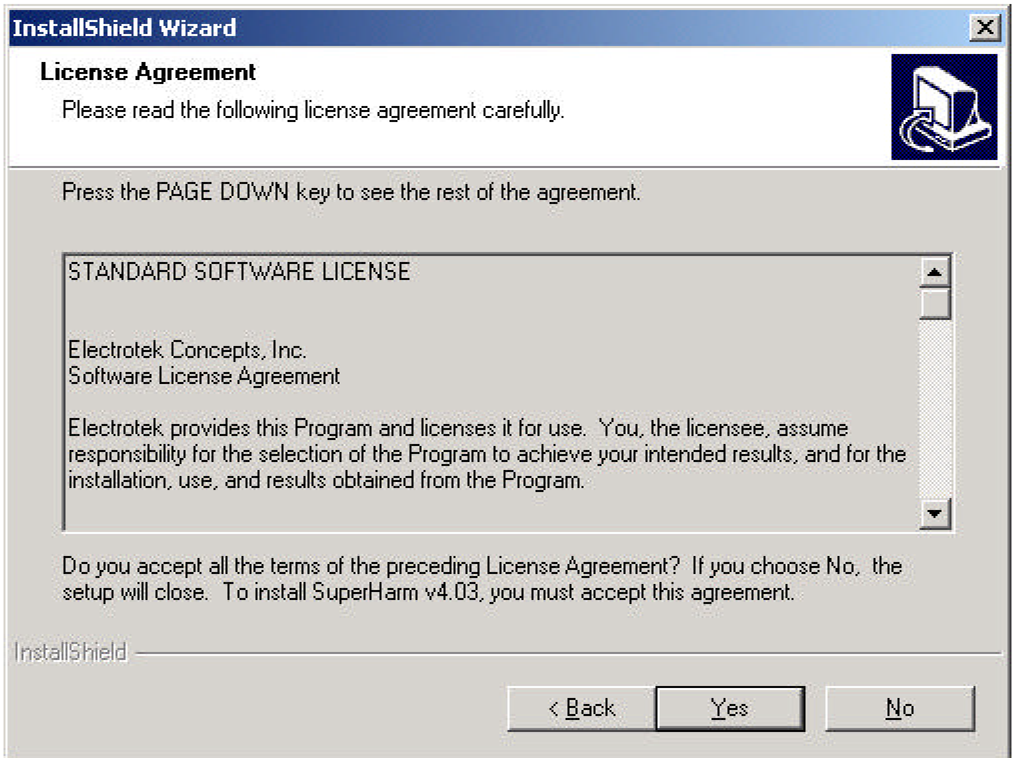


Figure i- 5: License Window

The installation program offers the option of changing directories if you do not want the program installed in the default directory. Simply click on the **Browse** button as shown in Figure i- 6. Select **Next** to continue to the next screen.

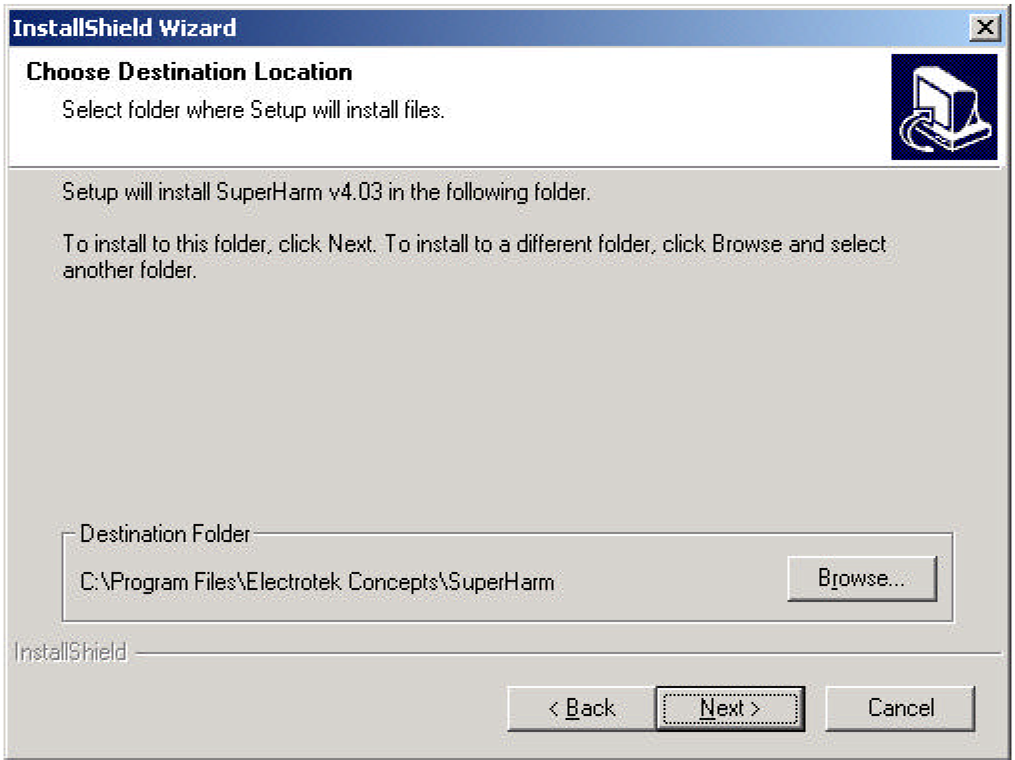


Figure i- 6:Choose Destination Location Window

The installation program then prompts for the folder where the program icons will be added. Select the desired folder, as illustrated in Figure i- 7, and click the **Next** button. Click **Finish** on the window that follows to complete the installation process.

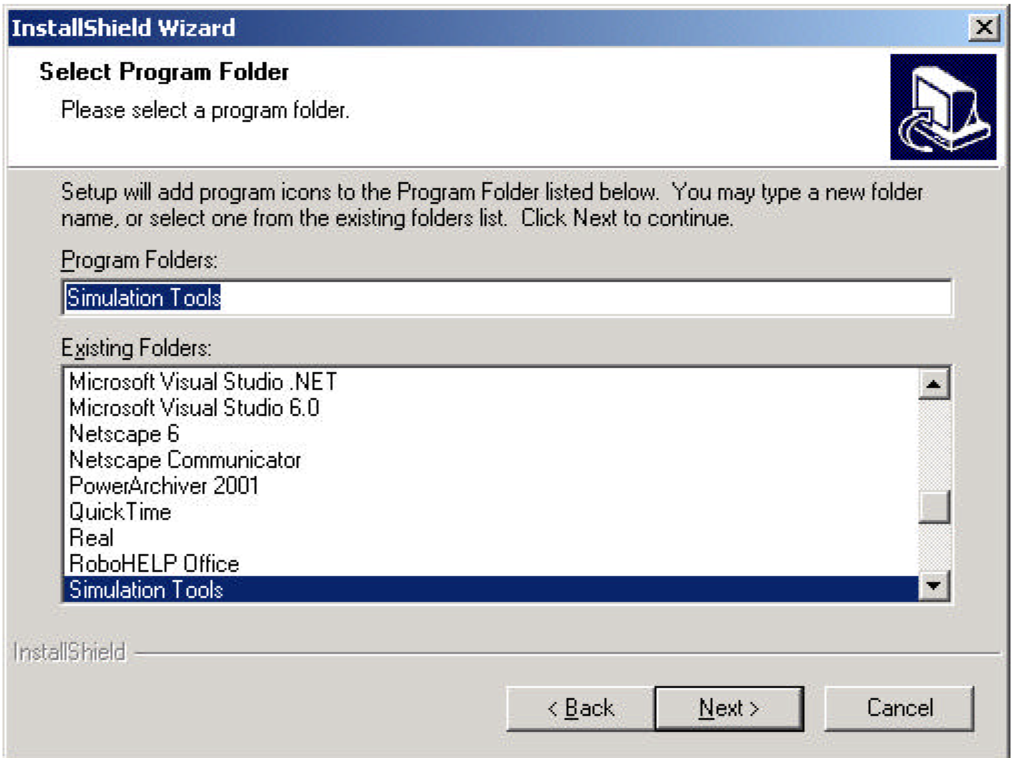


Figure i- 7:Select Program Folder Window

Uninstall

To uninstall the SuperHarm program, go to the Control Panel and open the Add/Remove Programs Properties dialog box shown in Figure i- 8. This figure may differ depending on the version of Windows installed on the computer. Follow the instructions given on the window.

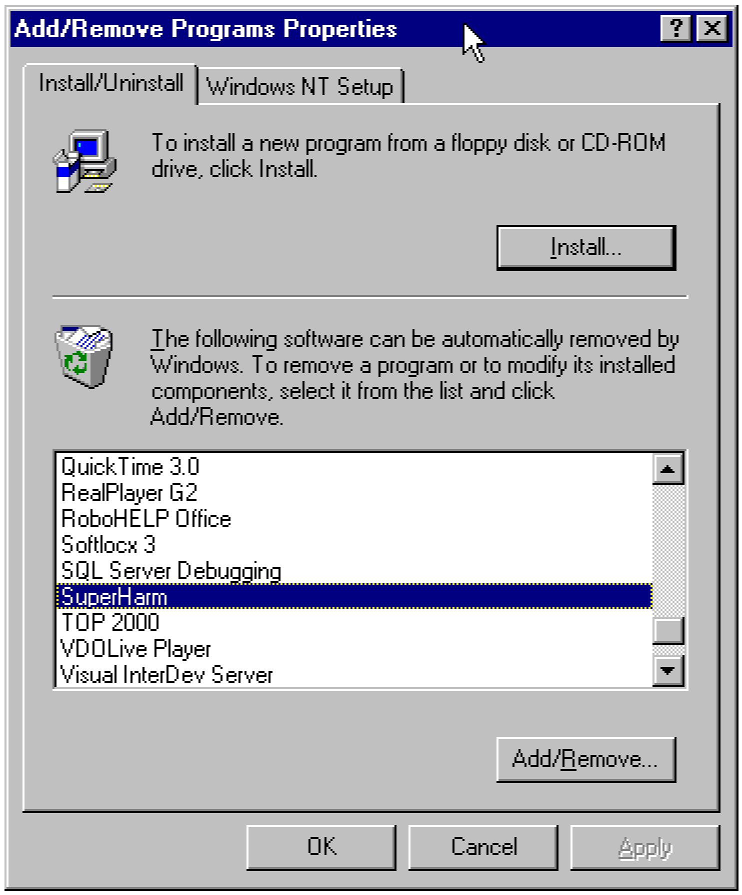


Figure i- 8:Add/Remove Programs Properties Window

Starting the Program

To start SuperHarm in Windows 95/98, NT, 2000, or XP, click the **Start** button on the taskbar and choose **SuperHarm** from the Start > Programs > Simulation Tools menu. The SuperHarm program window appears.

Entering a License Key

After you first install SuperHarm, the program will run for a 14-day grace period. The About Box in Figure i-10 shows the temporary license status and the number of days remaining. When the temporary license expires you will no longer be able to solve a case in SuperHarm. Reinstalling SuperHarm will not restore the 14-day grace period.

You can also use the **Help, About SuperHarm** menu command to get the dialog in Figure i- 9.

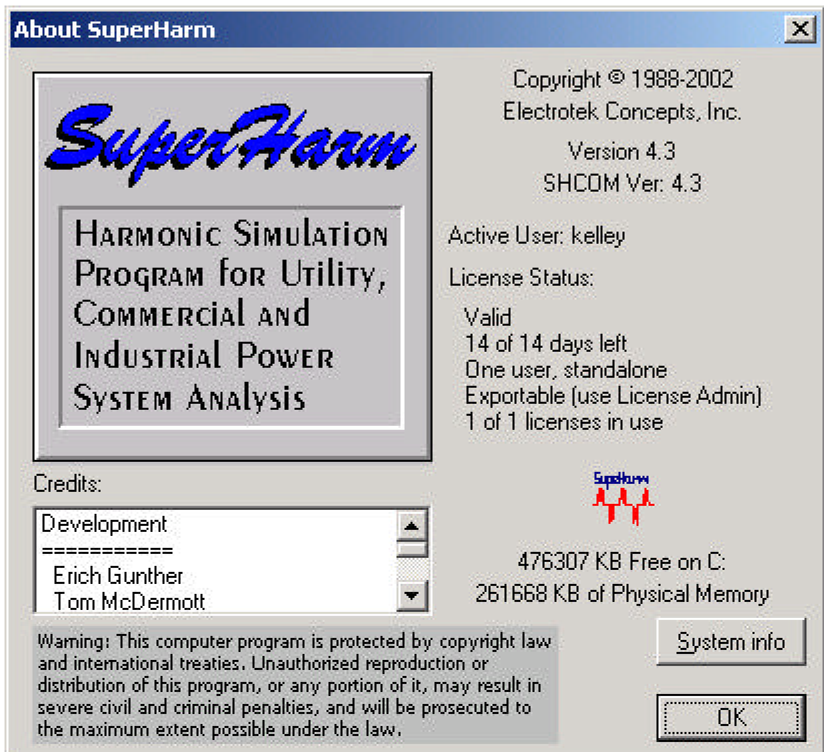


Figure i- 9:: About Box with a Temporary License

Before the license expires, you should obtain the machine-dependent license key from Electrotek. Until this has been done, the dialog in Figure i-11 will appear each time you start the program. Click **Register** to obtain the dialog in Figure i-12. The first time you do this, only the grayed edit field labeled "Reference Code" will be filled in. Email or fax this machine-dependent Reference Code to Electrotek. It will be used to generate your license key.

After noting the Reference Code, click **Cancel** in Figure i-12 and then **Continue** in Figure i-11. While waiting for your license key, you may continue to use SuperHarm by clicking Continue in Figure i-11 each time the program starts.

When you receive your license key from Electrotek, click **Register** in Figure i- 10 the next time the program starts. Type or paste your license key into the white edit box in Figure i- 11 and click **OK**. You should see the acknowledgment dialog in Figure i- 12. From now on, the about box dialog should appear as in Figure i- 13, indicating your version is registered.

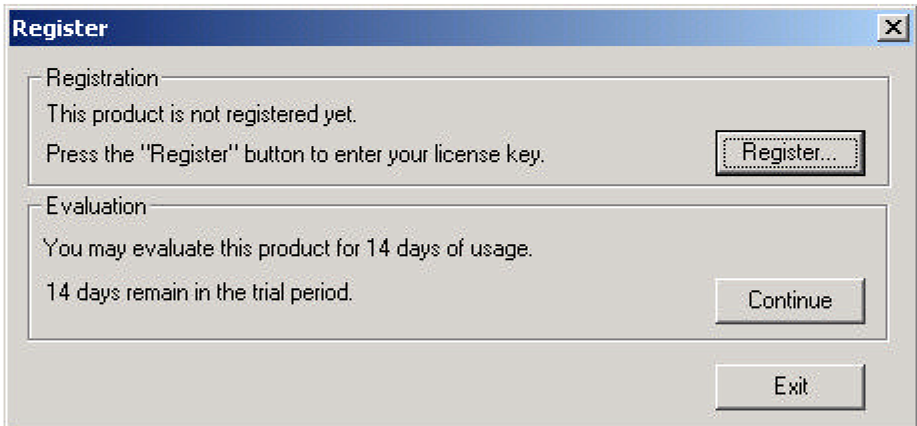


Figure i- 10:Click Register to obtain a Reference Code or enter a License Key

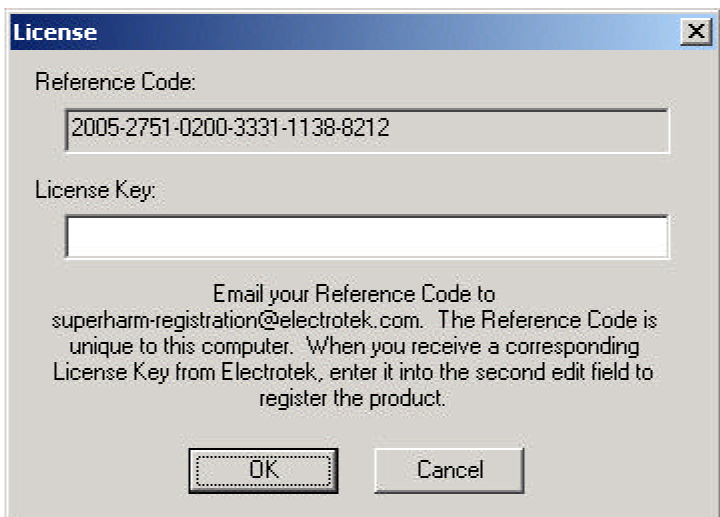


Figure i- 11:: Entering a License Key

Figure i- 12:: Message indicating successful registration

Figure i- 13:: About Box with a Valid License Key

The registration process creates license files in a folder under the Super-Harm application directory. The folder name will be a 20-digit product ID number; see the folder name in Figure i- 14 for an example. **These license files must not be deleted unless you wish to uninstall SuperHarm**.

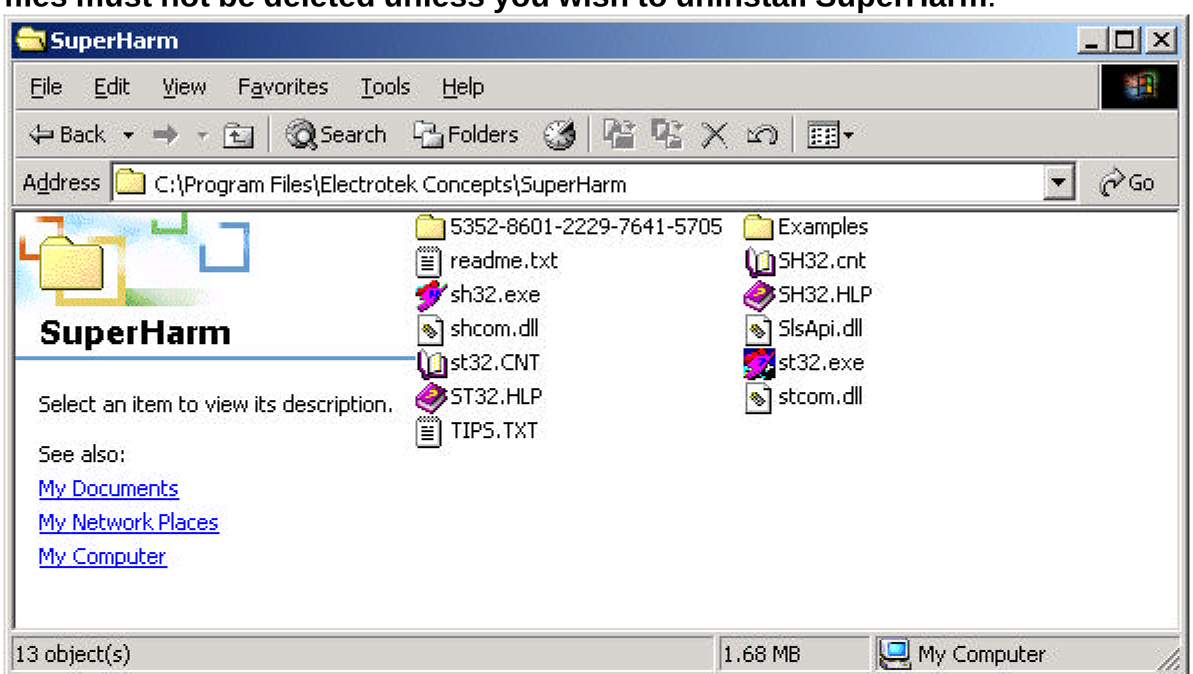


Figure i- 14: Installation directory for SuperHarm

Exporting a License Key

Your license key will not work on another machine, but you can transfer the licensed installation to another machine without contacting Electrotek. The procedure for this is:

- Install SuperHarm on the second machine, and note the Reference Code from the dialog in Figure i- 11 on that machine.
- On the first machine, that has a licensed copy, run the license administration program. Click the **Start** button on the taskbar and choose **License Admin** from the Start > Programs > Simulation Tools menu.
- Choose **Licence, Export** from the menu to obtain the dialog shown in Figure i- 15. Your Product Name combo box may list only SuperHarm, or it may list several products from Electrotek or other vendors that use the Sheriff licensing library.
- Choose "Electrotek SuperHarm" from the Product Name combo box, and then click **Export** to obtain the dialog in Figure i- 16.
- Enter the Reference Code from your second machine into the white edit box, and enter "1" for the number of Concurrent Users.
- ***Warning – this step is the "point of no return".*** Click the **Generate** button and write down the license key that it gives you in Figure i- 16.
- After you click **Close**, the next time you start SuperHarm on the first machine, the about box in Figure i- 9 will show that the license code has expired, and you will see a new Reference Code for the first machine in Figure i- 11. A new license key would be required to enable SuperHarm on the first machine.
- Use the new license key to register the copy on your second machine.

You can transfer the license back to the first machine, or any other computer, by following the procedure again. But note that you will have to enter the new license key on the second machine, before you can solve cases or transfer the license elsewhere.

If your need for SuperHarm on a second machine is temporary, you could simply install it and use it during the normal 14-day grace period. There is no limit on program capability, or marking in the output, for the unlicensed version. However, you will not be able to repeat this procedure again on that machine; it will require a valid license key.

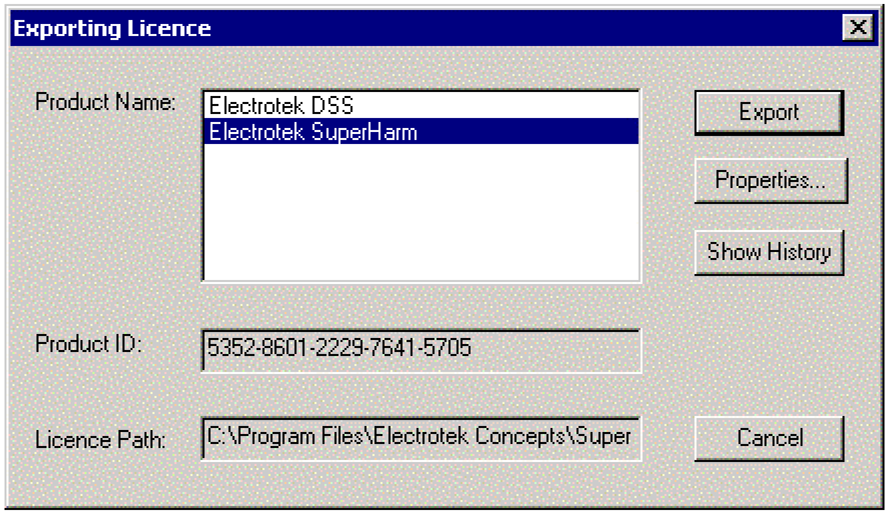


Figure i- 15:Transfering a License Key

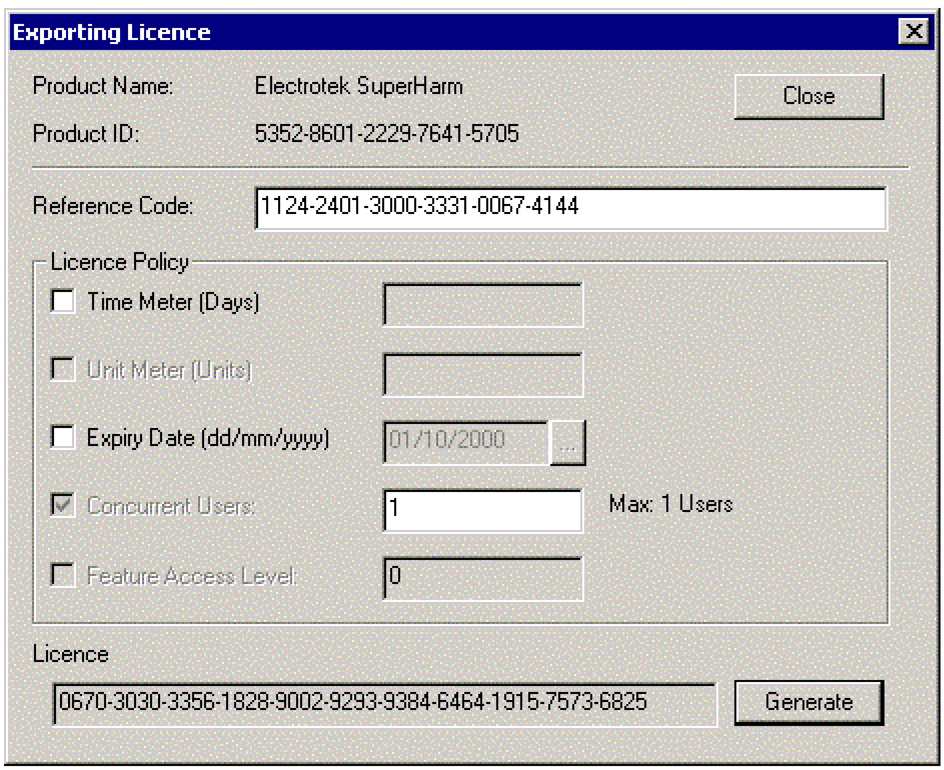


Figure i- 16: New License Key for the Second Machine

Network Licensing

Network licensing allows the use of SuperHarm on more than one machine. There will be a "concurrency limit" on the number of simultaneous users. With network licensing, a concurrency limit of 1 still allows several people to use SuperHarm on an intermittent basis.

SuperHarm "checks out" a license whenever displaying the about box, or whenever at least one file is open. The license is checked back in when SuperHarm closes, or if communications between the workstation and license server have been lost for about 10 minutes. If you close the last file in SuperHarm but leave the program running, the license will also be checked back in.

When the license has been checked in, another user on the network might check it out and it could become unavailable for your use. This can happen even when you keep SuperHarm running but close all the files.

If there is a network problem such that the license server is not available, the license will not be available and you won't be able to solve cases. This can happen even while SuperHarm is running; solutions in progress will complete, but you won't be able to start new solutions.

SuperHarm uses a third-party licensing library called Sheriff. A combination of license files and registry entries are used to manage the licenses. Once a license is checked out, SuperHarm notifies Sheriff every 5 minutes while the license is still in use. If these updates are missed, Sheriff will release the license, making it available for other users, but also making it unavailable for the first user. These updates have to account for system time differences between the license server and the user workstations.

To summarize the requirements and general procedures for network licensing:

- All workstations must have access to the license server machine over the network.
- Either SuperHarm, or just the SuperHarm license files, must be installed on the license server.
- All involved user accounts must have read-write permissions for the license files on the license server.
- The path to the license files must be visible to all client workstations as either a UNC or mapped drive path.

- A Sheriff system clock utility must be running on the license server.
- SuperHarm must be installed on each client workstation.
- The License Admin program must be used on each client workstation to enter the path to the license files on the license server.

Detailed procedures for the license server and the client workstation are described in the following two subsections.

License Server

If one person will be the primary SuperHarm user, the software can be installed on that person's workstation and registered normally. If a non-standalone license has been purchased, other casual users may install the software and register a path to the primary user's license files, as described in the next subsection. The **SysClock.exe** utility might also need to be installed and running on the primary user's workstation. Copy the program from the SuperHarm CD to the SuperHarm application directory, and then create a shortcut to it under the Startup folder.

The remainder of this subsection describes a more standard network licensing configuration. The SuperHarm license files and install image will be installed on a license server. All persons wishing to use the software will install it from the license server, and register a path to license files on the license server.

Figure i- 1 shows a directory set up this way. To produce a similar configuration for your network:

- Create a directory on your license server and copy all files from the SuperHarm CD to this directory.
- Double-click on **SlsAdmin.exe** in Figure i-1 to start the License Adminstration program.
- Choose **Licence, Registry** from the menu to obtain the dialog in Figure i- 17. Enter "Electrotek SuperHarm" for the Product Name and "5352-8601-2229-7641-5705" for the Product ID.
- Enter the Licence Path using local drive letters. For the UNC share \\SuperHarm\Disk1 in Figure i- 1, the local path on the license server is "c:\SuperHarm\Disk1" as entered in Figure i- 17.
- Click **Register** and then click **License**. The dialog in Figure i- 18 should appear. Email or fax the Reference Code from this dialog box to Electrotek.

- When you receive the license key from Electrotek, start the License Admin program on the license server. Use the menu command **Licence, Registry** and then click **License** to obtain Figure i-18 again.

- Enter your license key as shown in Figure i-19 and click **Save**.

- Click **Show Properties** to obtain the dialog in Figure i-20. It should indicate the number of concurrent users allowed by the license you have purchased, 2 in this example.

- Click **Close** in Figure i-20 and then **Cancel** in Figure i-19.

- Double-click on **SlsClock.exe** in Figure i-1 to start the Sheriff system clock utility. You should probably add a shortcut to this program under the Startup program folder on the license server so that it always runs. If the clock utility is not running on the license server, Sheriff will use the system time from the last computer that accessed the license database. If system times on the network computers are off by more than a few minutes, which is often the case, the license might be dropped on the computer with the "slowest" system clock.

- You may also wish to create a Start Menu shortcut to **SlsAdmin.exe** on the license server. This can be used to monitor the licenses in use, and also to "drop" users if someone else needs to use SuperHarm right away.

- Install and test SuperHarm on a client workstation, as described in the next subsection.

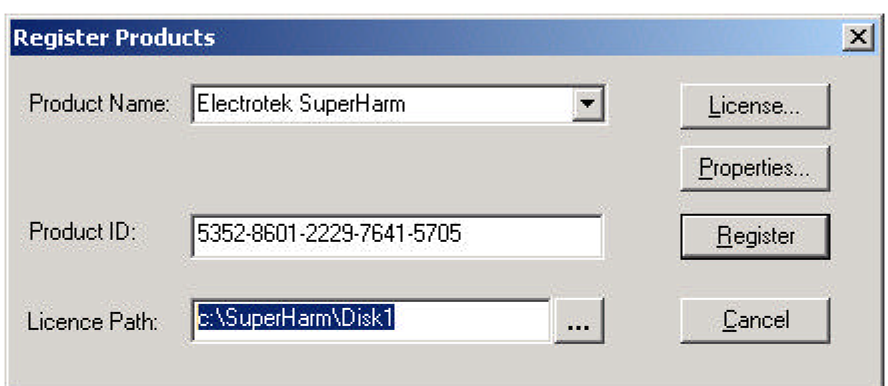


Figure i- 17: Entering a Product ID and License Path on the License Server

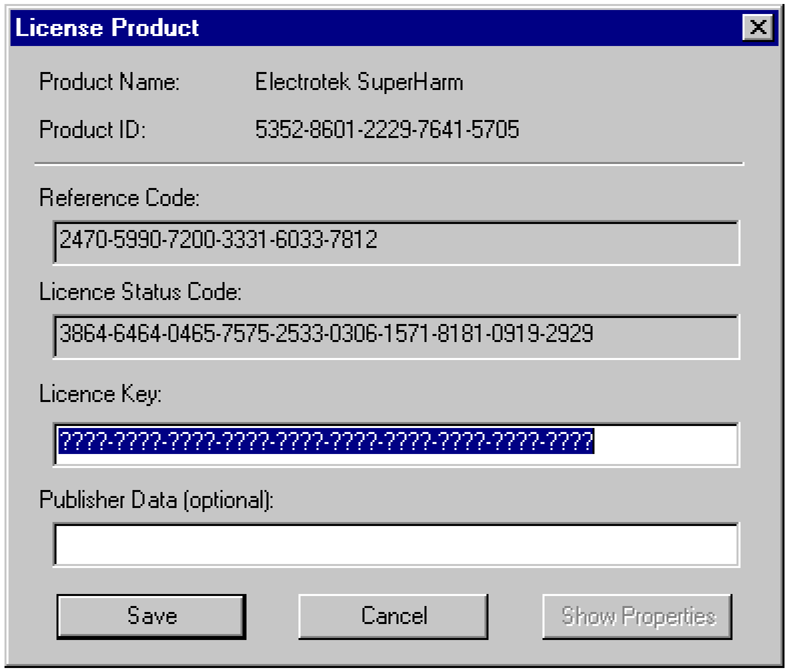


Figure i- 18:Obtaining the Reference Code for the License Server

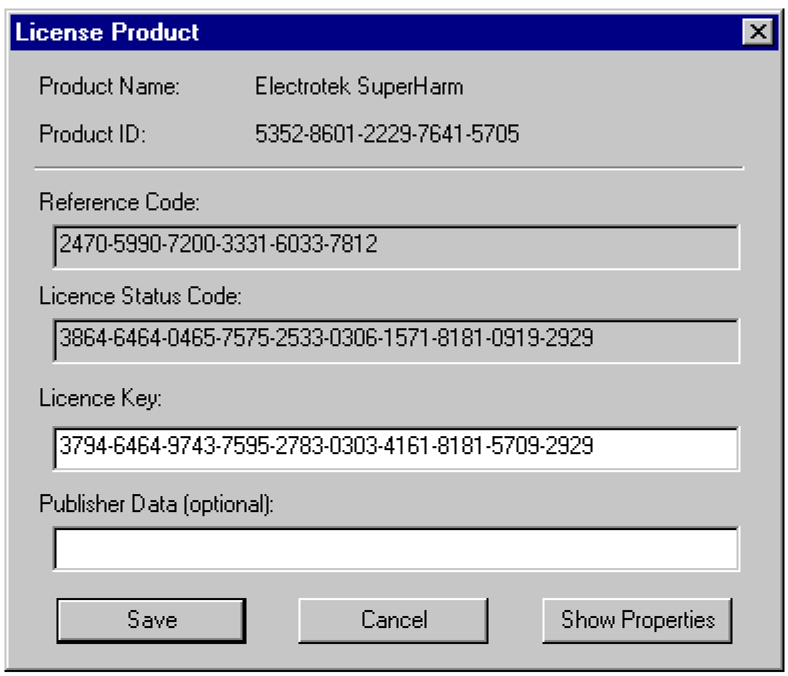


Figure i- 19: : Entering the License Key on the License Server

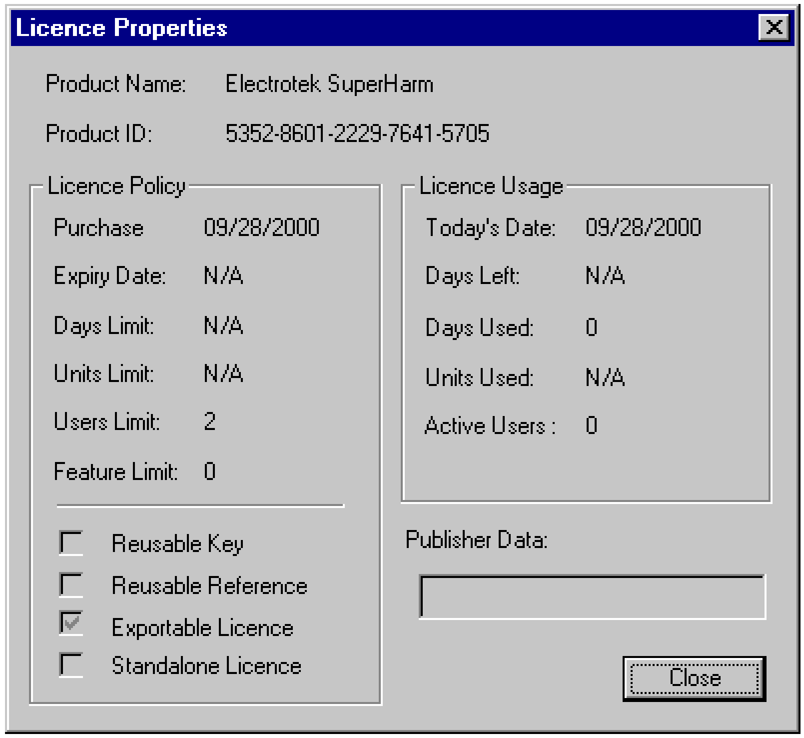


Figure i- 20: Reviewing License Properties on the License Server

The License Admin program on the license server can be used to monitor the SuperHarm licenses in use. To obtain the display shown in Figure i-21, use the **Licence, Monitor** menu command. In this example, both of the **l**-censes are checked out. If you want to "drop" an active user to let someone else run, click the red **X** button on the toolbar in Figure i-21.

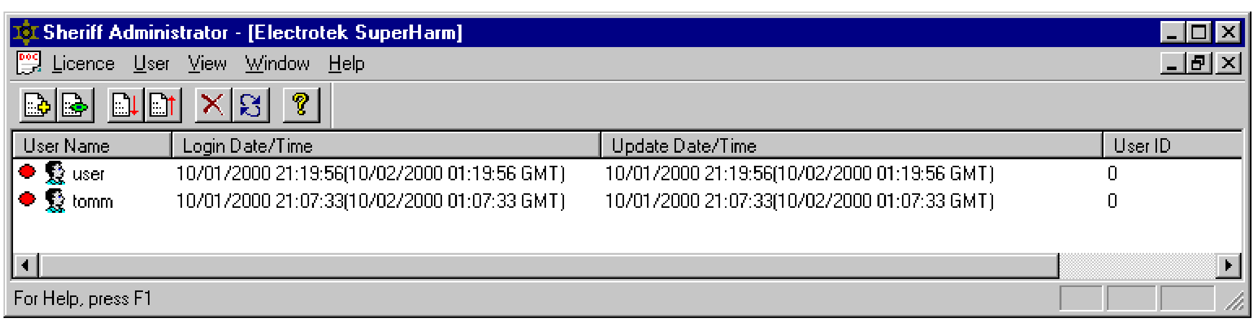


Figure i- 21: Monitoring Checked-out Licenses on the License Server

Network Client Workstation

To use the network licenses from a client workstation, first install SuperHarm from the license server by double-clicking on **setup.exe** in Figure i-1. After this installation, the 14-day trial license can be used on the client workstation without accessing the license server.

To begin using the network license from the client workstation:

- Make sure SuperHarm has been run once on the client workstation. This will create the trial license and save you from having to enter the Product ID on the client workstation.

    - Start the **License Admin** program from the Simulation Tools program folder.

    - Use the **Licence, Register** menu command to obtain the dialog in Figure i-22. Select "Electrotek SuperHarm" from the combo box, and the Product ID should fill in automatically. If not, you will have to enter them manually to match Figure i-22.

    - The Licence Path edit field probably shows the directory where SuperHarm was installed, which contains the 14-day trial license. Replace this with a UNC or mapped drive letter path to the license files, \\SuperHarm\Disk1 in Figure i-17.

    - Click **Register**, you should see the message in Figure i-23. Note that you do not click the License… button or enter a license key on the client workstation.

    - Click **Properties** to obtain the dialog in Figure i-24. This should show the number of concurrent users allowed by the license key you have purchased, 2 in this example.

    - Click **Close** in Figure i-24 and then **Cancel** in Figure i-22.

    - Exit the License Admin program. From a client workstation, it does not monitor license usage over the network, nor can it be used to export a network license. These functions must now be performed on the license server.

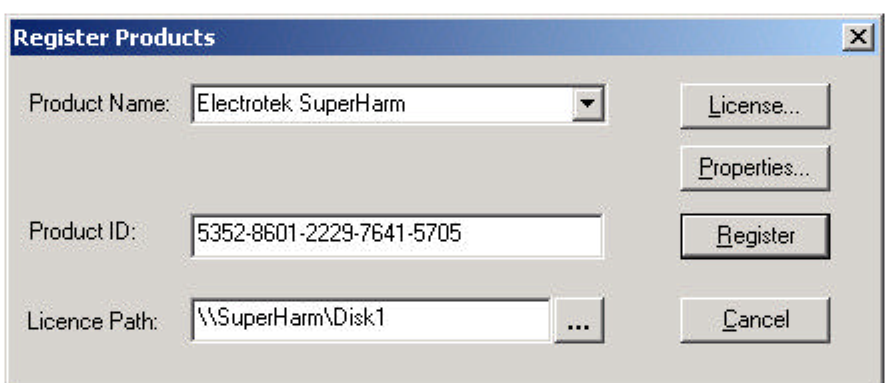


Figure i- 22: Entering the License Path on the Client Workstation

Figure i- 23: Successful Registration on the Client Workstation

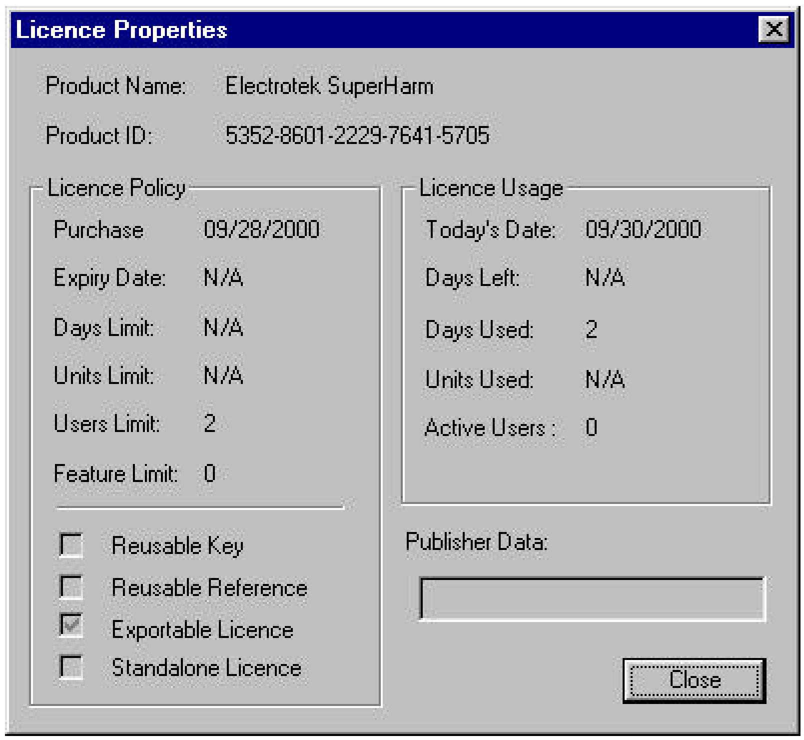


Figure i- 24: Reviewing License Properties on the Client Workstation

When you start SuperHarm with a network license, the about box will appear as in Figure i-25 to indicate the number of network licenses in use. If all the licenses are in use when you start SuperHarm, the message box in Figure i-26 will appear, and then the about box will appear as in Figure i-27. You might contact the users listed in Figure i-26 to see when they will be finished.

Figure i- 25:About Box with a Checked-out Network License

Figure i- 26: Message Dialog Indicating No More Licenses Available.

Figure i- 27: About Box with No Available Network Licenses

The SuperHarm Interface

Figure i-28 shows the program window with a simple data file loaded. This figure shows the menu, toolbar, and three main kinds of windows you will use. There is a status window to show solution progress, a sample voltage output table for the fundamental frequency solution, and a text editor window to make changes.

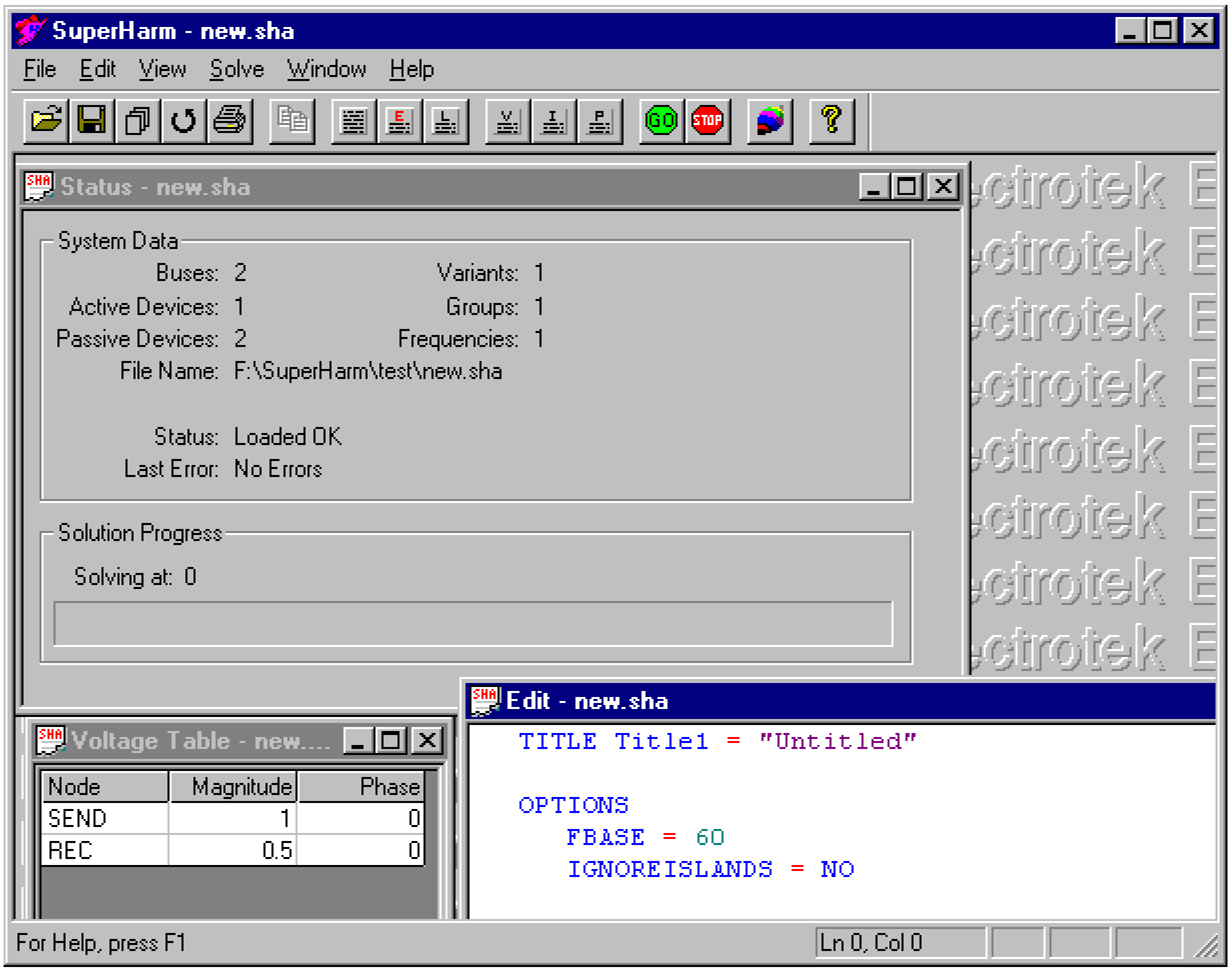


Figure i- 28: SuperHarm Window

SuperHarm user interface preferences include display of the tool and status bars. The tool bar items may be toggled on/off by selecting the **View** / **Tool Bar** and **View** / **Status Bar** menu options respectively. The tool bar (or ribbon), illustrated in Figure i-29, includes a number of command buttons that simplify the selection of common menu commands. Viewing from left to right, equivalent menu commands include:

(1) File, Open	**(2) File, Save**	**(3) File, Batch Solve**
(4) File, Reload	**(5) File, Print**	**(6) Edit, Copy**
(7) View, Data File	**(8) View Error File**	**(9) View, Parser File**
(10) View Voltage Table	**(11) View Current Table**	**(12) View Power Flow**
(13) Solve, Start Solution	**(14) Solve, Interrupt Solution**	**(15) TOP (execute)**
(16) Help, Help Topics		

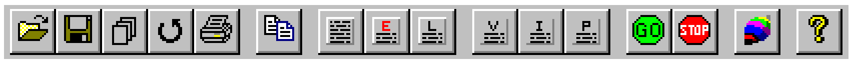


Figure i- 29: SuperHarm Command Ribbon

Linkage with TOP

SuperHarm uses TOP to display plotted and tabular outputs, including:

- Harmonic voltage and current spectra plots
- Frequency scan plots
- Tabulated distortion, TIF, and IT results
- IEEE 519 current evaluations

The TOP toolbar button shown Figure i-29 will automatically launch TOP on the active SuperHarm file. To enable this feature, you need to install TOP, and then ***run it one time from the Windows Start Menu***. This first execution will register TOP, so that SuperHarm will always be able to find it. You can install TOP from the SuperHarm CD, or from the URL *www.pqsoft.com/top*

How to use this Manual

This manual is divided into the following parts:

Part I: Using SuperHarm

Getting Started: This section contains a brief introduction and tutorial for SuperHarm, plus information that you need to install, start and configure the program.

1. ***The Simulation Process:*** This chapter discusses the procedures for creating, solving and editing circuit data files.

2. ***Data File Editor:*** This chapter describes how to use the built-in text editor for input data files.

Part II: Circuit Description Language Reference

3. ***An Overview of the CDL:*** This chapter describes SuperHarm file structure and organization.

4. ***Devices:*** This chapter contains SuperHarm device model descriptions. The models are listed alphabetically.

5. ***Directives:*** This chapter describes the various commands that can be placed in SuperHarm data files, and is also organized alphabetically.

Part III: Reference

A. ***Reference List:*** This appendix provides a chronological listing of pertinent technical papers related to harmonic modeling and analysis.

B. ***Example Data Files:*** This appendix provides a listing of the example data files distributed with the SuperHarm installation.

Getting Help

The SuperHarm help file may be accessed by selecting the **Help**, **Help Topics** menu option. Technical support may be obtained by calling Electrotek at

(800) 554-4767 or (865) 470-9222. Application support is provided by the PQSoft.

World Wide Web

You can obtain additional information about Electrotek and its products and services from the World Wide Web at *http://www.electrotek.com/*. The PQSoft website may be accessed at *http://www.pqsoft.com/*.

Training

Electrotek offers comprehensive in-house and offsite training for SuperHarm. This training provides an informal atmosphere, allowing attendees to progress at their own speed. The training sessions consist of two, three, or five days of case study analysis and practical applications. These sessions are based on existing course material but may be customized to meet the customer's requirements.

CHAPTER 1

THE SIMULATION PROCESS

SuperHarm is essentially a compiler - it translates one type of file into another. The program reads a data file that describes the circuit to be simulated, runs the simulation, and writes the results to an output file. If it can not complete the simulation due to data errors, it produces a file listing where and why errors occurred. This chapter provides the background necessary to create and execute a data file, while successfully utilizing the available features for debugging and verification purposes.

In This Chapter

- Simulation Process
- Creating a New File
- Opening a Data File
- Viewing an Error File
- Viewing the Parsing Log
- Editing Current Data File
- Reloading the Current Data File
- Solving the Current Data File
- Solving Several Data Files At Once
- Viewing the Results
- Printing
- Closing the Current Data File

Page Intentionally Left Blank

Simulation Process

The process for completing a harmonic simulation consists of first collecting and developing the necessary data to represent the circuit to be modeled. Often this system representation is completed by “describing” the interconnection and component values in a simple ASCII text file. For example, the following SuperHarm datafile excerpt represents a 1500kVA, three-phase step-down transformer.

```
//   Step down transformer #1 (@ service entrance)
//   1500 kVA
//   12.5kV / 480 Volt, (connection - delta / wye-gnd)
//   Z = 6% @ 1.5 MVA, X/R = 10
//   Ie = 1% @ 100% V

TRANSFORMER   NAME = STEP1   H = DELTA   X = WYE
   MVA = 1.50     %IMAG = 1.0    KV.H = 12.5     KV.X = 0.480
   H.A = PCC_A    H.B = PCC_B    H.C = PCC_C
   X.A = 4801A    X.B = 4801B    X.C = 4801C     X.N = GROUND
   MVAB.HX = 1.5  %R.HX = 0.6    %X.HX = 6.0
```

After the data file has been created, it is submitted to the harmonic solution engine (solver). The solver reads the data file, line-by-line, and reports any significant errors. Satisfied that the case will solve, the solver generates a matrix representation of the interconnected system. In general, there are two types of harmonic simulations:

1. ***Frequency Scans:*** The frequency scan is the simplest and most commonly used technique for harmonic analysis. A scan calculates the frequency response characteristic at a particular bus or node. Usually, this is accomplished by injecting one amp into the bus over a range of frequencies and then observing the resultant voltage. The resultant voltage is directly related to the system impedance in ohms. Frequency scan analysis is the best method for identifying resonance conditions. It has also been used a great deal in filter design.

2. ***Distortion Simulations:*** Harmonic distortion simulations use harmonic source characteristics of nonlinear loads to determine current and voltage distortion levels at various points in the system. Harmonic source characteristics (current sources) are obtained from field measurements, other simulation programs (e.g. Electromagnetic Transients Program - EMTP), or a library of typical waveforms. Distortion simulations are useful for evaluating component duty and determining harmonic limit compliance (e.g. IEEE 519-1992).

SuperHarm is one of three programs required to conduct a harmonic simulation. You must use a text editor program to prepare the data file, and TOP, The Output Processor to view the results. You can start either of these applications directly from SuperHarm. Figure 1-1 illustrates the simulation process including the SuperHarm component. The general process of harmonic simulation includes the following steps:

1. Load a *current data file* using either of the following two methods. SuperHarm immediately checks the file for errors. If it finds any, it creates a file listing each offending data line and why the data is incorrect.
   - Use the **File, Open** command to select an existing file, or
   - Use the **File, New** command to create a small skeleton file.
2. Use the **View, Data File** command to make modification and/or corrections to the data file using the editor. When finished, select the **File, Reload** command to repeat the error check.
3. Use the **View, Parser File** command to view a listing of the data after it has been parsed by SuperHarm.
4. Use the **View (Current Table, Voltage Table, Power Flow)** commands to display the output for the fundamental frequency, steady-state solution.
5. Use the **Solve, Start Solution** command to complete the harmonic solution and generate the output file.
6. Switch to TOP, The Output Processor to view the results of the simulation

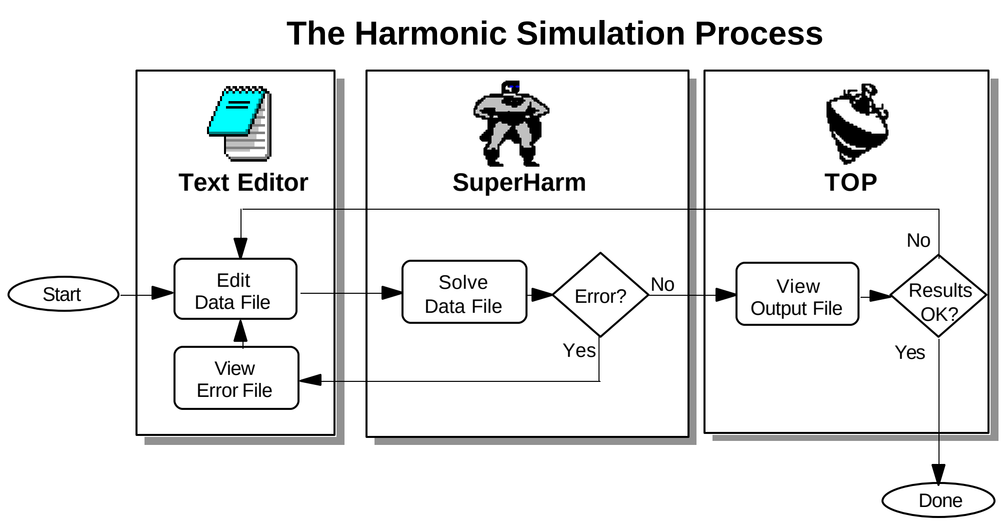


Figure 1-1: The Harmonic Simulation Process

Creating a New Data File

From the **File** menu, choose **New** to display the **Save As** dialog box shown in Figure 1-2. Enter a file name and click **Save** to write a small skeleton SuperHarm input file. Use the **Save as type** combo box to create a library or include file instead. You will be prompted before overwriting an existing file.

Figure 1-3 shows the resulting skeleton file after using the **View, Data File** menu command; is perhaps the simplest file that SuperHarm can successfully load. Make changes in this file to create your case input, save the file (using **File, Save** or **Ctrl-S**), reload it and solve.

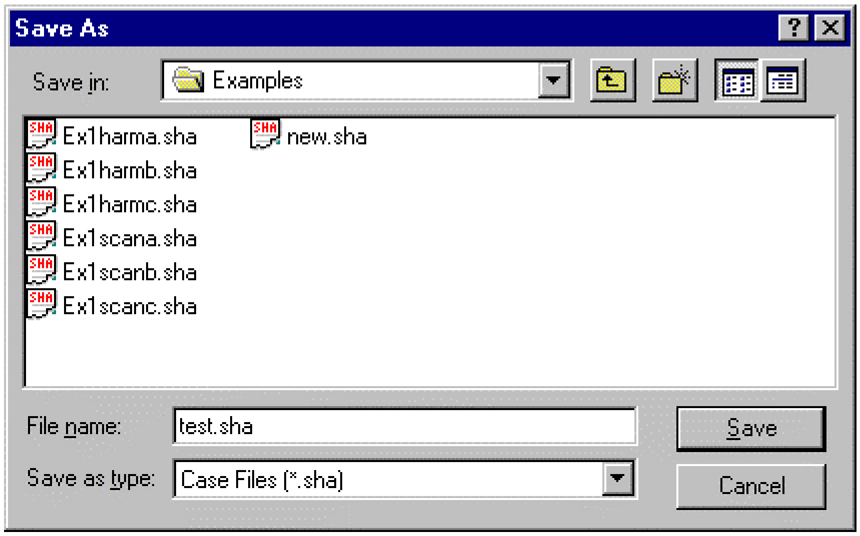


Figure 1-2: Save As dialog to create a new input file

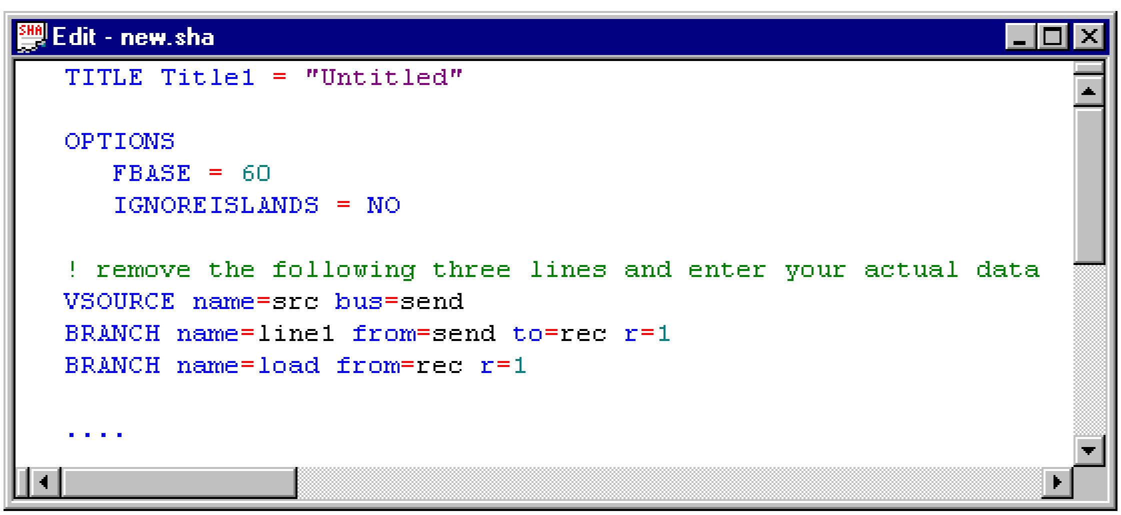


Figure 1-3: Skeleton input created by File, New menu command

Opening a Data File

From the **File** menu, choose **Open** [ribbon button: ] to display the **Open** dialog box shown in Figure 1-4. Select "SuperHarm Case Files (*.sha)" in the **Files of Type** combo box. Select the data file you want to open, and click **Open**. Note that you may shorten the operation by simply double-clicking on the desired file. You can also use the **Files of Type** combo box to open library or include files.

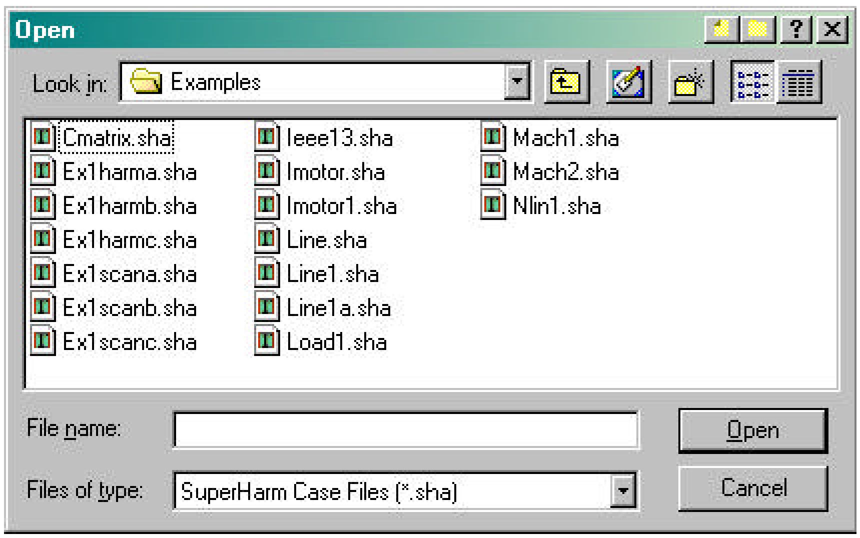


Figure 1-4: Open Dialog Box

Once a file has been opened, it becomes the current data file - the target of most of the operations described in the following pages. The name of the current data file, if any, is displayed in the title bar at the top of the SuperHarm main window.

TIP: Utilize the Windows "drag and drop" capability for opening a SuperHarm data file. Using Windows Explorer (or equivalent), simply point to a data file (*.sha), press and hold down the mouse button, and drag the file to the SuperHarm window.

SuperHarm displays a status window (refer to Figure 1-5) for each open data file. Information contained in the window includes:

- buses (note: a three-phase bus is counted as three buses)
- active devices (ISOURCE, VSOURCE, NONLINEARLOAD)
- passive devices

- variants
- source groups
- solution frequencies
- filename and path
- status indicator
- last error indicator
- solution progress

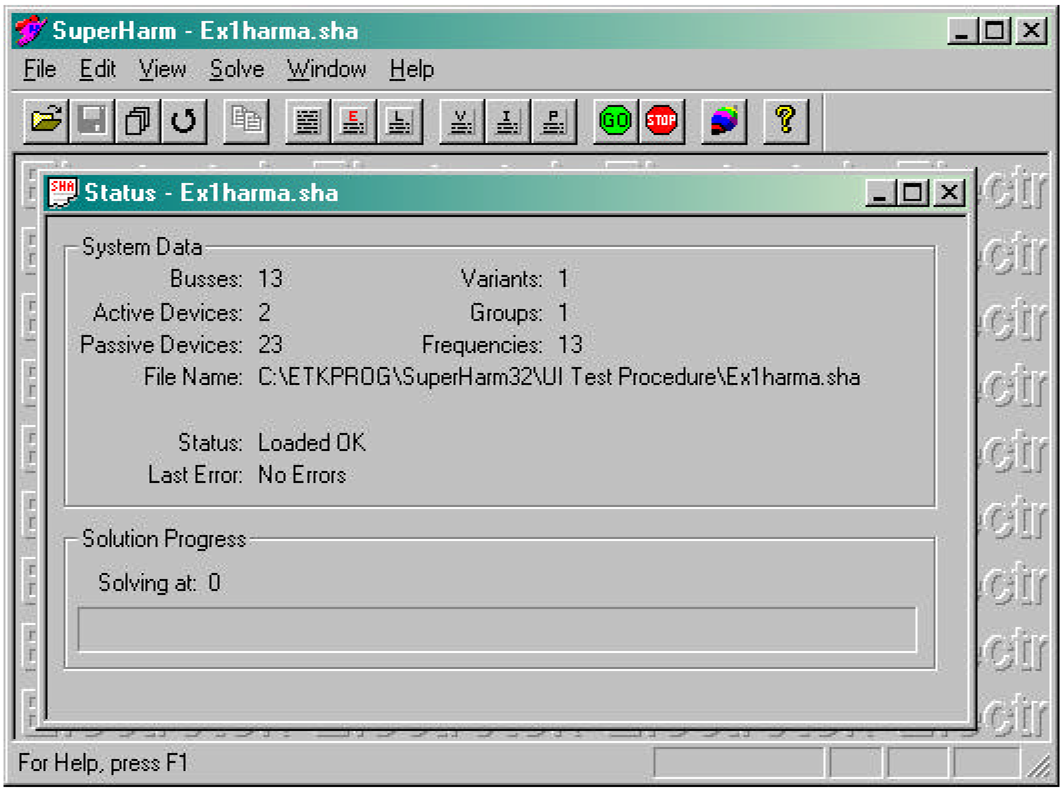


Figure 1-5: SuperHarm Program Window

Viewing an Error File

When SuperHarm opens a data file it immediately checks the file for errors. If it finds any, it creates a file listing each offending data line and why the data is incorrect. An error file has the same root name as the corresponding data (SHA) file, but uses a SHE (SuperHarm Error) extension. SuperHarm places the error file in the same directory as the data file.

The error file can be viewed by using the **View, Error File** [ribbon button:] command.

Viewing the Parser Log

The parser log is a text file created by SuperHarm after loading a data file. This log indicates line-by-line how each line was interpreted by the program. This output log can be an excellent tool when debugging a data file. A parser output file has the same root name as the corresponding data (SHA) file, but uses a SHP (SuperHarm Parser) extension. SuperHarm places the parser log in the same directory as the data file.

The log can be viewed by using the **View, Parser File** [ribbon button:] command.

Editing the Current Data File

From the **View** menu, choose **Data File** [ribbon button:] to display the current data file as shown in Figure 1-6. Be sure to use **File, Save** or **Ctrl-S**, and then **File, Reload** to include any changes in the next solution.

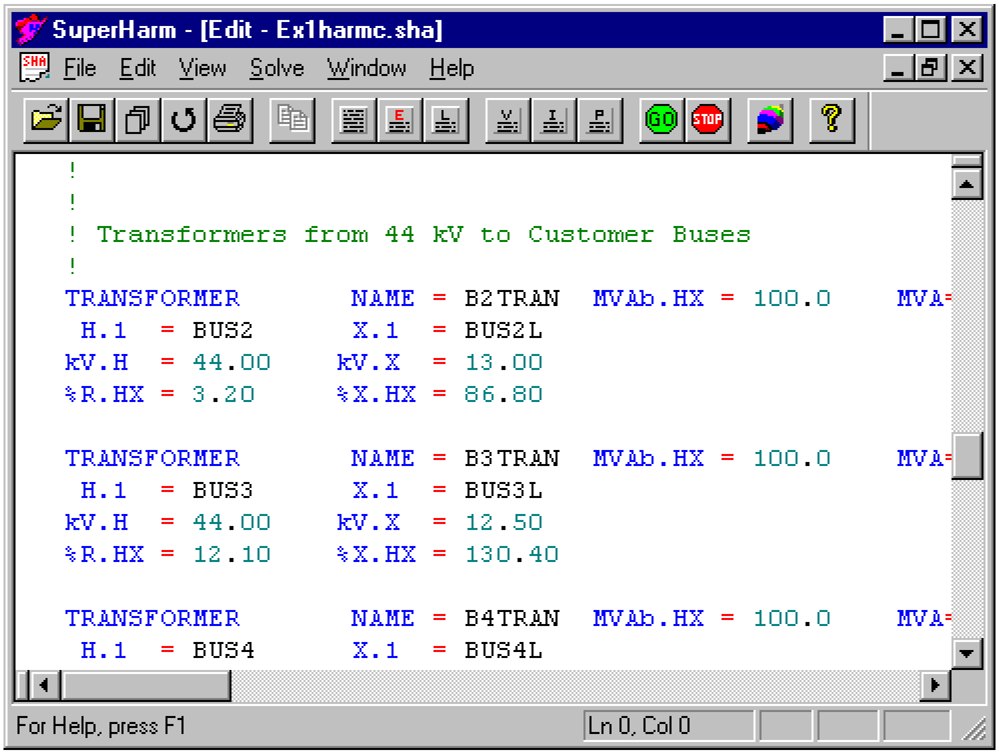


Figure 1-6: Edit Data File via Notepad

Reloading the Current Data File

This step is required if changes have been made to the current data file since it was initially selected with the **File, Open** command.

From the **File** menu, choose **Reload** [ribbon button: ] to reload the current data file.

Solving the Current Data File

If no errors are reported after you use the **File, Open** or **File, Reload** commands, you are ready to solve the data file. SuperHarm displays the status window as in Figure 1-7.

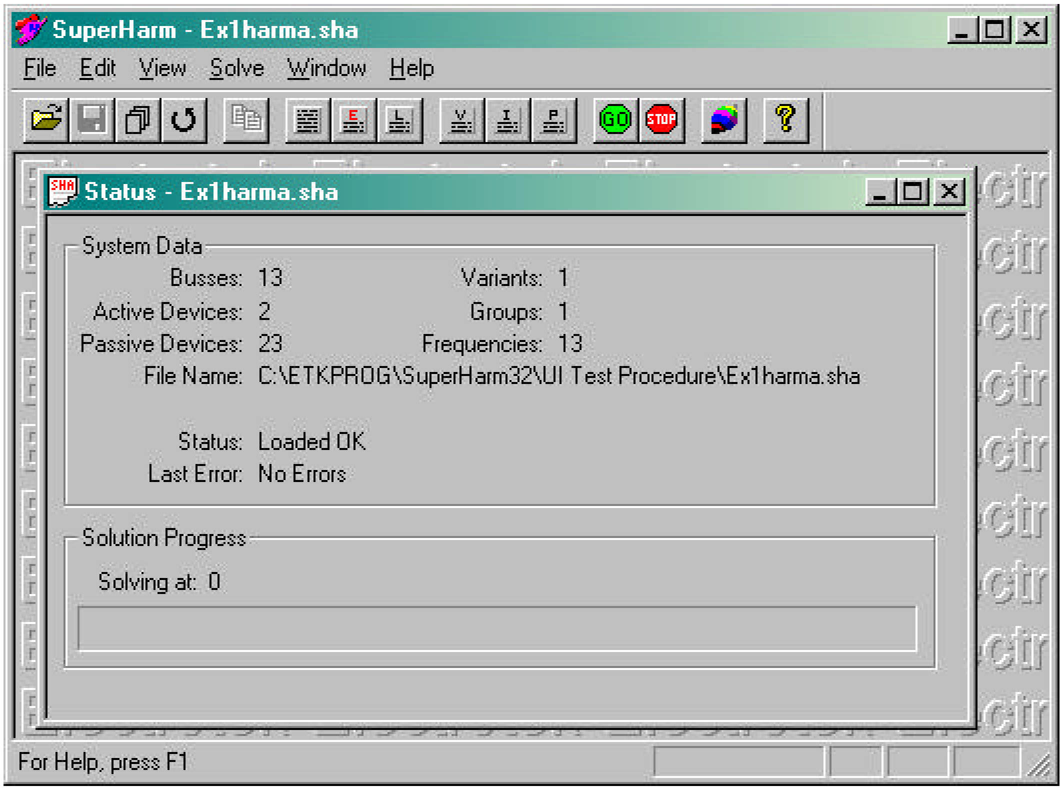


Figure 1-7: SuperHarm Window prior to Solution

The **System Data** portion of the window lists the number of nodes, devices, frequencies and other information concerning the current data file.

From the **Solve** menu, choose **Start Solution** [ribbon button: ] to begin the solution process. The **Solution Progress** portion of the window provides a visual indication of the solution status (refer to Figure 1-8).

To abort the solution before its completion, choose **Interrupt Solution** from the **Solve** menu [ribbon button: ].

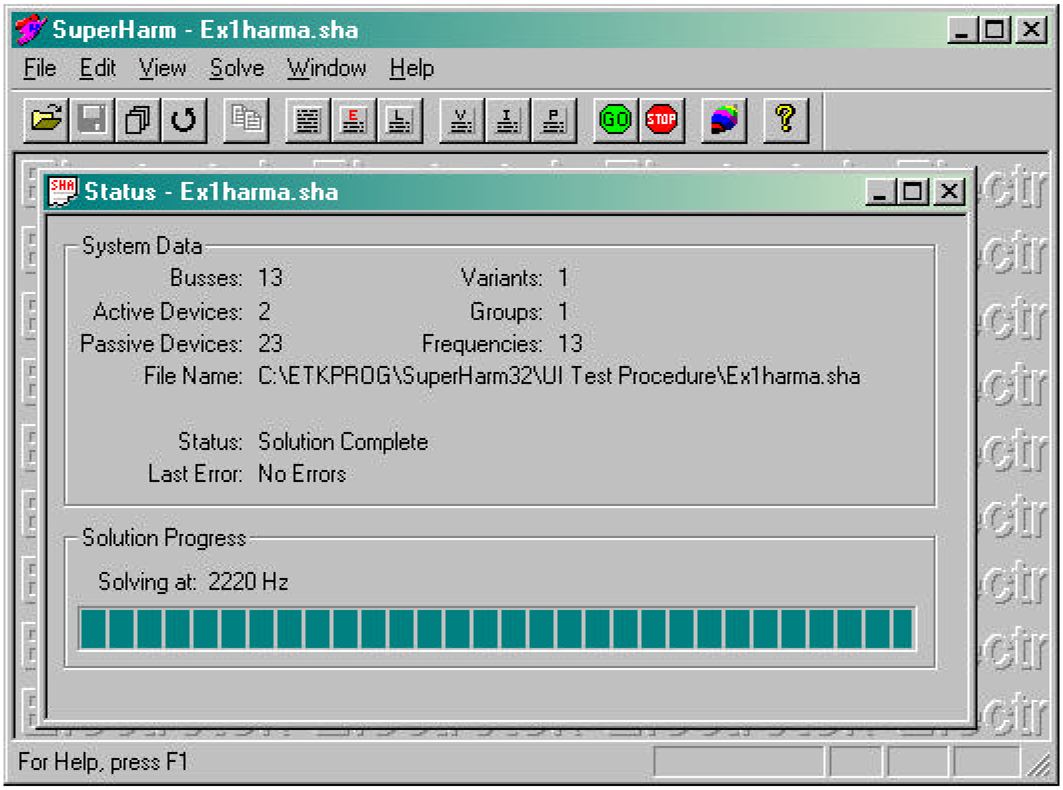


Figure 1-8: SuperHarm Window after Solution

Solving Several Data Files at Once

SuperHarm's batch solution capability allows multiple data cases to be run consecutively or the same data case to be repeated a specified number of times.

TIP: The user should review the guidelines for the **`BATCH_ID`** and **`OPTIONS OUTFILE = filename`** directives before using the batch solution function.

From the **File** menu, choose **Batch Solve** [ribbon button:] to display the Batch Solve File Select dialog box shown in Figure 1-9.

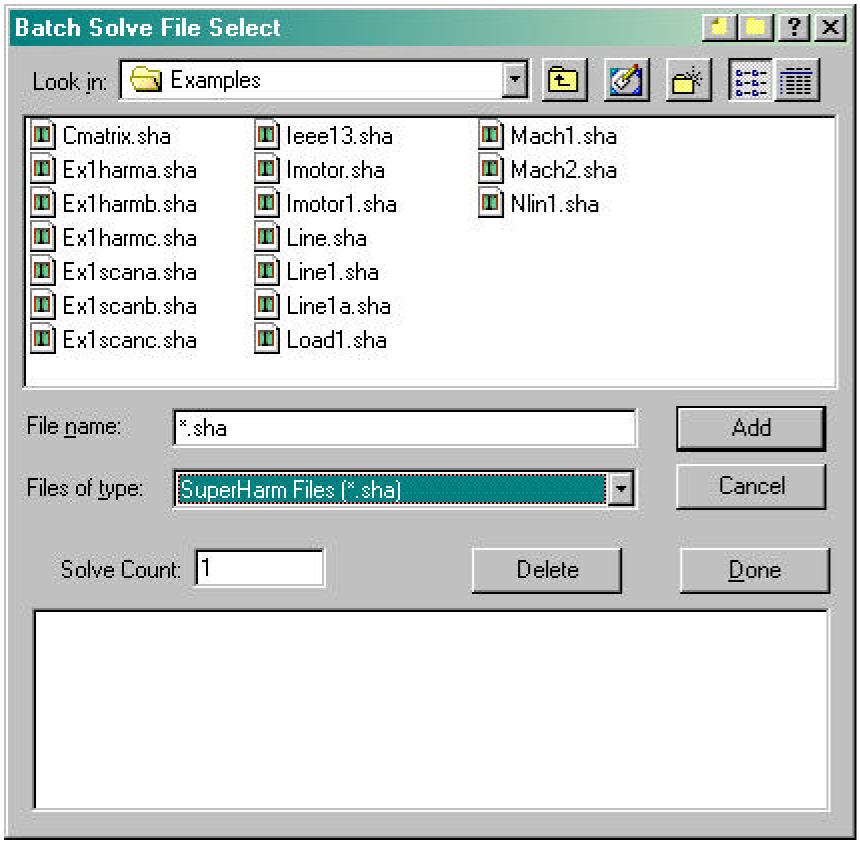


Figure 1-9: SuperHarm Batch Solve File Select Dialog Box

Switch to the directory containing the case(s) to be solved. Select the data file you want to add to the batch solution list, and click **Add**. Note that you may shorten the operation by simply double-clicking on the desired file. You can also select multiple files with **shift-click** and **control-click**, and then **Add**. Figure 1-10 illustrates the resulting batch solution list.

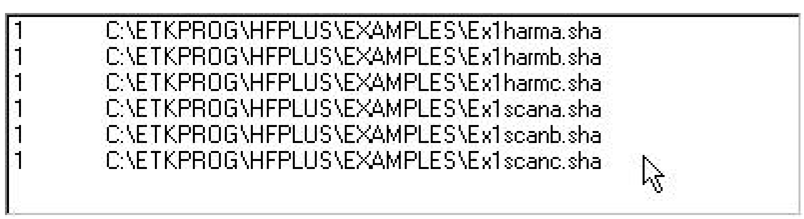


Figure 1-10: Batch Solution List

To remove a case from the list, select the desired file from the batch solution list (single-click) and click **Delete**.

To repeat the same case "Solve Count" times, enter the desired number in the Solve Count field, select the file to run, and click **Add**. Figure 1-11 illustrates the resulting batch solution list. Note that you may shorten the operation by simply double-clicking on the desired file. For the batch solution list shown in Figure 1-9, cases Ieee13.sha and Imotor.sha will be repeated five times each, while the other cases will run once each.

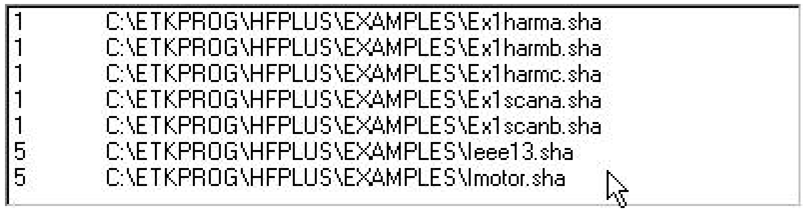


Figure 1-11: Batch Solution List Illustrating Solve Count

Click **Done** when finished with the batch solution list setup or **Cancel** to abort the operation. A batch solution status window similar to Figure 1-12 will be displayed. The window provides the following summary information:

1. The Status column provides the status of the case. Possible values include "Unsolved" and "Solution Complete" and "Solution Finished with Error".
2. The Total column provides the "Solve Count" specified for each case, while the Batch ID column indicates the current `BATCH_ID`.
3. The Last Error column provides the error condition for the case. Possible values include "No Errors" indicating a successful case and "Errors in Data File – Check Error File" indicating that an error was found.
4. The File column provides the data file name.

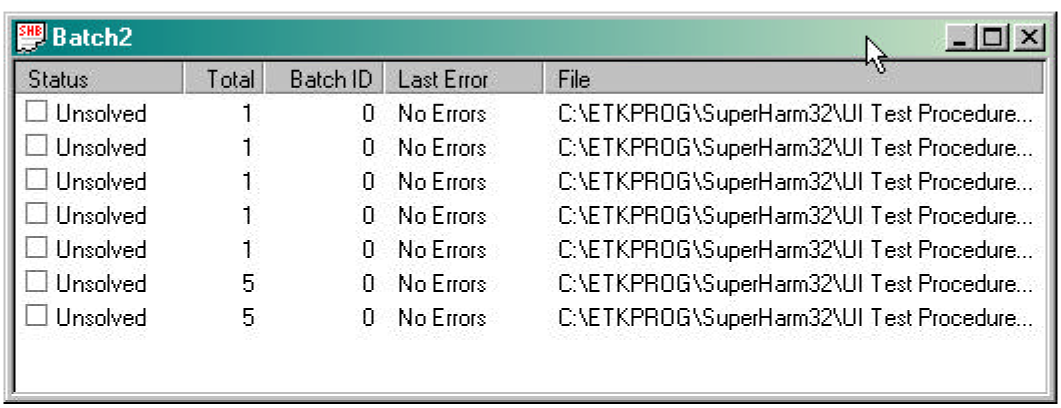
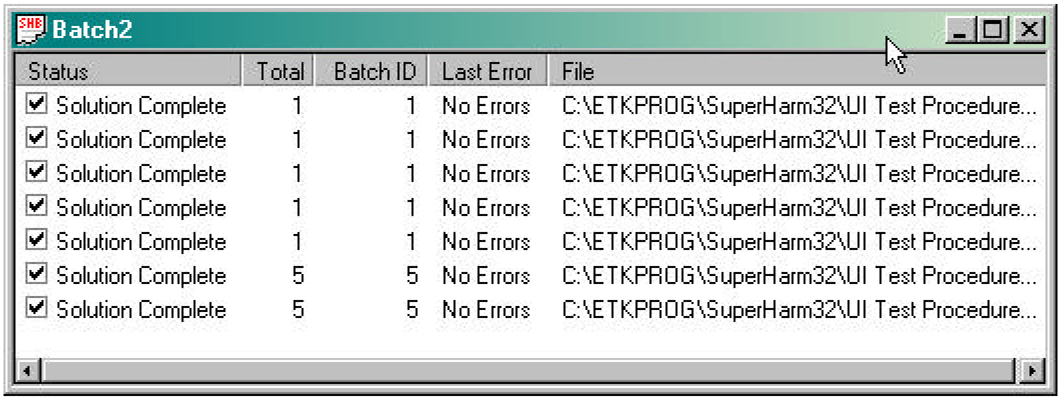


Figure 1-12: Batch Solution Status Window – Before Solution

From the **Solve** menu, choose **Start Solution** [ribbon button: ] to begin the batch solution process. The Status column provides a visual indication of the solution status (Figure 1-13). To abort the batch solution before its completion, choose **Interrupt Solution** from the **Solve** menu [ribbon button:]



Status	Total	Batch ID	Last Error	File
Solution Complete	1	1	No Errors	C:\ETKPROG\SuperHarm32\UI Test Procedure...
Solution Complete	1	1	No Errors	C:\ETKPROG\SuperHarm32\UI Test Procedure...
Solution Complete	1	1	No Errors	C:\ETKPROG\SuperHarm32\UI Test Procedure...
Solution Complete	1	1	No Errors	C:\ETKPROG\SuperHarm32\UI Test Procedure...
Solution Complete	1	1	No Errors	C:\ETKPROG\SuperHarm32\UI Test Procedure...
Solution Complete	5	5	No Errors	C:\ETKPROG\SuperHarm32\UI Test Procedure...
Solution Complete	5	5	No Errors	C:\ETKPROG\SuperHarm32\UI Test Procedure...

Figure 1-13: Batch Solution Status Window – After Solution

TIP: Quickly view a file by double-clicking on the respective entry in the Status column. The data file will be opened using Notepad if the case completed successfully or the error file will be loaded if the case ended with an error. Selecting a file and clicking on the TOP button will start TOP, load the respective output file, and display the stack load dialog.

Viewing the Simulation Results

After SuperHarm successfully opens a data file and checks it for errors, a fundamental frequency solution is completed. Tabular results for this analysis may be viewed from within SuperHarm. Note that this solution is not completed for Scan-Mode cases (files including the **SCAN** directive).

The voltage table can be viewed by using the **View, Voltage Table** [ribbon button:] command.

The current table can be viewed by using the **View, Current Table** [ribbon button:] command.

The power flow table can be viewed by using the **View, Power Flow** [ribbon button:] command.
Figures 1-14 and 1-15 illustrate example voltage, current, and power flow tables for a fundamental frequency steady-state solution.

Ex1harma.sha:2

Node	Magnitude	Phase
SRCV	25400	0
RECT	2022.95	-3.46898
bus7l	2316.29	-5.24686
44kvbus	25310.3	-2.70612
bus2l	7411.5	-4.82109
bus3l	7429.85	-3.75699
bus4l	7173.42	-4.55119
bus1	25296.3	-2.87345
bus6	24796.5	-3.74765
BUS2	25143.4	-3.368
BUS3	25388.7	-2.86406
BUS4	25194.1	-2.93501
BUS7	24739.4	-3.87444

Ex1harma.sha:3

Name	Magnitude	Phase
rectif.bus7l.0	300.939	-17.2844
rectif.RECT.1	300.939	162.716
equiv.44kvbus.0	194.859	-162.107
equiv.SRCV.1	194.859	17.893
bank01.44kvbus.0	71.7684	87.2939
bank01.GROUND.1	71.7684	-92.7061
bank02.bus2l.0	40.8637	85.1789
bank02.GROUND.1	40.8637	-94.8211
bank03.bus3l.0	70.2254	86.243
bank03.GROUND.1	70.2254	-93.757
bank04.bus3l.0	70.2254	86.243
bank04.GROUND.1	70.2254	-93.757
bank05.bus4l.0	28.2507	85.4488
bank05.GROUND.1	28.2507	-94.5512
LOAD1.bus1.0	58.7983	-10.9096
LOAD1.GROUND.1	58.7983	169.09

Figure 1-14: Tabular Steady-State Solution Output for Fundamental Voltage and Current

Ex1harma.sha:4

Bus	Magnitude	Angle					
	Device	Terminal	I Mag	I Ang	kW	kVAr	kVA
SRCV	25400.0	0.0					
	VSRC	1	0.0	0.0	0.000	0.000	0.0[illegible]
	equiv	2	194.9	17.9	4710.020	-1520.663	4949.41
RECT	2022.9	-3.5					
	SRC1	1	300.9	-17.3	591.171	145.374	608.7[illegible]
	rectif	2	300.9	162.7	-591.171	-145.374	608.7[illegible]
bus7l	2316.3	-5.2					
	rectif	1	300.9	-17.3	681.736	145.374	697.0[illegible]
	LOAD7	1	267.7	-31.1	558.050	270.276	620.0[illegible]
	B7TRAN	3	564.5	156.2	-1239.786	-415.650	1307.6[illegible]
44kvbus	25310.3	-2.7					
	equiv	1	194.9	-162.1	-4616.614	1735.194	4931.9[illegible]
	bank01	1	71.8	87.3	0.000	-1816.480	1816.4[illegible]
	LINE1	1	58.8	-10.9	1472.975	212.351	1488.2[illegible]

Figure 1-15: Tabular Steady-State Solution Output for Power

Tabular data may be copied to the Window's clipboard by using the **Edit, Copy** [ribbon button:] command.

SuperHarm utilizes TOP, The Output Processor to visualize the simulation results. The program takes advantage of the Microsoft Windows Graphical

User Interface and clipboard to allow the user to easily transfer data to other Windows programs such as Microsoft Excel® and Microsoft Word®.

Tabular and graphical output for a solved SuperHarm case may be viewed by clicking on the TOP icon [ribbon button:].

TOP will begin and the Load SuperHarm Data (stack load) dialog for the SuperHarm data format will be displayed for the current output file (illustrated in Figure 1-13).

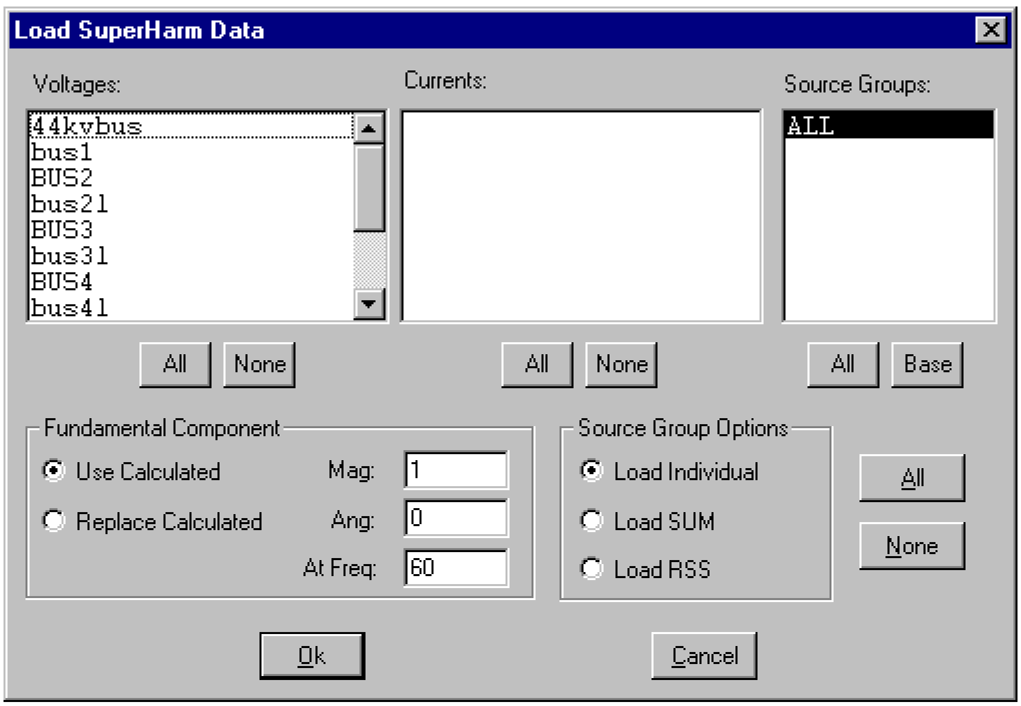


Figure 1-16: TOP – Load SuperHarm Data Dialog

Printing

You can print the input data file or any table by selecting the appropriate window, and then use the **File, Print** menu command, or the printer icon on the toolbar. TOP can also print its plots and tables.

Closing the Current Data File

From the **File** menu, choose **Close** [] to close the current data file.

CHAPTER 2

DATA FILE EDITOR

SuperHarm has a built-in text editor with color highlighting of SuperHarm keywords, numerical values, and comments. After either creating a new skeleton input file, or opening an existing file, you would use the editor to make changes before running each case. This chapter describes the editor features and commands, including keyboard and mouse shortcuts. The next three chapters describe the syntax of the input files you will create and edit.

In This Chapter

- The Editor Window
- The Editor's Feature Set
- Window Properties
- Find and Replace
- Using the Macro Recorder
- Keyboard Shortcuts
- Mouse Shortcuts

Page Intentionally Left Blank

The Editor Window

The SuperHarm editor window is opened with the current data file by using the **View**, **Data File** [ribbon button:] menu command. Figure 2-1 illustrates the default window. You can also use the **File, New** or the **File, Open** menu command to open an editor window on a library or include file. Modified files may be saved using the **File**, **Save** [ribbon button:] menu command, or the **File, Save As** menu command to save a copy under a different name.

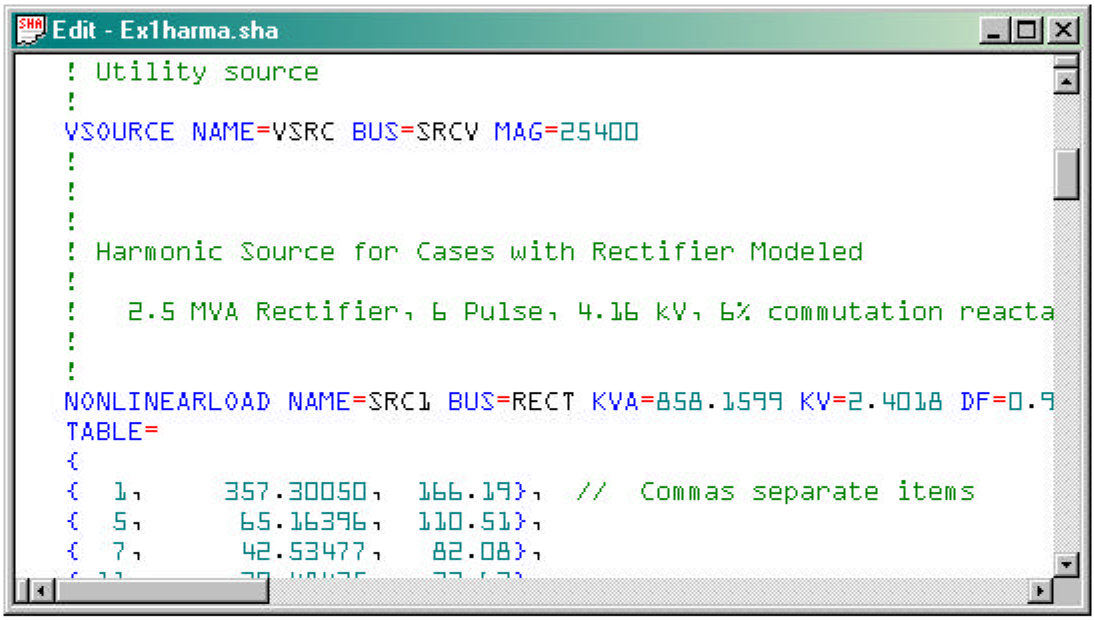


Figure 2-17: The SuperHarm Editor Window

The Editor's Feature Set

The editor supports a wide range of features, all of which are available to the user:

- ***Color Syntax Highlighting***. Language-specific keywords, comments, and strings are colored differently to set them apart from plain text.

- ***Fully Customizable Keyboard Mappings***. Keystrokes can be added and removed to emulate popular keyboard mappings, like Brief and Epsilon.

- ***Keystroke Macros***. Users can record a series of keystrokes and assign a keystroke to play back the keystrokes repeatedly. Up to 10 macros may be recorded.

- ***Multiple Split Views***. Users can create up to four separate views of the same edit buffer. Each view can be scrolled independently (refer to Figure 2-10).

- ***Drag and Drop Text Manipulation***. Highlighted text can be dragged and dropped between any window supporting OLE text Drag and Drop. Text may be copied or moved.

- ***Unlimited Undo/Redo***. All edit actions are fully undoable and redoable. A limit can be set on the number of edit actions that may be undone.

- ***Auto Indentation***. Once a language is chosen, as the user enters code, the editor will automatically indent lines to following the scoping rules of the language.

- ***Clipboard Support***. Text can be cut, copied and pasted onto the clipboard using the right-click menu or a keystroke.

- ***Column Selection and Manipulation***. Columns of text can be selected with the mouse and then manipulated. Empty columns (columns with a width of zero characters) can be selected, causing subsequent typing and deletion to occur over multiple lines at the same time (refer to Figure 2-10).

- ***Built-In Window Properties Dialog***. Most properties, including keyboard assignments, colors, current language, and tab settings are available to the user in a tabbed dialog available from the built-in right-click menu or a keystroke (Alt+Enter). This dialog can be easily disabled and/or overridden by the parent window.

- ***Built-In Right Click Menu***. A set of commonly used commands are available from a default right-click menu.

- ***Microsoft IntelliMouse Support***. Scrolling, word selection, and line selection is easily accomplished with the Microsoft IntelliMouse.

- ***Over 120 Separate Edit Commands***. More than 120 separate edit commands can be assigned to a keystroke and invoked by the user.

Window Properties

Most properties, including keyboard assignments, colors, current language, and tab settings are available to the user in a tabbed dialog available from the **Edit**, **Properties** menu command, the built-in right-click menu, or a key-stroke (Alt+Enter). Figure 2-2 shows the *Color/Font* tab of the Window Properties dialog. This tab includes a user-selectable color option, and a dialog for selecting the default editor font (Figure 2-3). All settings are saved in the Windows Registry and are restored when SuperHarm is closed and started again.

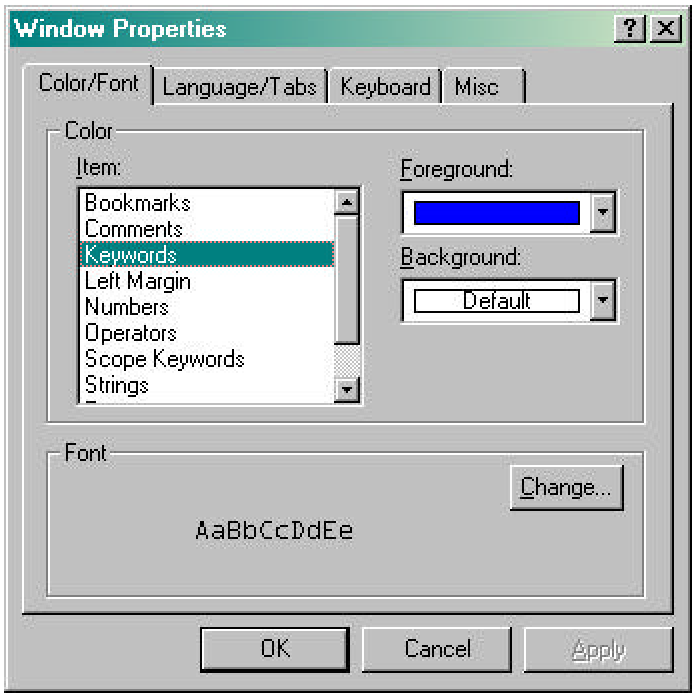


Figure 2-18: The Editor's Window Properties Dialog [Color/Font]

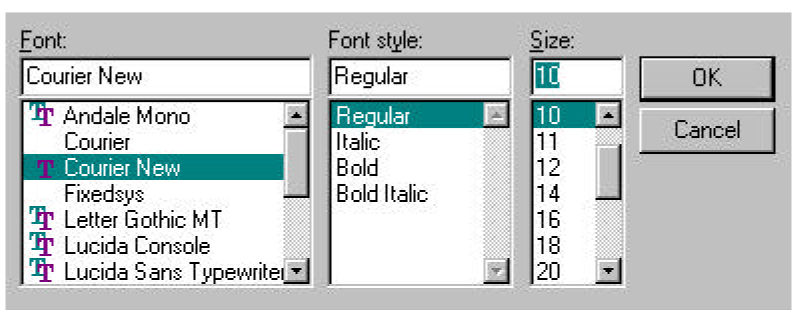


Figure 2-19: The Font Selection Dialog

Figure 2-4 shows the *Language/Tabs* tab of the Window Properties dialog. This tab includes user-selectable options for indentation style, tabs, and language (SuperHarm).

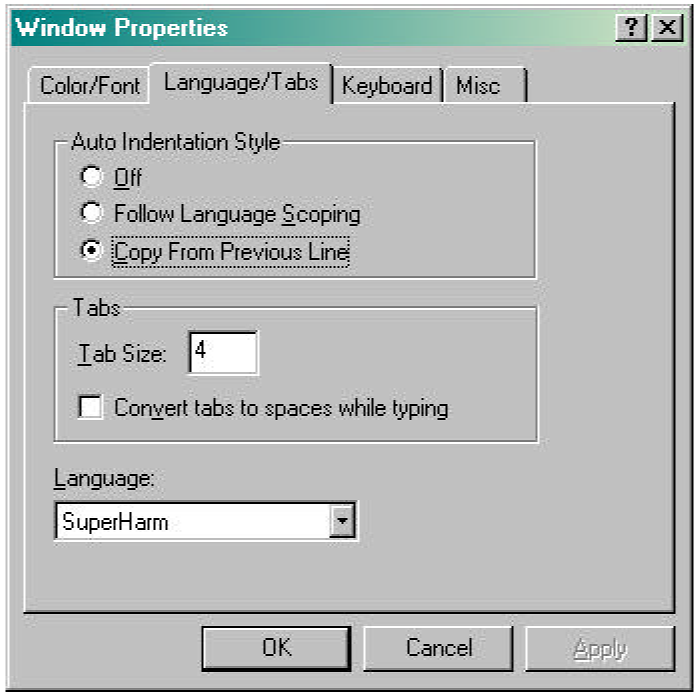


Figure 2-20: The Editor's Window Properties Dialog [Language/Tabs]

Figure 2-5 shows the *Keyboard* tab of the Window Properties dialog. This tab includes user-selectable options for keyboard mapping.

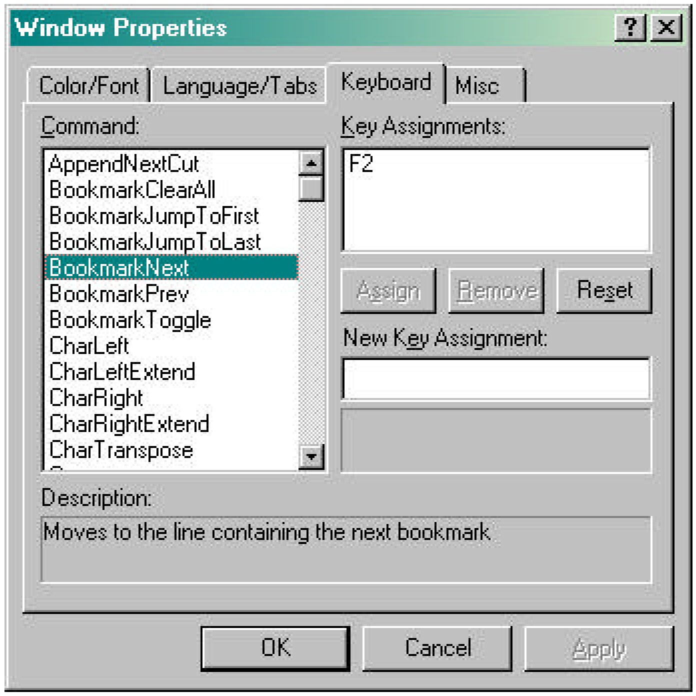


Figure 2-21: The Editor's Window Properties Dialog [Keyboard]

Figure 2-6 shows the *Misc* tab of the Window Properties dialog. This tab includes user-selectable options for a number of visual preferences.

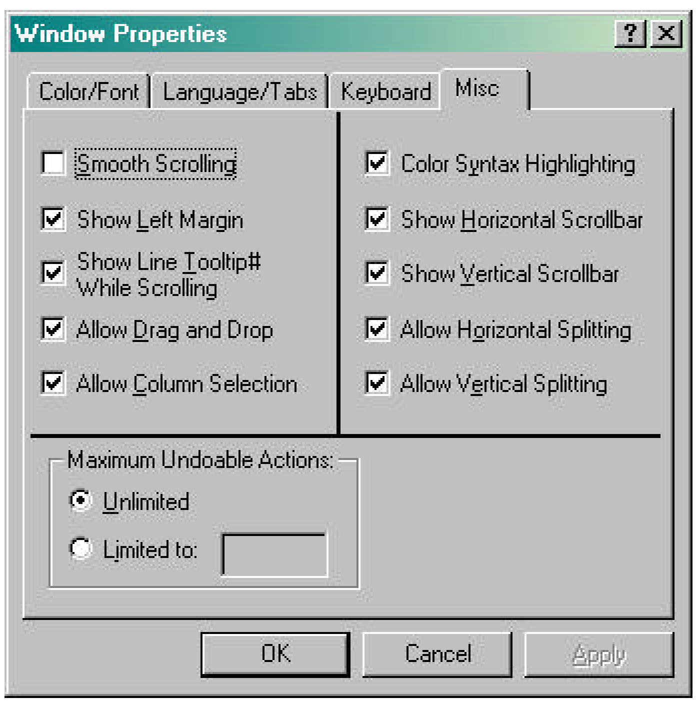


Figure 2-22: The Editor's Window Properties Dialog [Misc]

Find and Replace

The *Find* dialog box may be accessed by using the **Edit**, **Find** menu command or the Alt+F3 keystroke. Figure 2-7 shows the Find dialog.

Figure 2-23: The Editor's Find Dialog

The *Find & Replace* dialog box may be accessed by using the **Edit**, **Replace** menu command or the Ctrl+Alt+F3 keystroke. Figure 2-8 shows the *Find & Replace* dialog.

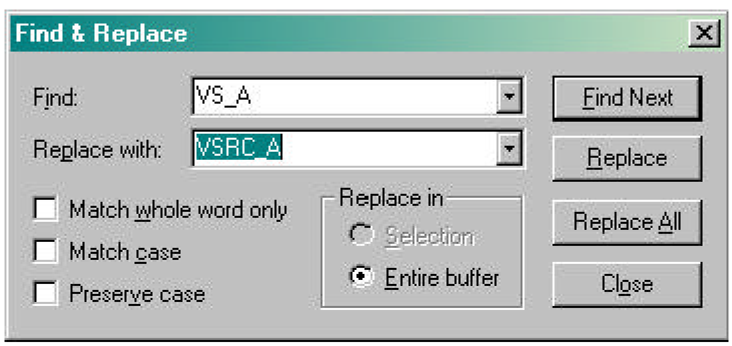


Figure 2-24: The Editor's Find & Replace Dialog

Using the Macro Recorder

Keystroke and menu command sequences (macros) may be recorded and saved for later use. The macro recorded is accessed using the **Edit**, **<MACRO>** menu command. Once activated, keystrokes and menu commands are recorded until the **Stop** button is pressed.

[Stop button for macro recorder]

When the **Stop** button is pressed, the user is presented with *the Save Macro* dialog. A new shortcut key combination is then assigned to one of the 10 macro locations.

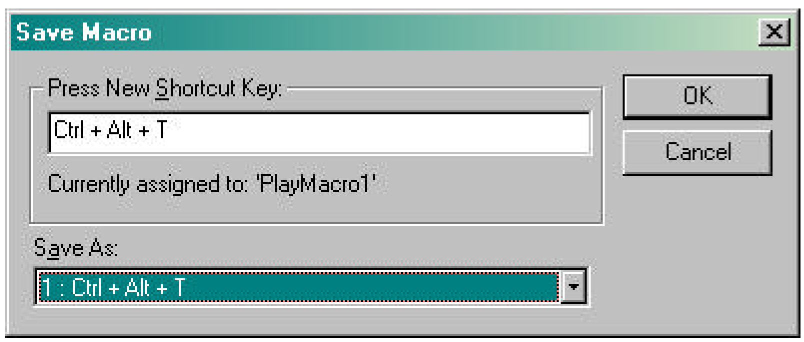


Figure 2-25: The Save Macro Dialog

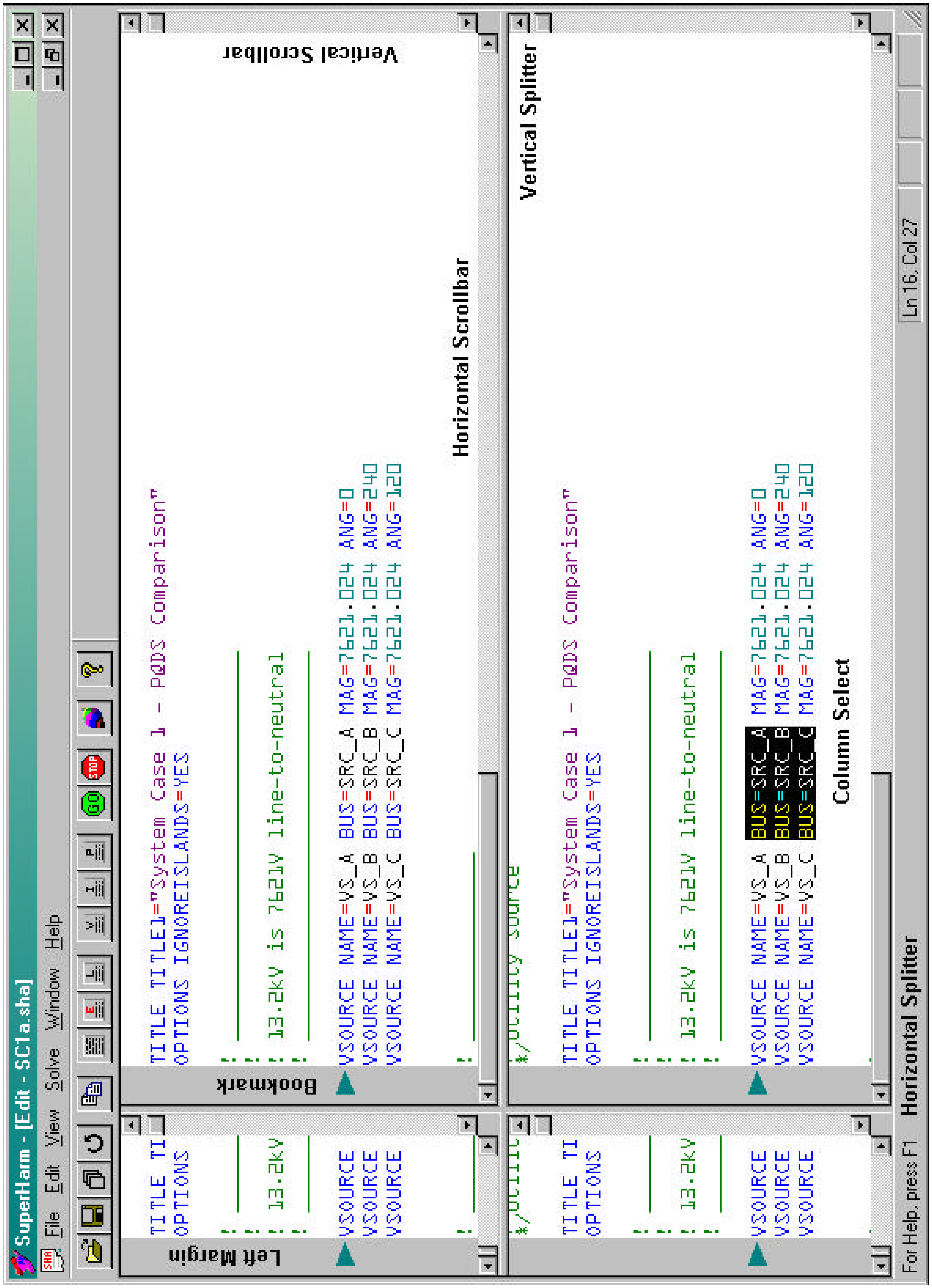


Figure 2-26: The Editor Window

Keyboard Shortcuts

The SuperHarm editor has a default set of keyboard mappings that are automatically registered when the program is installed. The table below shows the complete set of default keystroke mappings.

Command	Keystroke
BookmarkNext	F2 (refer to Figure 2-10)
BookmarkPrev	Shift + F2
BookmarkToggle	Control + F2
CharLeft	Left
CharLeftExtend	Shift + Left
CharRight	Right
CharRightExtend	Shift + Right
Copy	Control + C
Copy	Control + Insert
Cut	Shift + Delete
Cut	Control + X
CutSelection	Control + Alt + W
Delete	Delete
DeleteBack	Backspace
DocumentEnd	Control + End
DocumentEndExtend	Control + Shift + End
DocumentStart	Control + Home
DocumentStartExtend	Control + Shift + Home
Find	Alt + F3
Find	Control + F
FindNext	F3
FindNextWord	Control + F3
FindPrev	Shift + F3
FindPrevWord	Control + Shift + F3
FindReplace	Control + Alt + F3
GoToLine	Control + G
GoToMatchBrace	Control +]
Home	Home
HomeExtend	Shift + Home
IndentSelection	Tab
LineCut	Control + Y

LineDown	Down
LineDowNextend	Shift + Down
LineEnd	End
LineEndExtend	Shift + End
LineOpenAbove	Control + Shift + N
LineUp	Up
LineUpExtend	Shift + Up
LowercaseSelection	Control + U
PageDown	Next
PageDowNextend	Shift + Next
PageUp	PRIOR
PageUpExtend	Shift + Prior
Paste	Control + V
Paste	Shift + Insert
Properties	Alt + Enter
RecordMacro	Control + Shift + R
Redo	Control + A
SelectLine	Control + Alt + F8
SelectSwapAnchor	Control + Shift + X
SentenceCut	Control + Alt + K
SentenceLeft	Control + Alt + Left
SentenceRight	Control + Alt + Right
SetRepeatCount	Control + R
TabifySelection	Control + Shift + T
ToggleOvertype	Insert
ToggleWhitespaceDisplay	Control + Alt + T
Undo	Control + Z
Undo	Alt + Backspace
UnindentSelection	Shift + Tab
UntabifySelection	Control + Shift + Space
UppercaseSelection	Control + Shift + U
WindowScrollDown	Control + Up
WindowScrollLeft	Control + PageUp
WindowScrollRight	Control + PageDown
WindowScrollUp	Control + Down
WordDeleteToEnd	Control + Delete
WordDeleteToStart	Control + Backspace

WordLeft	Control + Left
WordLeftExtend	Control + Shift + Left
WordRight	Control + Right
WordRightExtend	Control + Shift + Right

Mouse Shortcuts

The default right-click menu is illustrated in Figure 2-11.

Figure 2-27: The Editor's Default Right-Click Menu

In addition, the editor supports the following mouse actions:

Mouse Action:	Result:
L-Button click over text	Changes the caret position
R-Button click	Displays the right click menu
L-Button down over selection, and drag	Moves text
Ctrl + L-Button down over selection, and drag	Copies text
L-Button click over left margin	Selects line
L-Button click over left margin, and drag	Selects multiple lines
Alt + L-Button down, and drag	Select columns of text (refer to Figure 2-10)
L-Button double click over text	Select word under cursor
Spin IntelliMouse mouse wheel	Scroll the window vertically
Single click IntelliMouse mouse wheel	Select the word under the cursor
Double click IntelliMouse mouse wheel	Select the line under the cursor
Click and drag splitter bar	Split the window into multiple views or adjust the current splitter position
Double click splitter bar	Split the window in half into multiple views or unsplit the window if already split

CHAPTER 3

AN OVERVIEW OF THE CDL

This chapter provides a description of specific types of data accepted by SuperHarm. In a data file, a circuit is described by a series of device entries. Each entry starts with a *keyword* indicating device type; the device name, terminal names and device-specific data follow. Data files also contain *directives*, which perform a number of functions, such as specifying what currents and voltages to save to the output file.

The final section of this chapter illustrates example SuperHarm data files.

In This Chapter

Page Intentionally Left Blank

Data Types

Device and Node Names

Each device in a circuit must be given a unique identifier, which may consist of up to sixteen alphanumeric characters. Case is not significant - 'A' and 'a' are considered to be the same character by the program. The underscore '_' may also be used.

The names of the terminal nodes must be specified for each device. The rules for naming nodes are identical to those used for naming devices: alphanumeric strings of eight characters or less. By convention, the phases in a three-phase node are given a common root name of six characters or less. A dot-plus-phase letter is appended to the root to distinguish the individual phases. For example, three-phase node "`BusX`" would be referenced as:

```
BusX.A    BusX.B    BusX.C
```

Voltage and current sources are identified by node names and are assumed to be between the node and the local ground. The node name "`GROUND`" is reserved for the local ground.

Device Example

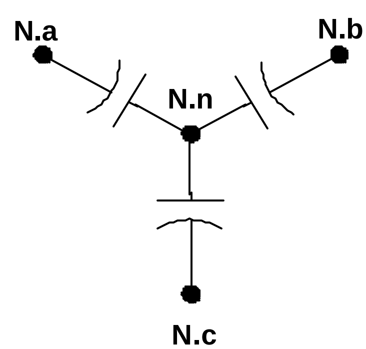


Figure 3-28: SuperHarm Device Example

A 10 MVAr, three-phase capacitor bank is applied at 60kV (rms line-to-line voltage). Using the three-phase capacitor model, the data is entered as follows:

```
CAPACITOR      Name = C3
Bus.A = N.a    Bus.B = N.b  Bus.C= N.c
Neutral = N.n  kV = 60          MVA = 10
```

Numeric Data

Any number required by SuperHarm can be entered using integer, floating point, or exponential notation. For example, the number `10.0` can be specified as:

```
10   +10    10.    10.0   +10.   +10.0  1E1
1E+1 1E+01  1.E1   1.0E1  +1E1   etc.
```

File Types

Data File Naming Convention

SuperHarm reads a circuit description from a simple text file that includes a series of device entries and directives. The circuit description may be a single file or may include a number of more advanced options, such as "include" and "library" files.

While the user has the complete control over file naming, a suggested standard file naming convention is summarized in the following table. Note that naming consistency may improve the efficient transfer of files between users.

Table 3-1: Data File Naming Convention

File Extension	Description
SHA	SuperHarm Input Data File
SHO	SuperHarm Output File
SHE	SuperHarm Error File
SHP	SuperHarm Parser Log
SHI	SuperHarm Include File
SHL	SuperHarm Library File
SHD	SuperHarm Define File

Tagged Field, List, and Table Formats

Tagged Fields

SuperHarm is very flexible with regard to how you enter data because it uses a tagged field data format: parameters are identified by name (e.g. `kV = 34.50`).

- Devices and directives can be entered in any order.
- Case is insignificant for both device and field names.
- Device parameters can be entered without regard to position or order.

Therefore, although the following device entry for capacitor C3 is not visually appealing, it is completely legal.

```
caPACitor mVa = 10
neutral = n.n kv = 60.00     bUS.C = n.C
NAME = C3      BUS.B = N.B   buS.A = N.a
```

Lists

A *list* is a set of items separated by commas and enclosed in braces. As an example, consider the following directive, which causes SuperHarm to save terminal currents for the specified devices:

```
RETAIN CURRENTLIST = {SourceZ, Xfmr, Fdr1, ASD1, Cap1}
```

Tables

A *table* is a list of items, which each item is itself a list. Consider NONLINEARLOAD - this model requires that the harmonic current injection characteristic of the device be entered as a table of magnitudes and angles for an arbitrary number of frequencies. As a list of lists, the syntax is exactly as you would expect.

TIP: Avoid two common table mistakes – be sure to add a comma after the last list entry, and end the table with a brace “}”.

```
NONLINEARLOAD NAME=SRC1 BUS=RECT KVA=858.1599 KV=2.4018 DF=0.97107
TABLE=
{
{  1,     357.30050,  166.19},  //  commas separate items
{  5,      65.16396,  110.51},
{  7,      42.53477,   82.08},
{ 11,      20.48435,   22.63},
{ 13,      14.32440,   -9.39},
{ 17,       6.91028,  -82.42},
{ 19,       4.96316, -125.51},
{ 23,       3.35249,  141.58},
{ 25,       3.07114,   98.55},
{ 29,       2.54135,   20.73},
{ 31,       2.21511,  -16.98},
{ 35,       1.57854,  -96.41},
{ 37,       1.34501, -139.45}   //  do not put a comma here
}
```

Figure 3-29: Table Example

Data File Examples

Simple Circuit Data File

```
TITLE TITLE1="Example 1 - 44 kV Distribution System"
      TITLE2="All Capacitors in Service"
      TITLE3="With Load"
!
! Case A: All Capacitors in except proposed bank for BUS6
!         With Load / Harmonic simulation case
!
! Utility source
!
VSOURCE NAME=VSRC BUS=SRCV MAG=25400
!
! Harmonic Source for Cases with Rectifier Modeled
!
!   2.5 MVA Rectifier, 6 Pulse, 4.16 kV, 6% commutation reactance
!
```

```
!
NONLINEARLOAD NAME=SRC1 BUS=RECT KVA=858.1599 KV=2.4018 DF=0.97107
TABLE=
{
{  1,     357.30050,  166.19},
{  5,      65.16396,  110.51},
{  7,      42.53477,   82.08},
{ 11,      20.48435,   22.63},
{ 13,      14.32440,   -9.39},
{ 17,       6.91028,  -82.42},
{ 19,       4.96316, -125.51},
{ 23,       3.35249,  141.58},
{ 25,       3.07114,   98.55},
{ 29,       2.54135,   20.73},
{ 31,       2.21511,  -16.98},
{ 35,       1.57854,  -96.41},
{ 37,       1.34501, -139.45}
}
!
!  Metering element in series with the rectifier load
!
BRANCH  NAME=rectif  FROM=bus7l  TO=rect  R=1.0  X=0.0
!
! Source Equivalent at 44KVBUS
!   Note: All Impedance Values Given in Ohms at 60 Hz
!
BRANCH NAME=equiv FROM=44kvbus TO=SRCV R=2.46 X=5.65
!
! Capacitor Banks on the System
!   All the Capacitors are in Service
!
CAPACITOR NAME=bank01 FROM=44kvbus R=0.0 KV=46.0 MVA=6.00
CAPACITOR NAME=bank02 FROM=bus2l   R=0.0 KV=13.8 MVA=1.05
CAPACITOR NAME=bank03 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank04 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank05 FROM=bus4l   R=0.0 KV=13.8 MVA=0.75
!
! Loads
!
```

```
LINEARLOAD NAME=LOAD1 FROM=bus1  KVA=4500.0 KV=44.00 DF=0.99018
LINEARLOAD NAME=LOAD2 FROM=bus2l KVA=3100.0 KV=13.00 DF=0.94606
LINEARLOAD NAME=LOAD3 FROM=bus3l KVA=1100.0 KV=12.50 DF=0.85749
LINEARLOAD NAME=LOAD4 FROM=bus4l KVA=1300.0 KV=12.50 DF=0.97823
LINEARLOAD NAME=LOAD6 FROM=bus6  KVA= 600.0 KV=44.00 DF=0.85749
LINEARLOAD NAME=LOAD7 FROM=bus7l KVA=2000.0 KV= 4.16 DF=0.90000
!
! Transformers from 44 kV to Customer Buses
!
TRANSFORMER       NAME = B2TRAN  MVAb.HX = 100.0     MVA=10
 H.1  = BUS2      X.1  = BUS2L
kV.H  = 44.00    kV.X  = 13.00
%R.HX = 3.20     %X.HX = 86.80

TRANSFORMER       NAME = B3TRAN  MVAb.HX = 100.0     MVA=10
 H.1  = BUS3      X.1  = BUS3L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 12.10    %X.HX = 130.40

TRANSFORMER       NAME = B4TRAN  MVAb.HX = 100.0     MVA=10
 H.1  = BUS4      X.1  = BUS4L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 29.20    %X.HX = 214.5

TRANSFORMER       NAME = B7TRAN  MVAb.HX = 7.5       MVA=10
 H.1  = BUS7      X.1  = BUS7L
kV.H  = 44.00    kV.X  = 4.16
                 %X.HX = 4.6
!
! 44 kV Distribution Lines
!   Note: Line capacitance not included due to short line lengths
!         Impedances are in Ohms at 44 kV
!
BRANCH NAME=LINE1 FROM=44KVBUS TO=BUS1  R=0.0581  X=1.2778
BRANCH NAME=LINE2 FROM=44KVBUS TO=BUS2  R=3.678   X=8.0150
BRANCH NAME=LINE3 FROM=44KVBUS TO=BUS3  R=0.987   X=2.9810
BRANCH NAME=LINE4 FROM=44KVBUS TO=BUS4  R=1.084   X=1.7040
BRANCH NAME=LINE5 FROM=BUS4    TO=BUS6  R=3.717   X=7.8410
BRANCH NAME=LINE6 FROM=BUS6    TO=BUS7  R=0.620   X=1.3340
```

```
!
! Save currents for harmonic mode cases
!
RETAIN CURRENTS=YES
!
! End of Input File
!
....
```

Sample Include File

```
!
!  Source Equivalent Voltage
!
VSOURCE NAME=VS25A BUS=V25EA MAG=V25ELG ANG=   0.0
VSOURCE NAME=VS25B BUS=V25EB MAG=V25ELG ANG=-120.0
VSOURCE NAME=VS25C BUS=V25EC MAG=V25ELG ANG= 120.0
!
!  Source Equivalent Impedance
!
BRANCH3 NAME=EQ25KV   KVLL=kVbLL  MVA3=MVAb  LENGTH=100.0
FROM.A=V25EA    TO.A=Zs_A
FROM.B=V25EB    TO.B=Zs_B
FROM.C=V25EC    TO.C=Zs_C
R1=0.14625      X1=0.92083
R0=0.11790      X0=0.14481
!
!  Line Info / Use line codes in library file MODEL.SHL
!
!  Parameters:
!       1 - Line Name
!       2 - From Bus Name (without phase identifier)
!       3 - To   Bus Name (without phase identifier)
!       4 - Length of line in miles
!
#library MODEL.SHL 3X296  (LZs-A,  Zs,  A,  0.25)
#library MODEL.SHL 3X296  ( LA-B,   A,  B,  0.25)
```

```
#library MODEL.SHL 3X296  ( LB-C,   B,  C,  0.50)
!
!  Capacitor Banks
!
!  Parameters:
!       1 - Capacitor Name
!       2 - Bus Name (without phase identifier)
!       3 - 3 phase kVAr at rated voltage
!       4 - Capacitor rated voltage in kV L-L
!
#library MODEL.SHL CAP3-GY(LC1042, A, 1200.0, kVCap, SWT1)
#library MODEL.SHL CAP3-GY(LC4175, B, 1200.0, kVCap, SWT2)
#library MODEL.SHL CAP3-GY(LC1022, L,  600.0, kVCap, SWT5)
!
```

Sample Library File

```
!
!  Model Line Library
!
[3X296]
!
!  Impedance values are in ohms/mile therefore
!  KVLL and MVA3 tags are not used
!
BRANCH3 NAME=%1 LENGTH=%4
FROM.A=%2^_A  TO.A=%3^_A
FROM.B=%2^_B  TO.B=%3^_B
FROM.C=%2^_C  TO.C=%3^_C
R1=0.306   X1=0.628
R0=0.593   X0=3.030
!
 [3X4ACSR]
!
!  Impedance values are in ohms/mile therefore
!  KVLL and MVA3 tags are not used
!
BRANCH3 NAME=%1 LENGTH=%4
```

```
FROM.A=%2^_A  TO.A=%3^_A
FROM.B=%2^_B  TO.B=%3^_B
FROM.C=%2^_C  TO.C=%3^_C
R1=2.240   X1=0.774
R0=2.520   X0=3.170
!
 [3X2ACSR]
!
!  Impedance values are in ohms/mile therefore
!  KVLL and MVA3 tags are not used
!
BRANCH3 NAME=%1 LENGTH=%4
FROM.A=%2^_A  TO.A=%3^_A
FROM.B=%2^_B  TO.B=%3^_B
FROM.C=%2^_C  TO.C=%3^_C
R1=1.410   X1=0.775
R0=1.690   X0=3.150
!
!
[3X2-0]
!
!  Impedance values are in ohms/mile therefore
!  KVLL and MVA3 tags are not used
!
BRANCH3 NAME=%1 LENGTH=%4
FROM.A=%2^_A  TO.A=%3^_A
FROM.B=%2^_B  TO.B=%3^_B
FROM.C=%2^_C  TO.C=%3^_C
R1=0.706   X1=0.717
R0=0.986   X0=3.120
!
!  Misc Modules
!
[CAP3-GY]
!
!  Capacitor Bank Helper
!
SWITCH NAME=S^%1^A FROM=%2^_A TO=X^%1^A STATE=%5
SWITCH NAME=S^%1^B FROM=%2^_B TO=X^%1^B STATE=%5
```

```
SWITCH NAME=S^%1^C FROM=%2^_C TO=X^%1^C STATE=%5
CAPACITOR NAME=%1 KVA=%3 KV=%4 CONNECTION=WYE
  BUS.A=X^%1^A
  BUS.B=X^%1^B
  BUS.C=X^%1^C
!
[CAP3-GD]
!
!  Capacitor Bank Helper
!
SWITCH NAME=S^%1^A FROM=%2^_A TO=X^%1^A STATE=%5
SWITCH NAME=S^%1^B FROM=%2^_B TO=X^%1^B STATE=%5
SWITCH NAME=S^%1^C FROM=%2^_C TO=X^%1^C STATE=%5
CAPACITOR NAME=%1 KVA=%3 KV=%4 CONNECTION=DELTA
  BUS.A=X^%1^A
  BUS.B=X^%1^B
  BUS.C=X^%1^C
!
```

CHAPTER 4

DEVICE LIBRARY

This chapter provides a description of the component devices in SuperHarm.

In This Chapter

- Device Descriptions

Page Intentionally Left Blank

BRANCH

```
BRANCH    Name   = DevName
From        =  NodeName  To =  Node-
                              Name
XRConstant  =  Yes|No
Length      =  Value
R           =  Value     X   =  Value
                     OR
TABLE =  {  {Freq1,  R1,  X1 },
            {Freq2,  R2,  X2 },
                        .
                        .
            {FreqN,  RN,  XN }
         }
FreqMult =   Value
                       OR
Filename =  FileSpec
```

`BRANCH` models a single-phase branch. The branch may be a simple `RX`, or it may include frequency dependence. The branch is connected between the nodes given by the `From` and `To` fields. `To` is optional; if omitted, ground is assumed.

Simple RX

To use this model, enter parameters R and X. Either parameter can be omitted (or zero), but not both. Both parameters must be non-negative. R is independent of frequency unless you specify `XRConstant = Yes.` In that case, R is varied to keep the `X/R` ratio constant with frequency. (One example where this flag would be useful is if you are using `BRANCH` to represent a transformer in a per unitized system.)

Frequency Dependent Model

If R and X are replaced with a table, as shown above, then SuperHarm assumes that the branch is frequency dependent. You can specify R and X for any number of frequencies. SuperHarm interpolates or extrapolates when it needs to determine impedance at frequencies not listed in the table. If the optional `Length` parameter is omitted, R and X are assumed to be in ohms. To enter values in ohms per unit length, set `Length` equal to total length.

An Example

This example illustrates how to model a machine as a frequency dependent branch using the BRANCH device model. The motor has the following values:

R = 0.398Ω, X = 4.598Ω Rd''=0.046Ω, Xd'' = 0.692Ω

Frequency	r	x
60	R	X
120	2 * Rd''	2 * Xd''
240	3 * Rd''	3 * Xd''

```
BRANCH     Name   = Machine1
From         = bus1      To   = ground
TABLE =   { {  60,   .398,  4.598  },
            { 120,   .093,  1.384  },
            ( 240,   .132,  2.876  },
          }
```

Using an ASCII Data File

The frequency dependent model discussed above can also be incorporated into SuperHarm as an ASCII file in the same directory as the data file. To use this option, specify the filename with the `File` field. The ASCII file must contain two or more rows of three values; `freq` in `Hz, R` in ohms at specified frequency, and `X` in ohms at specified frequency. Entries may be separated by spaces, commas, tabs, braces and brackets. Ensure that there are no trailing blank lines in the file.

Three-Phase Representation

`Branch` can also be utilized to represent a three-phase system:

```
BRANCH    Name = B3a  From = Src.a To = Eqiv.a
R = 1.234         X = 8.681
BRANCH    Name = B3b  From = Src.b To = Eqiv.b
R = 1.234         X = 8.681
BRANCH    Name = B3c  From = Src.c To = Eqiv.c
R = 1.234         X = 8.681
```

This representation can be used in place of `BRANCH3` when the coupling effects between reactors is not a concern or when the zero sequence information is not available.

BRANCH3

```
BRANCH3   Name = DevName
From.A   = NodeName  To.A   = NodeName
From.B   = NodeName  To.B   = NodeName
From.C   = NodeName  To.C   = NodeName
R0       = Value     X0     = Value
R1       = Value     X1     = Value
Length   = Value     MVA3   = Value
kVLL     = Value
```

`BRANCH3` models a three-phase branch. By modeling the positive and zero sequence components, the coupling effects can be incorporated. `To.A`, `To.B`, and `To.C` are optional; if omitted, ground is assumed. The fundamental frequency positive-sequence and zero-sequence impedances are R1 + j X1 and R0 + j X0, respectively. A resistance, or the corresponding reactance can be omitted (or zero), but not both. Both parameters must be non-negative.

If the optional `Length` parameter is omitted, resistances and reactances are assumed to be in ohms. To enter values in ohms per unit length, set `Length` equal to total length. The optional `MVA3` and `kVLL` are utilized if resistances and reactances are entered in percent.

An Example

The following model represents a line with the following fundamental frequency impedance values:

Positive-sequence impedance of 0.261 + j 0.682 Ω/mile

Zero-sequence impedance of 0.545 + j 3.012 Ω/mile

Length is entered in miles to be consistent with these values.

```
BRANCH3     Name = Line3
From.A = Bus1.A       To.A  = Bus2.A
From.B = Bus1.B       To.B  = Bus2.B
From.C = Bus1.C       To.C  = Bus2.C
R1     = 0.261        X1    = 0.682
R0     = 0.545        X0    = 3.012
Length = 7
```

CAPACITOR

Single-Phase Model

CAPACITOR	Name = DevName			
From	= NodeName		To	= NodeName
R	= Value			
uF	= Value	***OR***	MVA|kVA	= Value
			kV	= Value

Three-Phase Model

CAPACITOR	Name = DevName			
Bus.A	= NodeName		Bus.B	= NodeName
Bus.C	= NodeName		Neutral	= NodeName
R	= Value		Connection	= Wye|Delta
uF	= Value	***OR***	MVA|kVA	= Value
			kV	= Value

`CAPACITOR` represents a single-phase or three-phase capacitor connected between the nodes shown below:

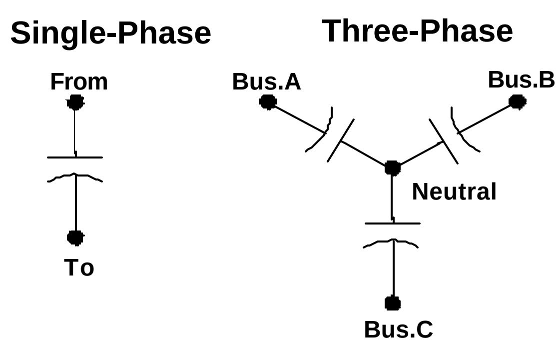


Figure 4-30: Capacitor Representation

The single-phase model's `To` node and the three-phase model's `Neutral` node are optional; if omitted, ground is assumed. Capacitance can be entered directly in microfarads (parameter uF), or indirectly, by specifying rated `MVA` and `kV`. (You may replace parameter `MVA` by `kVA` if you prefer.) Capacitor resistance `R` (ohms) is optional. The field `Connection` is used to specify the capacitor connections in the three-phase model as either wye or delta. If this field is omitted then wye is assumed.

An Example

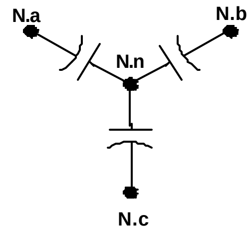


Figure 4-31: CAPACITOR Example

A 10 MVAr, three-phase ungrounded Y connected capacitor bank is applied at 60kV (rms line-to-line voltage). Using the three-phase capacitor model, the data is entered as follows:

```
CAPACITOR Name = C3
Bus.A = N.a
Bus.B = N.b
Bus.C = N.c
Neutral = N.n
kV = 60 MVA = 10
```

Alternate format:

```
uF = 7.3683
```

This capacitor bank may also be represented with three single-phase capacitor models:

```
CAPACITOR Name = C3a
From = N.a   To = N.n
kV = 34.64   MVA = 3.333

CAPACITOR Name = C3b
From = N.b   To = N.n
kV = 34.64   MVA = 3.333

CAPACITOR Name = C3c
From = N.c   To = N.n
kV = 34.64   MVA = 3.333
```

CMATRIX

```
CMATRIX      Name  = DevName
BUSLIST = { NodeName, NodeName, .  NodeName }
                                .
Mult    =  Value
Matrix = { {Value, Value,  . Value },
           {Value, Value,  . Value },
              .      .    .    .
              .      .    .    .
           {Value, Value,  . Value }
         }
```

CMATRIX allows the user to enter the capacitances for a system in a matrix formation. The diagonal values in the matrix are assumed to be shunt connected. The required parameters for this device are `BUSLIST`, `Matrix` and `Name`. `Mult` is optional and defaults to 1.

An Example

Given the system shown below:

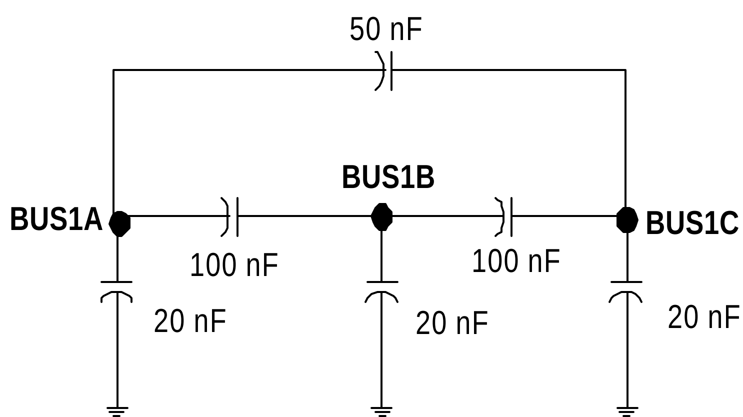


Figure 4-32: CMATRIX Example

The data is entered as follows:

```
CMATRIX NAME=CM1  MULT=1.0E-09
BUSLIST = { BUS1A, BUS1B, BUS1C }
MATRIX  = {
          {  20, 100,  50 },
          { 100,  20, 100 },
          {  50, 100,  20 }
          }
```

GENERATOR

Single-Phase Model

```
GENERATOR        Name  = DevName
From       = NodeName  To      = NodeName
MVA        = Value     kV      = Value
%Rs        = Value     %Xs     = Value
%Rh        = Value     %Xh     = Value
XRConstant = Value     Freq    = Value
Mag        = Value     Ang     = Value
```

Three-Phase Model

```
GENERATOR        Name  = DevName
Connection = Wye|Delta Bus.A   = NodeName
Bus.B      = NodeName  Bus.C   = NodeName
Neutral    = NodeName
MVA        = Value     kV      = Value
%Rs        = Value     %Xs     = Value
%Rh        = Value     %Xh     = Value
XRConstant = Value     Freq    = Value
Mag        = Value     Ang     = Value
```

GENERATOR model is simply a frequency dependent branch, similar to the MACHINE model, in parallel with a current source. It can be used to model a synchronous generator.

Parameter	Description	Required	Default
kV	Rated terminal voltage	Yes	---
MVA	Rated machine MVA, also base value for % impedances	Yes	---
Freq	Machine synchronous frequency	No	system base
Mag	Per-unit magnitude of voltage behind synchronous reactance	No	1
Ang	Angle of voltage behind synchronous reactance	No	0

For a three-phase model, Mag and Ang should be entered for the phase A to ground voltage reference, kV is the line-to-line rated voltage, and MVA is the three-phase MVA rating.

For a single-phase model, kV is the machine voltage rating, which may be line-to-line or line-to-neutral depending on the connection. MVA is also the machine rating. Mag and Ang should be referenced to the actual device connections, which may be line-to-line or line-to-neutral.

As for the MACHINE model, you may need to run an iterative load flow program to obtain good initial values for Mag and Ang.

The underlying model is a Norton equivalent, so you can't connect any voltage sources to the same node as a generator. However, you can parallel generators and other current sources at the same node. Also, if you have a system with only generators instead of VSOURCE devices, you would need a dummy VSOURCE at some node for SuperHarm to run.

Resistance and reactance should be entered in percent on the impedance base corresponding to the terminal voltage and MVA ratings. The branch impedance at fundamental frequency is given by %Rs and %Xs. Resistance and reactance at harmonic frequencies is given by %Rh and %Xh, respectively. Subtransient resistance and reactance are usually used for these parameters.

%Rh and %Xh are optional. The default values correspond to:

$$X_h = 0.170 * X_s$$

$$R_h = 0.025 * X_h$$

By default, losses are frequency-dependent; the program assumes that resistance, as well as reactance, increases linearly with frequency. However, if XRConstant = No is included in the device data, harmonic resistance remains constant.

An Example

A three-phase, 600-MVA, 13.2 kV machine could be specified as follows.

```
GENERATOR

NAME=gen1

BUS.A=srcbus.a BUS.B=srcbus.b BUS.C=srcbus.c

CONNECTION=delta

MVA=600.0   kV=13.2

%Xs=100     %Rs=5

%Xh=16      %Rh=0.8

FREQ=60     MAG=1.0    ANG=0.0
```

INDUCTIONMOTOR

Single-Phase Model

```
INDUCTIONMOTOR Name  = DevName
From     = NodeName  To      = NodeName
DF       = Value     kV      = Value
Poles    = Value     RPM     = Value
%Load    = Value     HP      = Value
%Eff     = Value
LRCode   = Value     LRKVA   = Value
LRPF     = Value     File    = FileSpec
```

Three-Phase Model

```
INDUCTIONMOTOR   Name  = DevName
Connection = Wye|Delta Bus.A  = NodeName
Bus.B      = NodeName  Bus.C  = NodeName
Neutral    = NodeName  HP     = Value
DF         = Value     kV     = Value
Poles      = Value     RPM    = Value
%Load      = Value     %Eff   = Value
LRCode     = Value     LRKVA  = Value
LRPF       = Value     File   = FileSpec
```

`INDUCTIONMOTOR` models a single-phase or three-phase induction motor. Besides the device name and bus names, only rated power (HP) and terminal voltage (kV, line-neutral for the single-phase model, line-line for the three-phase model) **must be** entered. The remaining parameters, if omitted, will default to the values indicated in this section.

The model shown in Figure 4-4 is the equivalent circuit for the induction motor.

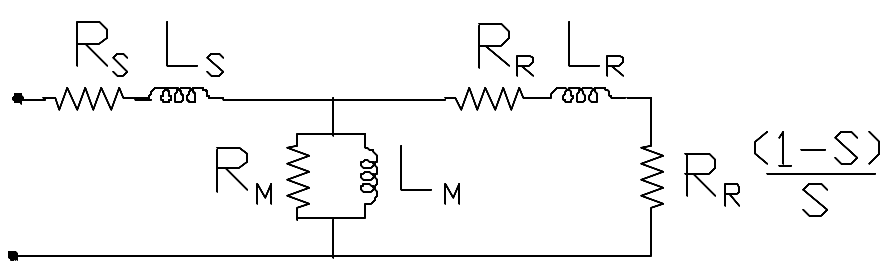


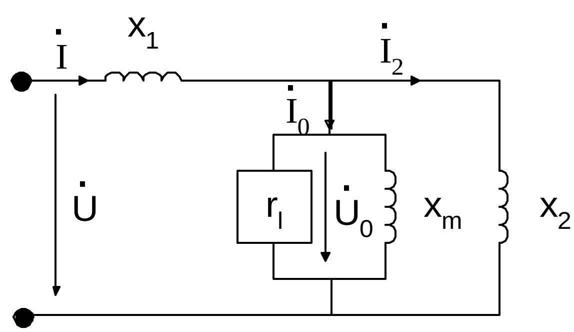
Figure 4-33: Induction Motor Equivalent Circuit

Parameter Determination – Method One (prior to Version 4)

The model parameters and assumptions utilized in this method are based upon a paper titled "*Studies on Modeling of Harmonic Impedance for Induction Motor*" by Zhang Jing and He Fengreng.

Four assumptions are made for this model:

- Stator impedance is equal to that of the rotor ($x_1 = x_2$)
- Exciting impedance is 35 times as high as that of stator ($x_m = 35x_1$)
- Exciting current is 30% of stator, and 70% of stator current flows through the rotor circuit (i.e., I_0 = 30% I_1 and I_2 = 70% I_1)
- Core loss of induction motor is 3% of nameplate rating



For the above figure, the reactive power equation is listed as follows:

$$Q_m/3 = I^2 x_1 + I_2^2 x_2 + I_0^2 x_m$$

Substituting with the assumptions made above, the following results:

$$x1 = (Qm\ U^2) / 4.64\ (Pm2 + Qm2)$$

core resistance is:

$r_l = (U_o^2) / 0.03\ P_m\ \eta$ – where η is the efficiency of the motor

Note: This calculation method may be accessed by using the following `OPTION`:

```
OPTIONS OLDMOTOR = YES
```

Parameter Determination – Method Two (Version 4)

The model parameters and assumptions utilized in this method are based upon standard locked-rotor kVA/HP ranges provided in NEMA MG 1-1993 (Revision 1, Part 10, Page 6), and the equivalent circuit for induction motors section of "*Electromagnetics and Electromechanical Machines*", by Leander Matsch (Third Edition, 1996, Harper and Row, ISBN 0-06-044271-9).

Assumptions utilized in this method include:

- Stator impedance is equal to that of the rotor ($x_1 = x_2$)
- Stator X/R ratio is equal to 4
- Core loss of induction motor is 1% of nameplate rating

NEMA designations for locked-rotor kVA per horsepower as measured at full voltage and rated frequencies are as follows.

NEMA Letter Designation	kVA/HP Range	SuperHarm Default
A	0 - 3.15	1.575
B	3.15 – 3.55	3.350
C	3.55 – 4.0	3.775
D	4.0 - 4.5	4.250
E	4.5 – 5.0	4.750
F	5.0 – 5.6	5.300
G	5.6 – 6.3	5.950
H	6.3 – 7.1	6.700 (default)
J	7.1 – 8.0	7.550
K	8.0 – 9.0	8.500
L	9.0 – 10.0	9.500
M	10.0 – 11.2	10.600
N	11.2 – 12.5	11.850
P	12.5 – 14.0	13.250
R	14.0 – 16.0	15.000
S	16.0 – 18.0	17.000
T	18.0 – 20.0	19.000
U	22.0 – 22.4	21.200
V	22.4 >	(not included)

Note: Locked kVA/HP range includes figure up to, but not including, the higher figure.

From the previous table, the locked-rotor kVA is:

$$kVA_{LR} = \left(kVA_{LR} / HP \right) HP$$

The locked-rotor parameters include:

$$I_{LR} = \frac{kVA_{LR}}{\sqrt{3} V_{LR}} \qquad Z_{LR} = \frac{\left(V_{LR} / \sqrt{3} \right)}{I_{LR}}$$

$$kW_{LR} = kVA_{LR} DF_{LR} \qquad R_{LR} = \frac{kW_{LR}}{3 I_{LR}^2}$$

$$X_{LR} = \sqrt{\left(Z_{LR}^2 - R_{LR}^2 \right)} \qquad X_{stator} = X_{rotor} = X_{LR} / 2$$

Note: This calculation method may be accessed by using the following `OPTION`:

`OPTIONS OLDMOTOR = NO (this is the default)`

Connections

The single-phase model must be connected in shunt; hence, only one terminal node `(Bus)` is entered. Three-phase terminal nodes are specified with the `Bus.A`, `Bus.B`, `Bus.C`, and `Neutral` parameters. `Connection` must be specified as `wye` or `delta.` The Neutral parameter is illegal with a delta connection. It is optional with a wye connection; if omitted, a grounded neutral is assumed.

Optional Parameters

Rotor slip, S, is computed from the number of stator pole pairs (`Poles`) and the operating shaft speed (`RPM`).

$$S = 1 - \frac{RPM \cdot Poles}{120 \cdot f}$$

where *f* is the fundamental frequency. A two-pole machine is assumed if `Poles` is omitted. The default `RPM` is the value corresponding to a slip of 2.0%.

`%DF`, `%Load`, and `%Eff` are the motor power factor, load factor, and efficiency, respectively. Their default values are:

```
DF ................. 0.85
%Load ............. 100.0
%Eff ............... 90.0
```

`LRCode`, `LRKVA`, and `LRDF` specify the NEMA letter designation, locked-rotor kVA/HP, and locked-rotor power factor, respectively. `LRCode` and `LRKVA` are mutually exclusive, but `LRDF` may be specified with either one. Their default values are:

```
LRCode ................ H
LRKVA ............... 6.7
LRDF ............. 0.2735
```

Examples

Given a 480V, 500HP, 2 pole induction motor with an efficiency of 90% and a displacement factor of 80%, the following datafile entry illustrates how this is entered with the `INDUCTIONMOTOR` device model as a three-phase representation.

```
OPTIONS OLDMOTOR = YES
INDUCTIONMOTOR   Name  = Motor1
Connection = Wye       Bus.A  = Vs.a
Bus.B      = Vs.b      Bus.C  = Vs.c
Neutral    = ground    HP     = 500
DF         = 0.80      kV     = 0.480
Poles      = 2         %Eff   = 90
%Load      = 100       File   = "Motor1.inf"
```

Given a standard three-phase, 480V, 100HP, NEMA class G motor, the following datafile entry illustrates how this is entered with the `INDUCTIONMOTOR` device model as a three-phase representation.

```
OPTIONS OLDMOTOR = NO
INDUCTIONMOTOR   Name  = Motor2
Connection = Wye       Bus.A  = Vs.a
Bus.B      = Vs.b      Bus.C  = Vs.c
Neutral    = ground    HP     = 100
kV         = 0.480     LRCode = G
```

ISOURCE

Single Frequency Model

```
ISOURCE    Name  = DevName
Bus      = NodeName Freq    = Value
Mag      = Value    Ang     = Value
FreqMult = Value    MagMult = Value
AngShift = Value
```

Multiple Frequency Model

```
ISOURCE      Name  = DevName
Bus     = NodeName FreqMult = Value
MagMult = Value    AngShift = Value
TABLE =  { {Freq1, Mag1, Ang1 },
           {Freq2, Mag2, Ang2 },
                      .
                      .
           {FreqN, MagN, AngN }
         }
```

ISOURCE is a constant current source model. There are two variations:

- The single frequency model injects current of known magnitude and angle at one frequency.
- The multiple frequency model injects currents at the specified frequencies, magnitudes, and angles.

The ISOURCE model can be helpful to users who are migrating from other admittance matrix harmonic simulation programs a familiar method of simulating nonlinear loads. However, the NONLINEARLOAD device is superior to ISOURCE for this function, and we recommend that you do not use ISOURCE.

The ISOURCE model can also be used for more general modeling of linear loads and sources with Norton equivalent circuits. The advantages over using VSOURCE to make Thevenin equivalents, are that several ISOURCE devices can be connected in parallel, and that two ISOURCE devices can represent a phase-phase connected load or source.

General Comments

- ISOURCE models are active in the fundamental frequency solution.

- `ISOURCE` models may not be connected to the same node as a `VSOURCE` model.

- A source must be connected in shunt.

- Three-phase sources must be modeled as three single-phase devices.

- The units for the current magnitude at each frequency (Mag) and the optional magnitude multiplier (MagMult) should be such that, when these quantities are multiplied, the product is the RMS amperes to inject at that frequency.

- Phase angles for all sources in the system model should be specified in terms of the same arbitrary reference. For example, if one of the 60 Hz voltage sources in the system is set to 0°, then the phase angles for the remaining sources are entered with entered respect to this 0° voltage.

The Single Frequency Model

The parameters for this model are frequency `Freq` in Hz, current magnitude `Mag` in amperes, and current phase angle `Ang` in degrees. `Mag` and `Ang` may be omitted; the default value are unity and zero, respectively.

The optional parameters `FreqMult, MagMult,` and `AngShift` are intended to be used with the multiple frequency model described below. However, they can also be used with the single frequency model, if desired.

The Multiple Frequency Model

A current source with arbitrary harmonic content can be specified by replacing the parameters `Freq, Mag,` and `Ang` with a table listing frequency, magnitude, and angle for each harmonic current injected by the source. Unlike the single frequency model, magnitudes and phase angles *must* be specified.

Three optional parameters are provided to simplify entry of table data:

- FreqMult - if given, frequencies are per unit of this value.

- MagMult - if given, magnitudes are per unit of this value.

- AngShift is used to time-shift the current waveform; a positive value corresponding to a forward shift. Phase angles are corrected as follows:

$$q_h \leftarrow q_h + h \cdot AngShift$$
$$1 \le h \le N$$

An Example

The use of these parameters to simplify data entry is illustrated by the following example:

> An ideal six-pulse converter is to be applied to a three-phase system model. The fundamental component of the phase A current is 50 amperes at an angle of -20 with respect to the system reference. The equations for the corresponding current source are:

$$I_h = \frac{I_1}{h}, \qquad q_{hA} = \begin{matrix} 0^\circ, & h = 1, \ 7, \ 13, ... \\ 180^\circ, & h = 5, \ 11, \ 17, ... \end{matrix}$$

$$q_{hB} = q_{hA} - h \cdot 120^\circ, \quad q_{hC} = q_{hA} + h \cdot 120^\circ$$

With Optional Parameters	Without Optional Parameters
ISOURCE Name= 6Pa Bus =Ex.A	ISOURCE Name= 6Pa Bus =Ex.A
// MagMult = I1 -> can enter Ih in pu of I1	// Manual correction: Ang1 = 0+(1*-20)
FreqMult =60 MagMult=50 AngShift =-20	// Ang5 = 180+(5*-20), Ang7 = 0+(7*-20)
TABLE = { { 1, 1, 0 },	TABLE = { { 60 50, -20 },
{ 5, .2, 180 },	{ 300 10, 80 },
{ 7, .1429, 0 } }	{ 420 7.14, -140 } }
ISOURCE Name= 6Pb Bus =Ex.B	ISOURCE Name= 6Pb Bus =Ex.B
// For phase B, AngShift = -20-120 = -140	// Manual correction: Ang1 = 0+(1*-140)
FreqMult =60 MagMult=50 AngShift =-140	// Ang5 = 180+(5*-140), Ang7 = 0+(7*-140)
TABLE = { { 1, 1, 0 },	TABLE = { { 60 50, -140 },
{ 5, .2, 180 },	{ 300 10, -160 },
{ 7, .1429, 0 } }	{ 420 7.14, 100 } }
ISOURCE Name= 6Pc Bus =Ex.C	ISOURCE Name= 6Pc Bus =Ex.C
// For phase C, AngShift = -20+120 = 100	// Manual correction: Ang1 = 0+
FreqMult =60 MagMult=50 AngShift =100	// Ang5 = 180+(5*100), Ang7 = 0+(7*100)
TABLE = { { 1, 1, 0 },	TABLE = { { 60 50, 100 },
{ 5, .2, 180 },	{ 300 10, -40 },
{ 7, .1429, 0 } }	{ 420 7.14, -20 } }

LINE

```
LINE                   Name        = DevName
Length     = Value     Nphase      = Value
Earth      = Value
Transposed = Yes|No    Segmented   = Yes|No
From       = List      To          = List
XCoord     = List      YCoord      = List
DCRes      = List      Diam        = List
GMR        = List
BSpac      = List      NCB         = List
Freq       = Value     Metric      = Yes|No
File       = FileSpecPath
```

`LINE` can be used to represent any number of mutually-coupled, multi-phase transmission lines. Phase unbalance, skin effect, and the frequency dependence of the earth return path (Carson's equations) are included in the model. `LINE` requires detailed information concerning conductor characteristics and tower geometry. If this information is not available, use the `BRANCH` or `PI` models.

`Metric` allows the choice of either using metric or english units. The default for this tag is `yes`. Line length (`Length`) is a required tag and may be entered in either (km or mi.) provided that it is consistent with the resistance data discussed below as well as the `Metric` flag. Earth resistivity is set by the `EARTH` tag and has a default value of 100 Ω-meters.

The `NPHASE` tag allows for the user to enter the number of phase conductors in the system. If the lists have more than `NPHASE` conductors, the `LINE` model assumes the extra conductors are ground wires.

The conductor data for the `LINE` model is entered in a series of lists. Each list contains the data for all the conductors of the type indicated by the list tag name. For example, `XCOORD` list might contain five entries - the first three are the `X` coordinates for the phase conductors, the rest for the ground conductors (note: ground conductors must be specified last in the parameter arrays). Each conductor also has a corresponding entry in the `FROM` and `TO` node name lists. Note that these are required tags.

Horizontal position `Xcoord` is with respect to an arbitrary reference line, as shown in Figure 4-5. Vertical position `Ycoord` is the average height of the conductor. Both of these tags are required.

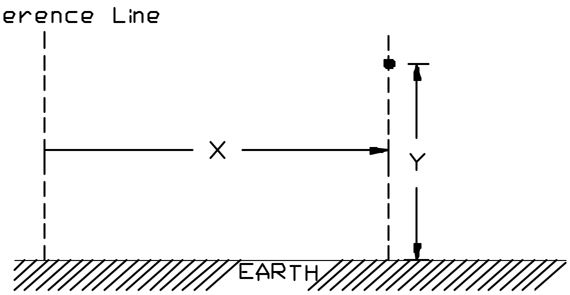


Figure 4-34: Conductor Position

Conductor characteristics:

Parameter	Description	Units
DCRES	Resistance at fundamental frequency	Ω/km or Ω/mile
DIAM	Outside diameter	cm or in
GMR	Geometric mean radius	m or ft

Note: The DCRES tag described above is a required input tag.

Bundled conductors

As described above, the `From`, `To`, `Xcoord`, and `Ycoord` parameters refer to single conductors. They may also be used to refer to a group of bundled conductors, provided that the conductors are identical, and the bundle geometry is such that all conductors are uniformly spaced around the edge of an imaginary circle, as shown in Figure 4-6.

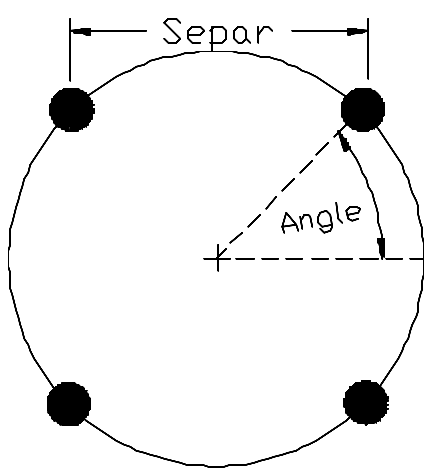


Figure 4-35: Conductor Bundle Geometry

To specify a conductor bundle, rather than a single conductor, enter the following information:

Bundle characteristics:

Parameter	Description	Units
NCB	Number of conductors in the bundle. (default is 1)	none
BSPAC	Bundle spacing - conductors arranged with equal spacing.	cm or in

Note: Each bundle is considered as one entry in the parameter lists.

It should be emphasized that the tags `XCOORD`, `YCOORD`, `FROM` and `TO` normally contain a list of parameters for each conductor, however they can also refer to the data for a conductor bundle when the associated `NCB` list member is > 1.

Transposed Conductors

In general, T+1 `LINE` models are required to simulate a line having T transpositions. However, a single line of N phases having N uniformly spaced transposition points can be represented with a single model by inserting the `Transposed = Yes` and utilizing the `Nphase` tag.

Ground Wires

Ground wires are treated like any other conductor; enter Ground for the `From` and `To` nodes. Segmented ground wires* are specified by entering `Segmented = Yes`. This then excludes the ground wires from the series impedance calculation.

Using the FILE Tag

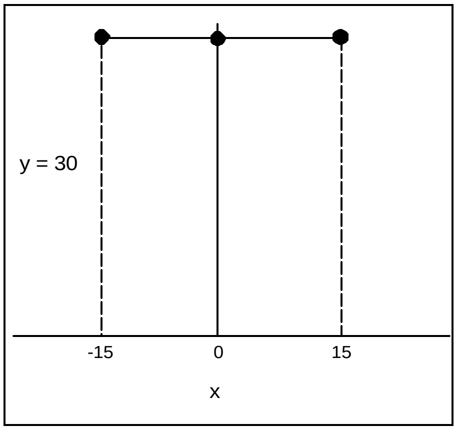
The `FILE` tag can be utilized to specify the full path name of a file that is to contain the results of the line constants calculation for the fundamental or a specified frequency. The `FREQ` tag can be used to set this frequency. This tag defaults to `FBASE`. An excerpt from a sample line constants output is shown on the next page.

* "Segmented" refers to a ground wire that does not form a continuous conducting path along the entire length of the line.

```
Number of Phase  Conductors …..: 3
Number of Ground Conductors ...: 0
Phase  Conductors Transposed ..: FALSE
Ground Conductors Segmented  ..: FALSE
Frequency for Constants (Hz) ..: 60
Earth Resistivity (ohm-meters).: 100
Units for Input Data ..........: Metric (SI)

ID  DC Res     GMR    Diameter     X Coord     Y Coord  NB    Spacing
    ohms/km     cm          cm      meters      meters             cm
--  --------  ------  ----------  --------    --------  --  ----------
 P  0.0863    0.0404       1.212     -17.5          96     1          0
 P  0.0863    0.0404       1.212     -13.5          81     1          0
 P  0.0863    0.0404       1.212     -13.5          13     1          0

L mH/km - Before reduction:
 2.9483        0.83434       0.81638
 0.83434       2.9430        0.69266
 0.81638       0.69266       2.9541

Z Ohms/km - Before ground wire reduction:
 0.13530       0.048923      0.047489
 1.1115        0.31454       0.30777

 0.048923      0.13668       0.048151
 0.31454       1.1095        0.26113

 0.047489      0.048151      0.13382
 0.30777       0.26113       1.1137

C nF/km:
 5.9374       -1.2067       -1.1936
-1.2067        5.8731       -0.72090
-1.1936       -0.72090       5.6898

Characteristic Impedance Zc (Mag/Angle) - Ohms:
 1089.5
-3.9291

 544.44
-3.2228

 596.15
-2.7979

Velocity - km/s:
 2.4232E+005
 2.5745E+005
 2.6098E+005
...
..
.
```

An Example

The following example illustrates how the `LINE` model can be utilized to represent the following system (length is in miles).




```
LINE NAME=L1
NPHASE=3
SEGMENTED=NO
TRANSPOSED=NO
METRIC=NO
LENGTH=2.0
  FROM = { BUS1A, BUS1B, BUS1C }
    TO = { BUS2A, BUS2B, BUS2C }
XCOORD = { -15.0,  0.0, 15.0 }
YCOORD = {  30.0, 30.0, 30.0 }
 DCRES = { 0.062137, 0.062137, 0.062137 }
   GMR = { 0.001042, 0. 001042, 0. 001042 }
  DIAM = { 0.0250, 0.0250, 0.0250 }
```

LINEARLOAD

Single-Phase Model

```
LINEARLOAD     Name  = DevName
From      = NodeName To         = Node-
                                  Name
kV        = Value    kVA        = Value
DF        = Value    XRConstant = Yes|No
%Parallel = Value    %Series    = Value
%R        = Value    %X         = Value
kVAXfmr   = Value
```

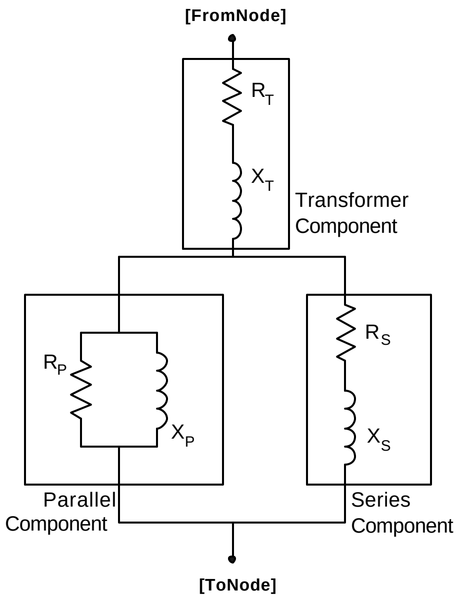
Three-Phase Model

```
LINEARLOAD       Name  = DevName
Bus.A      = NodeName  Bus.B   = NodeName
Bus.C      = NodeName  Neutral = NodeName
kV         = Value     kVA     = Value
XRConstant = Yes|No    DF      = Value
%X         = Value     %R      = Value
%Parallel  = Value     %Series = Value
kVAXfmr    = Value
```

`LINEARLOAD` models a single-phase or three-phase load given in terms of `kVA`, `kV`, and displacement power factor `DF`.

The model consists of three components, as shown in Figure 4-7. The series and parallel components consume the fraction of total kVA specified in `%Series` and `%Parallel`, respectively. These components are optional; if `%Series` and `%Parallel` are omitted, it is assumed that `%Parallel = 100`. If both components do exist, they must have the same displacement factor.

The transformer component is calculated from parameters `kVAXfmr`, `%R` and `%X`. The transformer is assumed not to exist if the `kVAXfmr` tag is omitted. Resistance for the transformer component is assumed to be constant with frequency. However, if the optional flag `XRConstant = Yes` is used, then SuperHarm will vary resistance in order to maintain a constant XR ratio. When this device is shunt connected - the usual case - it is unnecessary to enter the `To` parameter. `To = Ground` is assumed.



$$Z_{bt} = \frac{1000 * kV^2}{kVA_{xfmr}} \qquad Z_b = \frac{1000 * kV^2}{kVA}$$

$$R_T = \frac{\%R * Z_{bt}}{100} \qquad X_T = \frac{\%X * Z_{bt}}{100}$$

$$R_P = \frac{Z_b}{DF * \frac{\%Parallel}{100}} \qquad X_P = \frac{Z_b}{\frac{\%Parallel}{100} * \sin(a\cos(DF))}$$

$$R_S = \frac{Z_b * DF}{\frac{\%Series}{100}} \qquad X_S = \frac{Z_b * \sin(a\cos(DF))}{\frac{\%Series}{100}}$$

$$kVA_s = kVA * \%Series \qquad kVA_p = kVA * \%Parallel$$

Figure 4-36: Linear Load Model

An Example

A 480 V, 800 kVA linear load has a displacement power factor of 0.85 lagging. The load is supplied through a 1 MVA, 4.16 kV / 480 V transformer having a leakage impedance of 0.6 + j 5.9 % based on its rated MVA.

```
LINEARLOAD    Name  = LinLoad
From      = Bus4160v
kV        = 4.16     kVA     = 800
DF        = 0.85
kVAXfmr   = 1000
%R        = 0.6      %X      = 5.9
```

This device is shunt-connected since the `To` parameter is omitted. `%Parallel` and `%Series` default to 100 and 0, respectively, and the displacement factor is assumed to be lagging. Transformer resistance is constant with frequency since no `XRConstant` flag is given.

%Parallel and %Series Selection

The `%Parallel` and `%Series` selection relies entirely on the load to be modeled. Typically, all motor load should be 100% series and all power electronic load should be 100% parallel. However, it should be noted that a 100% series load would effectively remove the damping at high frequencies. A recommended practice with mixed loading is 50% series and 50% parallel.

MACHINE

Single-Phase Model

```
MACHINE         Name  = DevName
From       = NodeName  To      = NodeName
HP         = Value     kV      = Value
%Eff       = Value     %Load   = Value
DF         = Value     KVABase = Value
%Rh        = Value     %Xh     = Value
XRConstant = Value
```

Three-Phase Model

```
MACHINE         Name  = DevName
Connection = Wye|Delta Bus.A   = NodeName
Bus.B      = NodeName  Bus.C   = NodeName
Neutral    = NodeName  HP      = Value
kV         = Value     %Load   = Value
DF         = Value
XRConstant = Value     %Eff    = Value
%Rh        = Value     %Xh     = Value
```

`MACHINE` model is simply a frequency dependent branch. It is most suitable for modeling a synchronous motor, but can also model a synchronous generator in conjunction with `VSOURCE`. The branch impedance at fundamental frequency is calculated as follows:

$$R_1 = \frac{100 \cdot Z_b}{\%Load}, \quad X_1 = R_1 * Tan\left(Cos^{-1}\frac{\%PF}{100}\right)$$

Parameter	Description	Required	Default
kV	Rated terminal voltage	Yes	---
kVABase	Rated system kVA	No	HP entry converts to kVA using DF
HP	Rated horsepower	Yes	---
%Eff	Percent motor efficiency	No	90
%Load	Load factor in percent	No	100
DF	Displacement factor	No	85

Resistance and reactance at harmonic frequencies is given by %Rh and %Xh, respectively. Subtransient resistance and reactance are usually used for these parameters:

By default, losses are frequency-dependent; the program assumes that resistance, as well as reactance, increases linearly with frequency. However, if XRConstant = No is included in the device data, harmonic resistance remains constant.

Resistance and reactance should be entered in percent on the impedance base corresponding to the terminal voltage, horsepower, and efficiency:

$$Z_b = \frac{kV_b^{\,2}}{MVA_b} = \left(\frac{13.412 * \%Eff}{HP}\right) \cdot kV^2$$

%Rh and %Xh are optional. The default values correspond to:

$X_h = 0.170 * X_1$

$R_h = 0.025 * X_h$

Generator Representation

To model a generator with a MACHINE model, it is necessary to use the VSOURCE model behind it to represent the 60Hz voltage source. This is necessary to obtain the correct power flow generated between the generator and the system. It should be noted that it may be necessary to vary the voltage and differential angles to obtain the most accurate representation of the generator. If several parallel generator models are used in a model then it is more difficult to accomplish this. A load flow solution provided for the system can make this an easier process.

An example of when a generator model is found to be a necessity is when an islanding situation is being investigated. This situation could involve single or multiple generators supplying power to a customer off the system.

Note that the new GENERATOR model is a better choice for this application, because it is based on current sources. Several generators can be connected in parallel, or single-phase line-to-line.

Connections

The single-phase model must be connected in shunt; hence, only one terminal node (Bus) is entered. Three-phase terminal nodes are specified with the Bus.A, Bus.B, Bus.C, and Neutral parameters. The Connection parameter must be specified as wye or delta. The Neutral parameter is illegal with a delta connection. It is optional with a wye connection; if omitted, a grounded neutral is assumed.

An Example

Given a synchronous machine with a displacement power factor of 75% and operating at 100 HP the following model parameters are used to represent this as a frequency dependent branch.

```
MACHINE          Name  = M1
Bus        = NodeName  HP     = 100
kV         = 0.277     DF     = 0.75
KVABase    = 50
%Rh        = 0.05      %Xh    = 1.68
```

NONLINEARLOAD

Single-Phase Model

```
NONLINEARLOAD  Name  = DevName
Bus     = NodeName kVA      = Value
kV      = Value    DF       = Value
                   Leading  = Yes|No

TABLE =  { {H1,    Mag1, Ang1 },
           {H2,    Mag2, Ang2 },
                     .
                     .
           {HN,    MagN, AngN }
         }
FreqMult = Value
```

Three-Phase Model

```
NONLINEARLOAD  Name  = DevName
Bus.A   = NodeName Bus.B    = NodeName
Bus.C   = NodeName DF       = Value
kVA     = Value    Leading  = Yes|No
kV       = Value
TABLE =  { {H1,    Mag1, Ang1 },
           {H2,    Mag2, Ang2 },
                     .
                     .
           {HN,    MagN, AngN }
         }
FreqMult = Value
```

where:

- kV and kVA are the fundamental frequency RMS voltage and apparent power.
- DF is the displacement power factor. DF is assumed to be lagging unless the optional flag Leading = Yes is used.

HI, MagI, and AngI are the harmonic number, magnitude and degree phase angle of the Ith harmonic component. Note that the fundamental component **must** be entered in the table. This may seem redundant, since kV, kVA, and DF are also required. However, it allows you to enter magnitudes in any unit - amperes, percent, per unit, etc. - provided that the magnitudes are consistent with each other. Also, the AngI values can be chosen with respect to any common arbitrary reference angle. SuperHarm will adjust the magnitude and phase spectra automatically.

When simulating a harmonic producing device, you have two choices:

- Use an analytical model.
- Represent the device as a harmonic current source.

Since analytical models are not available for the majority of harmonic loads, you will use the harmonic current source representation most often. You may already be familiar with this technique from your experience with other simulation programs, and thus be tempted to simply use SuperHarm's `ISOURCE` model. However, the `NONLINEARLOAD` model is a better alternative. Here is why.

Admittance matrix simulation programs like SuperHarm use a single arbitrary reference for all voltage and current phase angles. Consider, for example, the following system, consisting of a voltage source at bus *S* and adjustable speed drives at *A* and *B:*

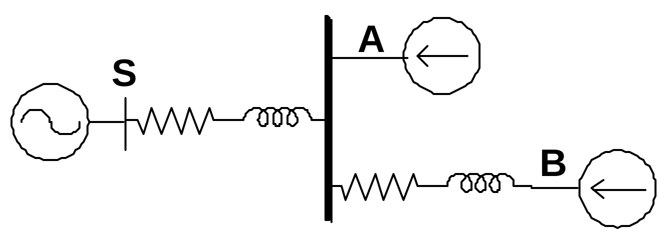


Figure 4-37: Harmonic Current Injection using a Nonlinear Load Model

Bus *S* is selected as the reference by setting Ang = 0 in the `VSOURCE` model. Now the phase angle spectra for the current sources representing the drives must be entered with respect to the fundamental voltage at *S*. But you probably do not have that information, even if harmonic measurements have been taken at the drives (this would require simultaneous monitoring of drive current and reference bus voltage).
If the measured phase angle spectra for each drive is with respect to the fundamental component of voltage at the drive bus (the usual case if the instru-

ment monitors voltage and current), then the current phase angles could be manually corrected as follows:

$$q_h = q_{h\,Meas} + h \cdot q_{V1}, \quad q_{V1} = \text{Angle of fund . voltage}$$

This corrected spectrum could then be used in an `ISOURCE` representation of the drive. The phase angle of the fundamental voltage at the drive must be known to make this correction; it can be estimated by running a preliminary fundamental frequency simulation with the drive represented as a `LINEARLOAD`.

The `NONLINEARLOAD` model makes this correction automatically. If SuperHarm finds one of these devices, it does the following:

1. Conducts a preliminary fundamental frequency simulation with each `NONLINEARLOAD` replaced by the impedance:

$$Z_1 = \frac{1000 \cdot \text{kV}^2}{\text{kVA}} \angle \text{Cos}^{-1}(\text{DF})$$

2. Uses the fundamental voltage angle at each `NONLINEARLOAD`, calculated in step 1, to correct the device's harmonic current angles.
3. Uses the magnitude of the fundamental voltage at each `NONLINEARLOAD`, calculated in step 1, and the given value of kVA to scale the harmonic current magnitudes:

$$I_h = I_{h_{given}} \left(\frac{1}{I_{1_{givien}}} \right) \left(\frac{kVA_{given}}{kV_{FundFreqSimulation}} \right)$$

Note that the calculated value of fundamental voltage, rather than the given value, is used to scale the current magnitudes. Therefore, unless the given kV is equal to the calculated kV, the scaled current magnitudes will not exactly match what you have specified.

TIP: Care should be exercised to assure that the measurement device and the simulation programs use the same convention for representation of the Fourier series (i.e. sine or cosine series). For example, the HARMFLO Program (V5.0), and the BMI 3030 use a sine series, and SuperHarm, the EMTP, and the Dranetz 658 use a cosine series. In the event that the measured waveform and simulation program use a different convention, the following relationship may be used to convert the phase angles:

$$\Theta_{sim} = \Theta_{meas} + (h-1) * 90°$$

An Example

An adjustable speed drive is modeled from harmonic current measurements as follows:

```
//  Model of an 250 HP ASD (dc drive) on a 480

NONLINEARLOAD    NAME = DCDRV1A
                BUS.A = 4802A
                BUS.B = 4802B
                BUS.C = 4802C
                   KV = 0.480          KVA =
250.0
                   DF = 0.75       LEADING = NO
             FREQMULT = 60
                TABLE = {
                        { 1,     100.0,    -75.0},
                        { 5,      33.6,    156.0},
                        { 7,       1.6,   -151.0},
                        {11,       8.7,   -131.0},
                        {13,       1.2,     54.0},
                        {17,       4.5,    -57.0},
                        {19,       1.3,   -226.0},
                        {23,       2.7,     17.0},
                        {25,       1.2,   -149.0}
                        }
```

PI

```
PI           Name = DevName
From   = NodeName         To     = NodeName
R      = Value            X      = Value
Xc     = Value        OR  CnF    = Value
Length = Value
```

`PI` is used to represent a pi model transmission line in a single-phase equivalent circuit. `PI` uses long-line correction, however, it does not account for phase unbalance, skin effect, or the frequency dependence of the neutral return path.

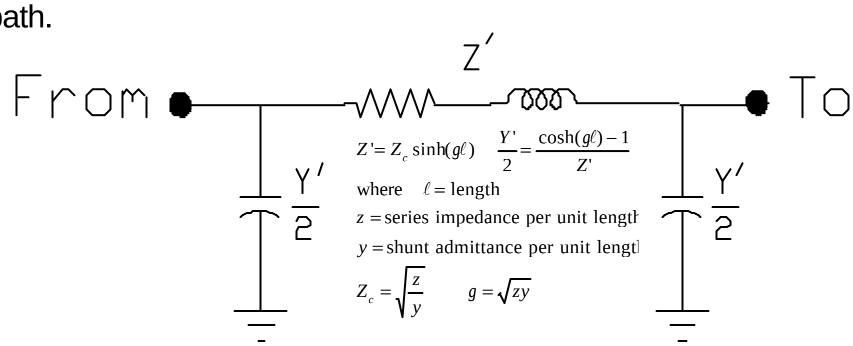


Figure 4-38: Single-Phase PI Model Equivalent Circuit

The pi circuit is characterized by the following parameters:

- Series resistance R and inductive reactance X, ohms per unit length at fundamental frequency
- Shunt capacitance CnF (nanofarads per unit length), or capacitive reactance Xc (ohms · unit length)
- Line Length, in units consistent with R, X, and CnF (Xc)

All parameters are required except for `R`, which defaults to 5% of X.

An Example

The following line has a series impedance of 0.261 + j 0.682 ohms per mile. The shunt capacitance is 17.5 nF per mile. Line length is entered in miles.

```
PI           Name = Line1
From   = Bus1         To      = Bus2
R      = 0.261        X       = 0.682
CnF    = 17.5         Length  = 7
```

PI3

```
PI3          Name = DevName
From.A = NodeName         To.A  = NodeName
From.B = NodeName         To.B  = NodeName
From.C = NodeName         To.C  = NodeName
R1     = Value            X1    = Value
Xc1    = Value        OR  CnF1  = Value
R0     = Value            X0    = Value
Xc0    = Value        OR  CnF0  = Value
Length = Value
```

`PI3` is used to represent a pi model transmission line in a three-phase circuit. Each pi circuit uses long-line correction, however, it does not account for phase unbalance, skin effect, or the frequency dependence of the neutral return path.

PI3 is characterized by the following parameters:

- Positive and zero sequence series resistance, R1 and R0, in ohms per unit length at fundamental frequency
- Positive and zero sequence series reactance, X1 and X0, in ohms per unit length at fundamental frequency
- Positive- and zero-sequence shunt capacitance CnF1 and CnF0 (nanofarads per unit length). Alternately, you can enter the capacitive reactances Xc1 and Xc0 (ohms · unit length)
- Line Length, in units consistent with R, X, and CnF (Xc)

All parameters are required except for `R1`, `R0`, and `CnF0`, which default to the following values:

$$R1 = \frac{XL1}{20}$$

$$R0 = \frac{XL0}{10}$$

$$CnF0 = 0.7\ CnF1$$

An Example

The following model represents a line with the following fundamental frequency values:

- Positive-sequence series impedance of 0.261 + j 0.682 ohms per mile
- Zero-sequence series impedance of 0.545 + j 3.012 ohms per mile
- Positive-sequence shunt capacitance of 17.5 nanofarads per mile
- Zero-sequence shunt capacitance of 6.7 nanofarads per mile

Line length is entered in miles to be consistent with these values.

```
PI3          Name = Line3
From.A = Bus1.A       To.A  = Bus2.A
From.B = Bus1.B       To.B  = Bus2.B
From.C = Bus1.C       To.C  = Bus2.C
R1     = 0.261        X1    = 0.682
CnF1   = 17.5         R0    = 0.545
X0     = 3.012        CnF0  = 6.7
Length = 7
```

SERIESFILTER

```
SERIESFILTER     Name  = DevName
CapBus     = NodeName  MidBus = NodeName
IndBus     = NodeName  kV     = Value
MVA        = Value  OR kVA    = Value
Harmonic   = Value  OR Freq   = Value
XRRatio    = Value
```

`SERIESFILTER` models a series RLC branch, as shown below. Parameters are:

- kV - Capacitor rated kV.
- MVA - Capacitor rated MVAr. Alternately, you can use parameter kVA.
- Freq is the filter's tuning (notch) frequency. Alternately, you can use parameter Harmonic, the harmonic number corresponding to the notch frequency.
- XRRatio is the XR ratio of the filter reactor. This parameter is optional, with a default value of 20.

$$X_f = \frac{X_c}{\mathrm{Harmonic}^2}$$

$$X_c = \frac{\mathrm{kV}^2}{\mathrm{MVA}}$$

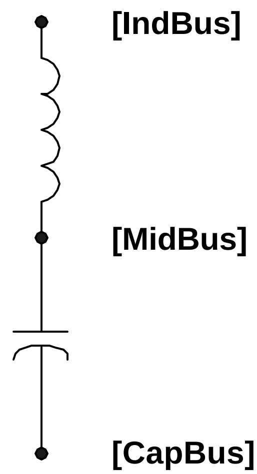


Figure 4-39: Series Filter Component

`SERIESFILTER` is a single-phase model; use three of these devices to simulate the three-phase grounded or ungrounded wye configuration. Other types of filters, such as a filter using a delta-connected capacitor can be modeled as a set of `BRANCH` and `CAPACITOR` devices.

An Example

The following devices model a filter with capacitors connected in ungrounded wye. The capacitors are rated 7.96kV, 300 kVAr per phase. The filter is tuned to the 4.7th harmonic:

```
SERIESFILTER     Name  = FltrA
CapBus     = F9N7.A    MidBus = F9N7M.A
IndBus     = F9N7.N    kV     = 7.96
Harmonic   = 4.7       kVA    = 300

SERIESFILTER     Name  = FltrA
CapBus     = F9N7.B    MidBus = F9N7M.B
IndBus     = F9N7.N    kV     = 7.96
Harmonic   = 4.7       kVA    = 300

SERIESFILTER     Name  = FltrA
CapBus     = F9N7.C    MidBus = F9N7M.C
IndBus     = F9N7.N    kV     = 7.96
Harmonic   = 4.7       kVA    = 300
```

SWITCH

Single-Phase Model

```
SWITCH       Name    = DevName
From     = NodeName  To    = NodeName
Onadmit  = Value     State = Open|Closed
Offadmit = Value
```

Three-Phase Model

```
SWITCH        Name   = DevName
From.A   = NodeName  To.A  = NodeName
From.B   = NodeName  To.B  = NodeName
From.C   = NodeName  To.C  = NodeName
Onadmit  = Value     State = Open|Closed
Offadmit = Value
```

The `SWITCH` model provides convenience in setting up a model imported from a transient program or from a one-line diagram that has switches included. This often eliminates the need to rename busses and thereby reduces the possibility of introducing errors. When the `SWITCH` is closed the admittance defaults to 1000 Siemens (0.001Ω) and when the `SWITCH` is open the admittance defaults to 0. These parameters can be set by the user with the `Onadmit` and `Offadmit` options. These values are in siemens.

An Example

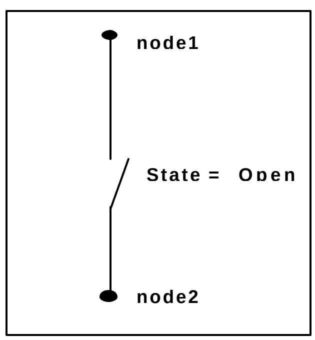


Figure 4-40: Single Phase Switch Component

```
SWITCH       Name    = sw1
From     = node1     To    = node2
                     State = Closed
```

TRANSFORMER

Single-Phase Model

```
TRANSFORMER          Name   = DevName
X.1      = NodeName X.2     = NodeName
H.1      = NodeName H.2     = NodeName
T.1      = NodeName T.2     = NodeName
MVA      = Value    %Imag   = Value  XRConstant = Yes|No
kV.X     = Value    kV.H    = Value  kV.T       = Value
MVAb.HX  = Value    %R.HX   = Value  %X.HX      = Value
MVAb.XT  = Value    %R.XT   = Value  %X.XT      = Value
MVAb.HT  = Value    %R.HT   = Value  %X.HT      = Value
```

Three-Phase Model

```
TRANSFORMER          Name   = DevName
H        = Wye|Delta X      = Wye|Delta
T        = Wye|Delta X.A    = NodeName
X.B      = NodeName  X.C    = NodeName
X.N      = NodeName  H.A    = NodeName
H.B      = NodeName  H.C    = NodeName
H.N      = NodeName  T.A    = NodeName
T.B      = NodeName  T.C    = NodeName
T.N      = NodeName
MVA      = Value     %Imag  = Value  XRConstant = Yes|No
kV.X     = Value     kV.H   = Value  kV.T       = Value
MVAb.HX  = Value     %R.HX  = Value  %X.HX      = Value
MVAb.XT  = Value     %R.XT  = Value  %X.XT      = Value
MVAb.HT  = Value     %R.HT  = Value  %X.HT      = Value
```

`TRANSFORMER` is used to model a single-phase or three-phase transformer. It can accommodate up to three windings per phase. (Use `ZMATRIX` to model transformers with four or more windings.)

Windings are designated as `H`, `X`, and `T`. For two winding transformers, all fields associated with the tertiary (T) winding should be omitted.

MVA rating, magnetizing current and frequency dependent losses

The `MVA` field specifies the H-X MVA rating of the transformer. The optional field `%Imag` (defaults to 1% of rating) represents magnetizing branch current, in percent based on the MVA rating. The model assumes that losses increase linearly with frequency- that is, transformer X/R is constant. Use the optional parameter `XRConstant = No` if losses are to be considered independent of frequency.

Winding Data

The `kV.H`, `kV.X`, and `kV.T` fields are used to specify rated voltage for each winding. Use line-to-neutral values for the single-phase model in wye, line-to-line values for the single-phase model in delta, and line-to-line values for the three-phase model.

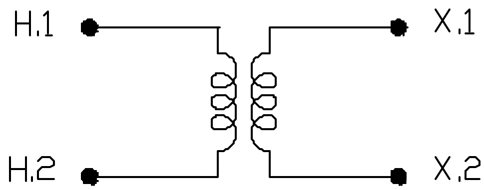


Figure 4-41: Winding Data for the Transformer Component

For single-phase transformers, node names are required only for the `H.1` and `X.1` fields. If the `H.2` is omitted, it is assumed that `H.2` is connected to ground. This default is also used for `X.2` and `T.2`.

The three-phase `TRANSFORMER` model requires that each winding be specified as either wye or delta. There are three or four terminal nodes per winding: the three phase connections (e.g., `H.A`, `H.B`, `H.C`) plus an optional `neutral` connection (e.g. `H.N`). All phase connections must be entered. The `neutral` connection may be omitted for the wye winding; this is equivalent to specifying a `connection` to ground. (a neutral connection for a delta winding should be avoided.)

Short-Circuit Data

Winding-winding short circuit resistances and reactances are entered in percent. Reactances are required; resistances are optional. If a percent resistance is omitted, the program assumes a value equal to 4% of the associated reactance. `%R` and `%X` values are assumed to be based on transformer rated MVA. Use the optional `MVAb` fields to specify different base values.

An Example

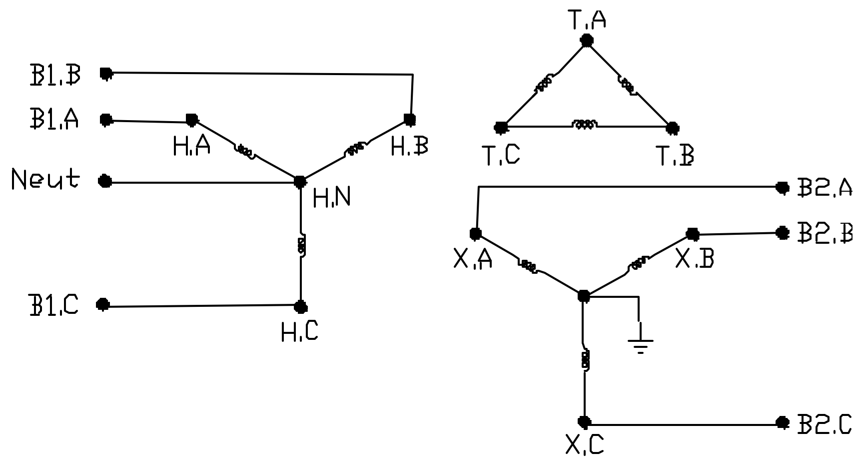


Figure 4-42: Transformer Component Example

A transformer with a 765kV wye primary winding connected to Bus B1, a 345kV grounded wye secondary winding connected to bus B2, and a 33kV floating delta tertiary winding would be specified as follows:

```
H   = Wye   kV.H= 765
X   = Wye   kV.X= 345
T   = Delta kV.T= 33
H.A = B1.A  H.B = B1.B H.C   = B1.C   H.N = Neut
X.A = B2.A  X.B = B2.B X.C   = B2.C
T.A = Ground    T.B    = TD.B T.C     = TD.C
```

Note the following:

- The floating delta tertiary winding would result in a singular matrix when the circuit is solved. To prevent this, one corner of the delta has been grounded. This is satisfactory for most situations. As an alternative, for cases where voltages at the delta are of interest, connect the delta nodes to ground through small capacitances.
- Although nothing is connected to the H winding neutral, a node name must be specified. If it were omitted, a grounded-wye connection would result. Likewise, names must be specified for the ungrounded nodes in the delta tertiary, or SuperHarm will report an error.

VSOURCE

Single Frequency Model

```
VSOURCE    Name  = DevName
Bus      = NodeName Freq    = Value
Mag      = Value    Ang     = Value
FreqMult = Value    MagMult = Value
AngShift = Value
```

Multiple Frequency Model

```
VSOURCE      Name  = DevName
Bus     = NodeName FreqMult = Value
MagMult = Value    AngShift = Value
TABLE =  { {Freq1, Mag1, Ang1 },
           {Freq2, Mag2, Ang2 },
                       .
                       .
           {FreqN, MagN, AngN }
         }
```

`VSOURCE` is a constant voltage source model. There are two variations:

- The single frequency model applies voltage of known magnitude and angle at one frequency.
- The multiple frequency model applies voltages at the specified frequencies, magnitudes, and angles.

General Comments

1. Only one source may be connected to a node.
2. A source must be connected in shunt.
3. Three-phase sources must be modeled as three single-phase devices.
4. The units for the voltage magnitude at each frequency (`Mag`) and the optional magnitude multiplier (`MagMult`) should be such that, when these quantities are multiplied, the product is the *RMS volts* to apply at that frequency.
5. Phase angles for all sources in the system model should be specified in terms of the same arbitrary reference. For example, if one of the 60 Hz voltage sources in the system is set to 0°, then the phase angles for the remaining sources are entered with respect to this 0° voltage.

The Single Frequency Model

Enter the following parameters:

- Frequency Freq in hertz
- Voltage magnitude Mag in volts. Mag may be omitted; the default value is unity
- Voltage phase angle Ang in degrees. Ang may be omitted; the default value is 0°

The optional parameters `FreqMult`, `MagMult`, and `AngShift` are intended to be used with the multiple frequency model described below. However, they can also be used with the single frequency model, if desired.

The Multiple Frequency Model

A voltage source with arbitrary harmonic content can be specified by replacing the parameters `Freq`, `Mag`, and `Ang` with a table listing frequency, magnitude, and angle for each harmonic voltage applied by the source. Unlike the single frequency model, magnitudes and phase angles *must* be specified.

Three optional parameters are provided to simplify entry of table data:

- FreqMult - if given, frequencies are per unit of this value
- MagMult - if given, magnitudes are per unit of this value
- AngShift is used to time-shift the current waveform; a positive value corresponding to a forward shift. Phase angles are corrected as follows:

$$q_h = q_1 + h \cdot AngShift\ , \quad 1 \le h \le N$$

An Example

Three `VSOURCE` models are used to represent a three-phase, positive sequence 60 kV line-line voltage source. Magnitudes are entered as line-neutral voltages. Ang is not specified for the phase A source, so it defaults to 0°.

```
VSOURCE Name = VsA           Bus = VSBus.A
       Freq = 60   Mag = 346410
VSOURCE Name = VsB           Bus = VSBus.B
       Freq = 60   Mag = 346410 Ang = -120
VSOURCE Name = VsC           Bus = VSBus.C
       Freq = 60   Mag = 346410 Ang =  120
```

ZYCMATRIX

```
ZYCMATRIX    Name  = DevName
From      = NodeNameList To    =  NodeNameList
Length    = Value
YCMult    = Value        ZMult =  Value
ZRMATRIX= { {Value, Value,  … Value },
            {Value, Value,  … Value },
                .       .     .    .
                .       .     .    .
            {Value, Value,  … Value }
          }
ZXMATRIX= { {Value, Value,  … Value },
            {Value, Value,  … Value },
                .       .     .    .
                .       .     .    .
            {Value, Value,  … Value }
          }
YCMATRIX= { {Value, Value,  … Value },
            {Value, Value,  … Value },
                .       .     .    .
                .       .     .    .
            {Value, Value,  … Value }
          }
```

`ZYCMATRIX` allows representation of any number of mutually coupled branches. The branches are specified by listing the pairs of connecting nodes in the `BRANCHES` list. Branch self and mutual impedances are then entered as R (`ZRMATRIX`) and X (`ZXMATRIX`) matrices, where:

`rii`, `xii` are the resistance and inductive reactance of branch i,

`rij`, `xij` are the real and reactive components of the coupling impedance from branch i to branch j.

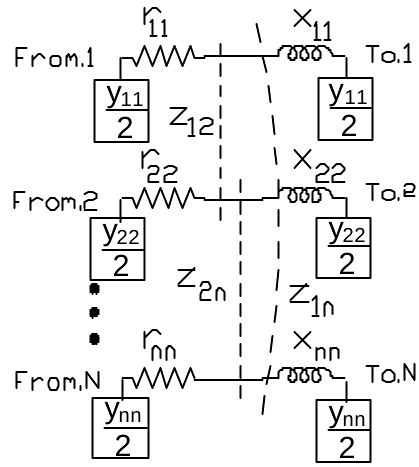


Figure 4-43: ZYCMATRIX Component

Reactance values should be entered at fundamental frequency. They may be inductive (X > 0) or capacitive (X < 0). All components of the [R] matrix, on the other hand, must be non-negative.

Zero values are acceptable for coupling impedances, but not for self impedances. That is, for branches i and j, r_{ij}, x_{ij} may both be zero; but r_{ii} and x_{ii} cannot both be zero.

The admittance values are entered using `YCMATRIX`, where:

`yii`	the admittance of branch i,
`yij`	the admittance from branch i to branch j.

Per unitized values may be entered for [R], [X] and [Y] by using the optional `Zmult` and/or `Ymult` parameters. Both of these tags default to 1.0. The program uses the product of the given matrix and the associated multiplier, if any. The `LENGTH` tag is a multiplier for the overall model of [R], [X], and [Y] are entered as per unit length quantities. The default for this tag is 1.0.

An Example

The following example indicates the usage of this device model as well as the equivalent using `BRANCH3` and discrete `CAPACITOR`s (shown in Figure 4-16). The system modeled is a three-phase system with the parameters from the source to the first bus being described with the following `ZYCMATRIX` device model (Figure 4-15).

```
ZYCMATRIX NAME = ZMATRX LENGTH = 6
         ZMULT = 3.0    YCMULT = @"1 6 /"
          FROM = {SRCA, SRCB, SRCC}
            TO = {BUS1A, BUS1B, BUS1C}

!  Ohms/Mile

ZRMATRIX={
         {0.2493, 0.052, 0.052},
         {0.052, 0.2493, 0.052},
         {0.052, 0.052, 0.2493}
         }
ZXMATRIX={
         {0.3987, 0.1443, 0.1443},
         {0.1443, 0.3987, 0.1443},
         {0.1443, 0.1443, 0.3987}
         }

!  Siemens/Mile

YCMATRIX= {
          {5.655e-6, 0, 0},
          {0, 5.655e-6, 0},
          {0, 0, 5.655e-6}
          }
```

Figure 4-44: ZYCMATRIX Component Example

```
BRANCH3 NAME=BRANCH
        FROM.A=SRCA   TO.A=BUS1A
        FROM.B=SRCB   TO.B=BUS1B
        FROM.C=SRCC   TO.C=BUS1C
        LENGTH=6
        R0=1.061         X0=2.061
        R1=0.592         X1=0.763

CAPACITOR NAME=CAP1 FROM=SRCA  UF=0.0075
CAPACITOR NAME=CAP2 FROM=SRCB  UF=0.0075
CAPACITOR NAME=CAP3 FROM=SRCC  UF=0.0075
CAPACITOR NAME=CAP4 FROM=BUS1A UF=0.0075
CAPACITOR NAME=CAP5 FROM=BUS1B UF=0.0075
CAPACITOR NAME=CAP6 FROM=BUS1C UF=0.0075
```

Figure 4-45: ZYCMATRIX Equivalent

The R, X and Y values used in the `ZYCMATRIX` model are phase component values. To incorporate them in the `BRANCH3` model it is necessary to convert them to sequence values. Once this is done they can be simply modeled as branches. The calculated capacitance needs to be divided by two and set at the end of each node to ground.

Page Left Intentionally Blank

CHAPTER 5

DIRECTIVES

This chapter provides a description of the commands and program directives in SuperHarm. Each of these directives has unique applications in accomplishing specific goals and objectives.

In This Chapter

Page Intentionally Left Blank

Block and Line Comments

Block and line comments can be included in a SuperHarm data file to provide supplemental documentation of component and directive usage. Comments are designated by using the "//", "!", or "/*...*/" directives:

```
// The double slash causes SuperHarm to ignore all
// characters between the double slash and the end
// of the line.  The double slash can be placed
// anywhere on a line.
```

```
! The exclamation point is like the double slash,
! but it must be placed in column 1.  Since
! SuperHarm does not have to search the entire
! line, it can process ! comments much faster
! than // comments.
```

```
/*
Two or more lines are considered a comment if they
start with a slash-asterisk and end with an aster-
isk-slash.  These characters must be placed in col-
umns 1 and 2, as shown here.
*/
```

The following example shows a block comment with a blank first line, and with each subsequent line beginning with a double asterisk. Though not required, this makes the data file easier to read.

```
/*
** SuperHarm assumes that there is no data between
** the block comment end and the end of line.  So,
** BRANCH Wrong is ignored; BRANCH Right is OK.
** BRANCH Name = Wrong    From = X To Y R = 1
*/
BRANCH     Name = Right   From = Y    To Z   R = 1
```

The block comment block is particularly useful for temporarily hiding circuit data from SuperHarm when debugging a data file:

```
/*
BRANCH Name = HideThis    From = X   To = Y  R = 1
*/
```

#DEFINE and #UNDEF

```
#DEFINE  Identifier   Text_String
```

```
#UNDEF   Identifier
```

`#DEFINE` is used to associate a text string with an identifier. Once defined, you can use the identifier in place of the text string as a parameter value (anything to the right of an equal sign). SuperHarm will make the substitution as long as the identifier is used *after* the `#DEFINE` directive and *before* the corresponding `#UNDEF` (undefine) directive, if any.

The first character of the identifier must be alpha (A…Z, a…z), with each remaining character either alphanumeric or an underscore. The length of the text string is limited to 255 characters.

`#DEFINE` can be used as a convenience (substituting short identifiers for long text strings), and to increase the readability of the circuit description. For example, suppose that you are modeling a 480 volt system using a single-phase equivalent circuit. This requires that you use line-to-neutral voltage in the circuit data. Rather than entering this value repeatedly, you could place the following directive at the beginning of the SHA file:

```
#DEFINE LN480V  277.1281
```

Now, whenever you need to use the value, you can use the label instead:

```
VSOURCE       Name =VS1
Bus     = 480VBus    Freq = 60
Mag     = LN480V     Ang  = 0
```

Note that in this example, the text string is a *number*. Should SuperHarm flag this as an error? No, because before the input processor sees the circuit description, a preprocessor checks for `#DEFINE` directives, then makes the appropriate substitutions. To the preprocessor, everything is text.

The preprocessor does not enforce the standard rules for entering text strings. Consider the following statements:

```
#DEFINE MyName This name is illegal

BRANCH  Name = MyName
From = X  To = Y  R =1
```

The preprocessor treats everything after `MyName` as the substitute text, and makes the following substitution:

```
BRANCH  Name = This name is illegal
From = X  To = Y  R =1
```

This is flagged as an error by the input processor. We bring this point up because incorrect use of `#DEFINE` can lead to obscure error messages, since the input processor does not check the circuit description until all substitutions have been made.

Header Files

The `#DEFINE` and `#INCLUDE` directives can be combined to a create a *header file* which lists a set of standard definitions. An example header file, LNVolt.SHI, is shown below. This file contains `#DEFINE`s for the line-neutral equivalents of all standard U.S. line-line voltages. You can use these `#DEFINE`s if you place the following directive at the top of the data file:

```
#INCLUDE "LNVolt.SHI"
```

```
!*** LNVolt.SHI ***
! Line-to-neutral values for standard U.S. voltages

#DEFINE LN480V             277.1281
#DEFINE LN600V             346.4102
#DEFINE LN2400V           1385.6406
                  •
                  •
                  •
#DEFINE LN500000V 288675.1346
#DEFINE LN765000V 441672.9559
```

See page 5-9 for a description of the `#INCLUDE` directive.

Using #DEFINE with Inline Math

Text_String may contain an inline math (page 5-15) expression. For example:

```
#DEFINE LN480V @"480  3   SQRT /"
```

Global Variable - BATCH_ID

BATCH_ID is a global variable that starts at 1 when SuperHarm is in batch solve mode and at 0 if SuperHarm is in normal solve mode. If four files were selected for a batch run then when the first file is running the BATCH_ID is 1. The second file running sets the BATCH_ID flag to 2, and so on until the fourth file sets the flag to 4. This variable allows for the implementation of sophisticated techniques to simplify certain analysis procedures. The following statement illustrates the usage of this global variable:

```
#DEFINE CASE_ID BATCH_ID
```

The above line simply defines a defined variable, CASE_ID, as the value indicated by the BATCH_ID variable.

```
#DEFINE CASE_ID @"BATCH_ID 1 -"
```

The above statement indicates, as seen in the previous section, that BATCH_ID can be utilized with inline math as well.

DISCARD and RETAIN

```
DISCARD

Currents = Yes|No  or CURRENTLIST  = {DevName, DevName, ...}
Voltages = Yes|No  or VOLTAGELIST  = {NodeName, NodeName, ...}
```

```
RETAIN

Currents  = Yes|No  or CURRENTLIST = {DevName, DevName, ...}
Voltages  = Yes|No  or VOLTAGELIST = {NodeName, NodeName, ...}
```

By default, SuperHarm saves voltages and discards currents when solving a circuit description. This directive will save currents as well as voltages:

```
RETAIN Currents = Yes
```

For larger systems, you should save only quantities of interest. Doing so decreases solution time and the disk space needed to store the output file. A circuit description can contain any number of RETAIN and DISCARD directives. SuperHarm processes the directives in the order that they are listed, adding and deleting quantities from its list of quantities to save.

End of Case Flag

SuperHarm SHA files must end with an ellipsis - four periods without spaces between periods. The ellipsis flags the end of data for the case. It is *not* an end of file designator and should not be placed at the end of include (SHI) files. (If SuperHarm finds an ellipsis in an include file, it stops reading data for the case, without returning to the SHA file.) Any characters placed after the ellipsis are ignored, so you can place comments or hide data fragments at the end of the SHA file:

```
TITLE Title1="Ellipsis"

! Circuit Data

BRANCH Name = Last          From = X    To = Y R =
1

....

SuperHarm can not see this:

BRANCH Name = New From = Y       To = Z     R = 1
```

#IFDEF

```
#IFDEF   Identifier
```

```
#ENDIF
```

The `#IFDEF` directive provides a conditional expression that evaluates True/False depending on whether the identifier was previously created using the `#DEFINE` directive. Each `#IFDEF` directive must have a matching `#ENDIF`. Nesting of `#IFDEF` statements is not permitted.

For example, the following directive creates an identifier `INC` file and assigns it a value of 1. This assignment results in the `#INCLUDE` directive, within the `#IFDEF` / `#ENDIF` conditional statement, being executed.

```
#DEFINE INCFILE 1

#IFDEF INCFILE
   #INCLUDE include.shi     2.0, 10.0
#ENDIF
```

#IF and #ELSEIF

```
#IF      condition
```

```
#ELSEIF  condition
```

```
#ENDIF
```

The `#IF/#ELSEIF` directive provides a conditional expression that evaluates True or False depending on whether the condition has been satisfied. Note that the `#ELSEIF` clause is optional and that each `#IF` directive must have a matching `#ENDIF`. Nesting of `#IFDEF` statements is not permitted. The conditional expression for the `#IF/#ELSEIF` directive is created using the following rules:

condition → `(parameter1  operator  parameter2)`

where:

parameter1 and parameter2 must evaluate to integers and be ≥ 0

operator must be one of the following:

>	greater than
<	less than
=	equal (also ==)
!=	not equal to

For example, the following data selects a motor model based on the `MOTORTYPE` assignment.

```
#DEFINE MOTORTYPE   2


#IF (MOTORTYPE = 1)
   INDUCTIONMOTOR NAME=MOTOR HP=500
   %PF=80.0 %EFF=90.0 KV=4.160
   CONNECTION=DELTA
   BUS.A=BUS34.A  BUS.B=BUS34.B  BUS.C=BUS34.C
#ELSEIF (MOTORTYPE = 2)
   MACHINE NAME=MOTOR HP=500
   DF=0.80 %EFF=90.0 KV=4.160
   CONNECTION=DELTA
   BUS.A=BUS34.A  BUS.B=BUS34.B  BUS.C=BUS34.C
#ENDIF
```

#INCLUDE

Simple INCLUDE

```
#INCLUDE FileSpec
```

INCLUDE with parameter passing

```
#INCLUDE FileSpec (Param_1, Param_2, …)
```

`#INCLUDE` causes SuperHarm to switch to the file whose name is given by `FileSpec`. The include file is read in its entirety, then SuperHarm resumes reading the SHA file at the first line after the `#INCLUDE` keyword. Include files should have the file extension SHI, which should be included in `FileSpec`. The file name can be simply typed in or included in quotation marks. If the include file is not in the same directory as the file using the `#INCLUDE` directive, add the file path to `FileSpec`. Alternately, you can use the `INCPATH` tag in the `Options` directive to specify the directory that include files are to be read from. For example:

```
#INCLUDE "C:\SuperHarm32\Directives\Voltages.shi”
```

is equivalent to:

```
OPTIONS
INCPATH = "C:\SuperHarm32\Directives\"
#INCLUDE Voltages.shi
```

TIP: File and path names should be quoted in directives, especially if they include embedded blanks.

The usefulness of the `#INCLUDE` directive may be enhanced by passing parameters from the data file (SHA) to the include file (SHI) using the following guidelines:

- In the include file, instead of entering a value for a parameter that you wish to vary, enter a place holder of the form *%n,* where *n* indicates that this parameter is n^{th} in a list of parameters added to the `INCLUDE` directive for this file.

The process and an example are illustrated in Figure 5-1.

MAIN FILE

```
#INCLUDE    "x.shi"  (B1,        N1,      N2,       0.1,      0.3)
```

INCLUDE FILE (x.shi)

```
BRANCH                  Name=%1   From=%2   To=%3      R=%4      X=%5
```

RESULT

```
BRANCH                  Name=B1   From=N1   To=N2     R=0.1    X=0.3
```

Figure 5-46: #INCLUDE Parameter Passing

This example shows that both numbers and text can be passed to an include file. However, there are two restrictions with regard to passing text:

1. A text parameter cannot use quotation marks or spaces.
2. If a text parameter is to be used as a root name to which a suffix will be added, the concatenation character ^ must be inserted between the %n place holder and the suffix.

In the example of Figure 5-1, the text parameters `B1`, `N1`, and `N2` represent complete node and device names in the result:

```
BRANCH Name = B1    From = N1     To = N2
```

That is, there is *white space to the right* of the name. Therefore, the concatenation character is not necessary

```
BRANCH Name = %1    From = %2     To = %3
```

But, suppose that you wish to generate three-phase node and device names from the passed parameters `B1`, `N1`, and `N2`:

```
BRANCH Name = B1A    From = N1.A   To = N2.A
BRANCH Name = B1B    From = N1.B   To = N2.B
BRANCH Name = B1C    From = N1.C   To = N2.C
```

Then you are confronted with restriction #2. You must use ^ to concatenate the passed parameters and the phase identifiers:

```
BRANCH Name = %1^A   From = %2^.A  To = %3^.A
BRANCH Name = %1^B   From = %2^.B  To = %3^.B
BRANCH Name = %1^C   From = %2^.C  To = %3^.C
```

Device Modules

Device modules create the ability to pass parameters to transform simple include files into custom device modules. Unlike a simple include file, a device module can:

1. ***Be applied at any bus on the system.*** To achieve this, you must never refer to an external node (a node that is used both in the module and the main file) by name in the module. External nodes should be passed as parameters, and referenced as *%n* (single-phase node) or *%n*^.A (three-phase node) in the file.

2. ***Be applied more than once on the same system.*** This requires that you adopt a method for generating unique names for devices and nodes that are internal (referenced only in the module). There are three ways to do this:

   - Pass these names as parameters, as is done for external nodes. However, only a few names can be passed before the 80 character limit on the #INCLUDE directive is reached.

   - Have SuperHarm create node and device names automatically. This is done by using the auto-increment (#) character to represent a node or device name. SuperHarm replaces the first occurrence of # in the data file with the number 0, the second # with 1, and so on. You can also use the # with the concatenation character, as shown below:

DATA FILE			RESULT		
BRANCH	Name=#	...	BRANCH	Name=0	...
CAPACITOR	Name=#	...	CAPACITOR	Name=1	...
BRANCH	Name=Br^#	...	BRANCH	Name=Br2	...
VSOURCE	Name=Vs^#	...	VSOURCE	Name=Vs3	...

Figure 5-47: Auto-Increment Character Usage

If you wish to look at device currents in TOP, The Output Processor, you must, of course, know the device name. But, although you can determine the name that replaces each occurrence of #, this is usually not worth the effort. If you wish to view current for a device, use an instance number parameter, as described below, instead of #.

Note that using # to the right of an equal sign makes it impossible to use this character to generate node names. This is because a node name will always be used at least twice in a circuit description, whereas the # character generates a unique number at each occurrence. Use the #DEFINE directive if you wish to use auto-incrementing node names:

DATA FILE

```
#DEFINE          N1 #
#DEFINE          N2 #
BRANCH3          Name=#
   From.A=N1^.A  To.A=N2^.A   ...
```

RESULT

```
BRANCH3          Name=3
From.A=1.A       To.A=2.A     ...
```

Figure 5-48: Auto-Increment using #DEFINE Directive

– Pass an "instance number" parameter. The module uses this number to create unique names, as shown in the example of Figure 5-3. With an instance number, successive #INCLUDEs of the same module would be of the form:

```
!                          Ext   Inst  Other
!                          Node   Num  parms

#INCLUDE mod.shi (        BusX,    1,  …)

#INCLUDE mod.shi (        BusY,    2,  …)

#INCLUDE mod.shi (        BusZ,    3,  …)
```

•

3. ***Allow variable device parameters.***

Mathematically Manipulating *%n* Parameters

You can create even more powerful device modules by using inline math to mathematically manipulate *%n* parameters. For example:

```
! Lines and Cables          (length * 1000')
!             type       from     to    length
#INCLUDE "CA00010.SHI" (BUS41,  BUS42, 4.8020)

[CA00010.SHI]
! Parameters: %1 - From Node      %2 - To Node
!             %3 - Length in 1000'
BRANCH        Name=%1^_^%2   From=%1   To=%2
              Length=%3   R=.0586   X=.0227
CAPACITOR     Name=C^%1     From=%1   uF=@"%3 .0195
*"
CAPACITOR     Name=C^%2     From=%2   uF=@"%3 .0195
*"
```

Nested #INCLUDEs

An include file may itself use an `#INCLUDE` directive to switch SuperHarm to yet another include file, as shown below:

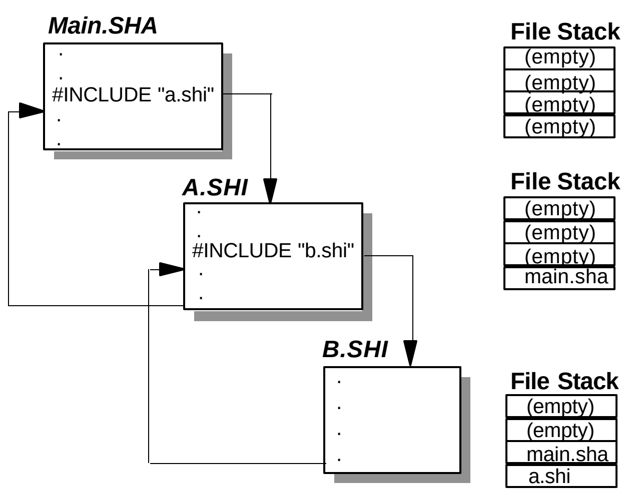


Figure 5-49: Nested Include Files

Before switching to a new include file, SuperHarm pushes the name of the current file onto the file stack. It pops this name off the stack when it returns from the include file. Note that the file stack has room for only four file names, which means that although you can use any number of `#INCLUDE` directives in the circuit description, *you must not nest include files more than five layers deep.*

Inline Math

The inline math capability allows the user to substitute a mathematical expression in place of a numeric value.

At any place in the data file where SuperHarm expects a number (that is, to the right of an equal sign) you can substitute a mathematical expression of the form @"expression". For example, suppose you are constructing a single-phase equivalent of a 480v (line-to-line) system.

A `VSOURCE` for this system should be entered with the corresponding line-neutral RMS magnitude. Rather than entering the magnitude directly:

```
VSOURCE  Name = 480vSrc  Bus = 480VBus  Freq = 60
         Mag  = 277.1281
```

you can let SuperHarm perform the calculation:

```
VSOURCE  Name = 480vSrc  Bus = 480VBus  Freq = 60
         Mag  = @"480 3 SQRT /"
```

Reverse Polish Notation

The syntax of the @ expression given above is known as Reverse Polish Notation (RPN). In the 1920's, Polish mathematician Jan Lukasiewicz (Wu-ka-SHAVE-itch) developed a formal logic system that allowed mathematical expressions to be specified without parentheses by placing the operators before (prefix notation) or after (postfix notation) the operands. For example the (infix notation) expression

    (4 + 5) × 6

could be expressed in prefix notation as

    × 6 + 4 5   or   × + 4 5 6

and could be expressed in postfix notation as

    4 5 + 6 ×   or   6 4 5 + ×

Prefix notation also came to be known as Polish Notation in honor of Lukasiewicz. Hewlett-Packard adjusted the postfix notation for a calculator keyboard, added a stack to hold the operands and functions to reorder the stack. HP dubbed the result Reverse Polish Notation (RPN) also in honor of Lukasiewicz.

With RPN, the mathematical operator is placed after the arguments, instead of between arguments, as is done in algebraic notation. Several examples include

Algebraic	RPN
$5 \cdot 5$	5 5 *
$\frac{480}{\sqrt{3}}$	480 3 SQRT /
$0.00517\left(\frac{4.16^2}{100}\right)$	0.00517 4.16 2 ^ 100 / *
$16.92 + 5\left(57.32\cos^{-1}(0.839)\right)$	0.839 Acos 57.32 * 5 * 16.92 +

The last example looks intimidating until one understands the concept of the *RPN stack.* SuperHarm reads the expression from left to right. If it decodes a number, it pushes that number onto the bottom of the stack, lifting each number already on the stack into the next higher storage location. If it decodes an operation, it pops the bottom one or two numbers (depending on the number of arguments required for the operation) off the stack, performs the operation, and pushes the result back onto the bottom of the stack. The process is illustrated in figure below:

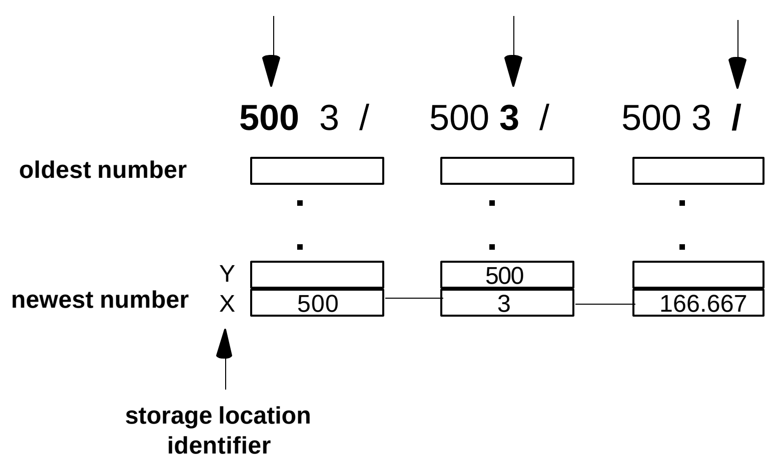


Figure 5-50: RPN Stack Concept

Unlike the typical RPN calculator, SuperHarm's RPN stack is not limited to four memory locations. You should never encounter a case where you overflow the stack.

Operations

The current version of SuperHarm supports the following operations. In the description, X and Y refer to the two most recent numbers in the RPN stack, as illustrated in Figure 5-5.

Operator	Description	Operator	Description
+	y+x	*	y*x
-	y-x	/	y/x
sqrt	$\sqrt{x}$	pow	y^x
cos	Cos x	acos	Arccos x
sin	Sin x	asin	Arcsin x
tan	Tan x	atan	Arctan x
atan2	Arctan (y/x)	<	x < y
>	x > y	=	x = y

Notes:

1. X and Y may be an integer or real number for any of these functions
2. The trigonometric functions return angle in degrees
3. The inverse trigonometric functions require X to be in degrees
4. Spaces are required between operators and data.

Additional operators are available for inline math. These are listed and described below:

and, xor, or, not

These are logical functions that return either a 1 or 0. The truth tables for these functions are reviewed below. Note that in the tables below, true refers to 1 and false refers to 0.

AND Truth Table	
True AND True	True
True AND False	False
False AND True	False
False AND False	False

OR Truth Table	
True OR True	True
True OR False	True
False OR True	True
False OR False	False

XOR Truth Table	
True XOR True	False
True XOR False	True
False XOR True	True
False XOR False	False

NOT Truth Table	
NOT True	False
NOT False	True

andb, orb, notb

These are binary equivalents of the logical functions. The binary versions perform the bitwise numerical operation on 16 bit words. Each bitwise logical operator performs a bit-by-bit operation on the data. In bitwise truth tables, the 1 and 0 should replace the TRUE and FALSE.

mod, fmod

The modulus operator (mod) returns the remainder from an integer division operation. Both x and y need to be integer numbers. The operator, fmod, returns the floating point remainder from a floating point function division operation. When using this operation, keep in mind that y cannot be zero.

bit

The bit operator returns the status at a particular position in a binary digit. An example of the usage of this operator can be seen below.

```
#define SW1 @"Batch_ID 0 bit"
#define SW2 @"Batch_ID 1 bit"
```

The above statement converts the `BATCH_ID` global variable into binary flags. The variable `SW1` is the state of the 0th bit of the binary flag (`BATCH_ID`). If the 0 position flag is set then the bit operator returns a 1. If the 0 position flag is not set then the bit operator returns a 0. The variable `SW2` is the state of the 1st bit of the binary flag.

Using *%n* Parameters with Inline Math

Parameters passed to an include file may be mathematically manipulated. As an example, consider a three-phase voltage source module which takes three parameters:

%1 Three-phase bus name, e.g. BusX The module constructs individual node names and `VSOURCE` device names from this parameter. (e.g., BusX.A and BusXA for the phase A node and device, respectively).

%2 $|V_{LL}|$, which must be converted to $|V_{LN}|$, as required by the `VSOURCE` model.

%3 θ_A, which is used to compute θ_B and θ_C.

```
VSOURCE Name = %1^A          Bus = %1^.A Freq =
60
        Mag = @"%2 3 Sqrt /"
        Ang = %3
VSOURCE Name = %1^B          Bus = %1^.B Freq =
60
        Mag = @"%2 3 Sqrt /"
        Ang = @"%3 120 -"
VSOURCE Name = %1^C          Bus = %1^.C Freq =
60
        Mag = @"%2 3 Sqrt /"
        Ang = @"%3 120 +"
```

#LIBRARY

Simple LIBRARY

```
#LIBRARY FileSpec ModuleName
```

LIBRARY with parameter passing

```
#LIBRARY FileSpec ModuleName (Param_1, Param_2, …)
```

TIP: File and path names should be quoted in directives, especially if they include embedded blanks.

The `#LIBRARY` directive can be used in place of several `#INCLUDE` files. `#LIBRARY` causes SuperHarm to switch to the file whose name is given by `FileSpec`. The library file is only read for the specified module, then SuperHarm resumes reading the SHA file at the first line after the `#LIBRARY` keyword.

Library files should have the file extension SHL, which should be included in `FileSpec`. If the library file is not in the same directory as the file using the `#LIBRARY` directive, add the file path to `FileSpec`, e.g., "c:\etkprog\shlib\myfile.shl."

Alternately, you can use the LIBPATH tag in the `Options` directive to specify the directory that library files are to be read from.

The following example illustrates the usage of the `#LIBRARY` directive. Figure 5-6 shows a portion of the SuperHarm data file, system.sha. In this file the `#LIBRARY` directive is used to pass specific parameters to two different linear load representations (`5L144 & 5L242`).

Figure 5-7 shows a portion of the library file, esso.shl. Note that the module names must be enclosed in square brackets. SuperHarm will only read the data code from the specified module name to the next module name.

```
!
!  Linear Loads
!
!  Parameters:
!       1 - Load Name
!       2 - Bus Name (without phase identifier)
!       3 - Line-Line Bus Voltage at load
!       4 - % of peak load
!
#library ESSO.SHL 5L144(LD144A, M, kVbLL,  40.0)
#library ESSO.SHL 5L144(LD144B, K, kVbLL,  40.0)
#library ESSO.SHL 5L144(LD144C, D, kVbLL,  20.0)
#library ESSO.SHL 5L242(LD242,  C, kVbLL, 100.0)
```

Figure 5-51: SuperHarm Data File [system.sha]

```
!
!  Load Helpers
!
[5L144]
LINEARLOAD NAME=%1 KV=%3 KVA=@"13.000 %4 *" DF=0.65
%PARALLEL=20.0
  %SERIES=80.0
  BUS.A=%2^_A
  BUS.B=%2^_B
  BUS.C=%2^_C

[5L242]
LINEARLOAD NAME=%1 KV=%3 KVA=@"14.400 %4 *" DF=0.97
%PARALLEL=50.0
  %SERIES=50.0
  BUS.A=%2^_A
  BUS.B=%2^_B
  BUS.C=%2^_C
```

Figure 5-52: SuperHarm Library File [esso.shl]

OPTIONS

```
OPTIONS

FBASE = Value

IGNOREISLANDS =

Yes|No

OPTIMALORDER = Yes|No

INCPATH = PathName

LIBPATH = PathName

OUTFILE = FileSpec

OLDMOTOR = Yes|No
```

FBASE
This parameter specifies the frequency base for inductive and capacitive reactances given in the circuit description. 60 Hz is assumed if `FBASE` is omitted.

```
[60Hz System]
OPTIONS
FBASE = 60

[50Hz System]
OPTIONS
FBASE = 50
```

IGNOREISLANDS
Upon reading all system data, the input processor checks for the existence of subnetworks, which normally indicate an error. However, it may be convenient to disable this check during initial building of a multiphase circuit, before interphase coupling is included. To have SuperHarm ignore subnetworks, set `IGNOREISLANDS = Yes.` The default is `IGNOREISLANDS = No`.

OPTIMALORDER
This option controls whether optimal ordering of the matrix takes place to minimize the number of fill-ins in the sparse matrix during factorization. This is often times useful in tracking down solution roundoff errors associated with

ill-conditioned (physically unrealizable) systems. If the answers change depending on the state of this tag, an ill-conditioned system is probably the cause. The default is `OPTIMALORDER = YES`.

INCPATH

Users may specify the path name that points to the directory where all files specified using the `#INCLUDE` directive are to be found. This option eliminates the redundant specification of the complete path for the include files in a datafile when being called upon by the `#INCLUDE` directive. The path name should start with the drive letter and end with a backslash (\). The default for this tag is the directory where the datafile is found.

TIP: File and path names should be quoted in directives, especially if they include embedded blanks.

```
OPTIONS
INCPATH = "C:\SuperHarm32\Include Files\"
```

LIBPATH

Users may specify the path name that points to the directory where all files specified using the `#LIBRARY` directive are to be found. This option eliminates the redundant specification of the complete path for the library files in a datafile when being called upon by the `#LIBRARY` directive. The path name should start with the drive letter and end with a backslash (\). The default for this tag is the directory where the datafile is found.

TIP: File and path names should be quoted in directives, especially if they include embedded blanks.

```
OPTIONS
LIBPATH = "C:\SuperHarm32\Library Files\"
```

OUTFILE

Users may specify the output file with this option. If nothing is indicated SuperHarm automatically creates an output file with the same name as the input datafile with the extension SHO. The `OUTFILE` tag can only contain a new root (no extension) file name. No drive or directory path can be specified.

```
OPTIONS
```

```
OUTFILE = Case1a
```

`OUTFILE` can also be utilized to facilitate batch case solutions. This is illustrated in the following example.

An OUTFILE Example

In this example the batch solve mode will be utilized to run a single case a specific number of times. To insure that the output file generated is not continually written over, concatenation (^) is utilized to append the output file name with the case number being ran. The statement below insures that a new output file name is created for each case ran.

```
OPTIONS OUTFILE = CASE^BATCH_ID
```

Therefore, when the datafile TEST.SHA is selected for batch solve mode four times, the above statement allows four different output files to be created called `CASE1.SHO, CASE2.SHO, CASE3.SHO` and `CASE4.SHO`.

OLDMOTOR

A flag to instruct SuperHarm on which method to utilize for the calculation of induction motor parameters. Refer to the induction motor component section of Chapter 3 for additional details regarding this option. The default is `OLDMOTOR = No`.

SCAN

Single Source Scan

```
SCAN         Name  = DevName
Bus      = NodeName Ang  =  Value
FMin     = Value    Fmax =  Value
FInc     = Value    Type =  Volt-
                            age|Current
```

Multiple Source Scan

```
SCAN           Name  = DevName
FMin    = Value    FMax  =  Value
FInc    = Value    Type  =  Volt-
                            age|Current
BUSLIST = {NodeName, NodeName, ...}
ANGLIST = {Value, Value, ...}
```

The `SCAN` directive applies one or more constant current or voltage sources to the system, each injecting unity amp or volt at the frequencies FMin, FMin + FInc, FMin + 2 FInc, … Fmax. The user can specify whether or not they would like to perform a `current` or a `voltage` scan by using the tag, `TYPE`. The default for this tag is `current`.

This directive causes SuperHarm to kill all sources in the system (`VSOURCE`, `ISOURCE`, and `NONLINEARLOAD`). All `VSOURCE` sources are shorted. All `NONLINEARLOAD` and `ISOURCE` sources are open circuited. Therefore, you *do not* have to remove all sources and ground voltage source nodes when using `SCAN`. (But you *do* have to do these things if you use `ISOURCE` or `VSOURCE` to produce a scan.)

If only one source is applied, use `Bus` to specify the node to which the source is connected, and `Ang` to specify the phase angle of the injected current, in degrees, with respect to the system reference. Use `BUSLIST` and `ANGLIST` for multiple source scans. The following example shows the correct syntax for positive, negative, and zero sequence current scans:

```
SCAN    Name = PSeq // Pos. seq. scan at ASD1
        FMin = 60 FMax = 1200 FInc = 10
        BUSLIST = {ASD1.A, ASD1.B, ASD1.C}
        ANGLIST = {0, 240, 120}

SCAN    Name = NSeq // Neg. seq. scan at ASD2
        FMin = 60 FMax = 1200 FInc = 10
        BUSLIST = {ASD2.A, ASD2.B, ASD2.C}
        ANGLIST = {0, 120, 240}

SCAN    Name = ZSeq // Zero seq. scan at ASD3
        FMin = 60 FMax = 1200 FInc = 10
        BUSLIST = {ASD3.A, ASD3.B, ASD3.C}
       ANGLIST = {0, 0, 0}
```

The sources applied to the system by a `SCAN` directive are treated as devices and as source groups. Because they are treated as devices, you can view the injected current just as if `ISOURCE` had been used. Because they are treated as source groups, you can use more than one `SCAN` directive. For example, the directives given above create three groups (PSeq, NSeq, ZSeq) of three sources each. When producing frequency scans of this system in TOP, you can apply any combination of these groups.

See the `SOURCEGROUP` directive (page 5-26) for more information on the *concept* of source groups. But note that in the case of frequency scans, `SOURCEGROUP` is not needed - the `SCAN` directive creates the source groups.

Voltage Scan

It should be noted that voltage scans are seldom done. When performing a voltage scan, voltage sources are shorted and current sources are opened (as were done with current scans). Before doing this type of a scan, the user should ensure that this is what they want to do. Sometimes it might be desirable for the user to manually comment out other voltage sources depending on why the voltage scan is being done.

A voltage scan is required if one is interested in investigating the impact of harmonic voltage levels on the utility system to current flow in filters. We often assume a value, but a voltage scan allows the engineer to calculate the amps per volt transfer function between a point in the utility supply and a filter or other customer device.

Voltage scans are also done when studying transfer functions in systems that have harmonic components due to arcing sources. An arc is best modeled as a harmonic voltage source (nearly a square wave) behind an impedance. Voltage scans are also useful for understanding harmonic voltage propagation on the network and general power system transfer function characteristics.

SOURCEGROUP

```
SOURCEGROUP
SOURCELIST = {DevName, DevName, …}
```

This option allows you to specify a set of one or more `ISOURCE`, `NONLINEARLOAD`, or `VSOURCE` devices that should always be applied to the system together. No other sources will be active; the `SOURCEGROUP` must stand on its own. Therefore, some sources are likely to appear in more than one `SOURCEGROUP`. You can use this directive as often as needed to create multiple source groups.

When you use `SOURCEGROUP`, SuperHarm calculates a solution for each group in turn. Then, in TOP, The Output Processor, you can select one or more groups to be used when calculating harmonic currents and voltages. If this directive is not used, SuperHarm adds the solutions due to each group in turn to obtain a total solution, and saves only the total solution.

An Example

An facility contains four adjustable-speed drives. Drives 1A and 2A are modeled from measurements using `NONLINEARLOAD`s. There are no measurements available for drives 1B and 2B, so they are assumed spectrums and modeled using `NONLINEARLOAD`s.

The process is such that the “A” and “B” drives within a group must operate simultaneously, but the groups may operate individually or simultaneously.

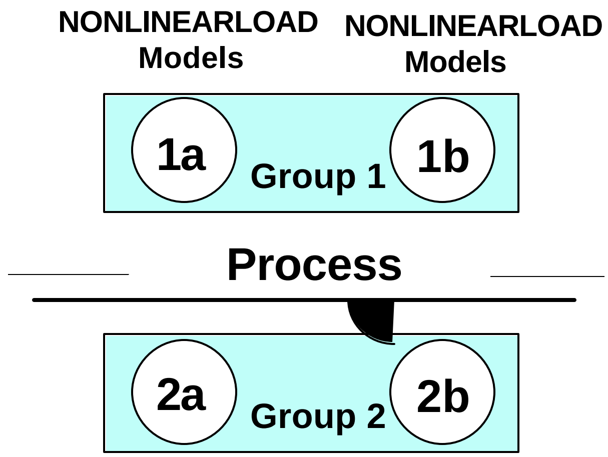


Figure 5-53: Sourcegroup Process

The circuit description for this system would be:

```
NONLINEARLOAD  Name = Drive1a  // Device data …
NONLINEARLOAD  Name = Drive2a  // Device data …
NONLINEARLOAD  Name = Drive1b  // Device data …
NONLINEARLOAD  Name = Drive2b  // Device data …
SOURCEGROUP    SourceList = {Drive1a, Drive1b}
SOURCEGROUP    SourceList = {Drive2a, Drive2b}
```

If you have a utility substation represented with `VSOURCE` models, those should also appear in <u>both</u> `SOURCEGROUP`s.

As described in the *TOP User's Guide,* you select the quantities that you wish to view (**Voltages** and **Currents** list boxes). Then you select which source groups should be applied to generate these quantities (**Sources** list box). If source groups were not used, the **Sources** list box would contain a single entry - ALL - which indicates that all sources in the system are always applied together.

TITLE

```
TITLE
Title1  = "Text String"
Title2  = "Text String"
Title3  = "Text String"
```

This directive is used to provide from one to three captions to be used by TOP, The Output Processor to label graphs and tables. A circuit description must have exactly one `TITLE` directive with at least `Title1` specified. The text strings entered for Title1 - Title3 should be enclosed in quotation marks. Imbedded spaces are OK. Strings of more than 64 characters are truncated by the output processor.

An Example

```
TITLE
Title1="Acme Rocket Motor Company"
Title3="Frequency scan at ASD bus"
```

Note that `Title2` is omitted. This causes TOP to insert a blank line between the two lines.

APPENDIX A

REFERENCE LIST

This appendix provides a chronological list of papers that may be utilized by the user as references. Each paper examines critical issues and modeling techniques for various harmonic concerns.

Page Intentionally Left Blank

Reference List

1. Owen, M.F. McGranaghan, and J.R. Vivirito*, "Distribution System Harmonics: Controls for Large Power Converters*," IEEE Transactions-PAS, Vol. 102, No. 3, March, 1982.

2. M.F. McGranaghan and R.F. Gustin, "*Power Converter Harmonics and Their Control on the Power System*," Rocky Mountain Electric League Spring Conference, Cheyenne, Wyoming, May 2-4, 1982.

3. "*Harmonics of Dispersed Generation and Effects on the Distribution System*," R. C. Dugan, Pennsylvania Electric Association, System Planning Committee Meeting, September 1983.

4. "*Distribution Feeder Harmonic Study Methodology*," M. F. McGranaghan, R. C. Dugan, J. A. King, W. T. Jewell, *IEEE Transactions on Power Apparatus and Systems,* pp. 3663-3671, Dec 1984.

5. "*Check Dispersed Generation Harmonics*," R. C. Dugan, *Electrical World,* Jan 1984.

6. "*Analyzing and Controlling Harmonic Distortion on Distribution Feeders*," R. C. Dugan, C. D. Ko, International Conference on Harmonics in Power Systems, Worcester Polytechnic Institute, Worcester, MA, Oct 1984.

7. M.F. McGranaghan, "*Instrumentation and Methodology for Power System Harmonic Measurements*," International Conference on Harmonics in Power Systems, Worcester, Mass., October 22-23, 1984.

8. M.F. McGranaghan, R.C. Dugan, J.A. King, and W.T. Jewell, "*Distribution Feeder Harmonic Study Methodology*," Presented at the 1984 IEEE T&D Show, Kansas City, Missouri, March, 1984.

9. E. W. Gunther, J. R. Frider, "*Measurement of Distribution System Ambient Harmonic Levels*", Pacific Coast Electrical Asso-

ciation Engineering and Operating Conference, March 17-18, 1986, San Francisco, California.

10. E. W. Gunther, M. F. McGranaghan, "*A PC-Based Simulation Program for Power System Harmonic Analysis*", Proceedings of the Second International Conference on Harmonics in Power Systems", October 6-8, 1986, Winnipeg, Manitoba, Canada.

11. Z. Jing, H. Fengreng, "*Studies on Modeling of Harmonic Impedance for Induction Motor*", Proceedings of the Second International Conference on Harmonics in Power Systems", October 6 -8, 1986, Winnipeg, Manitoba, Canada.

12. M. F. McGranaghan, E. W. Gunther, "*Measurement and Simulation of Distribution Harmonic Levels*", Proceedings of the Minnesota Power Systems Conference, October 7-9 1986, St. Paul, Minnesota.

13. T. Rizy, E.W. Gunther, and M.F. McGranaghan*, "Harmonics and Switching Concerns Associated with an Automated Distribution System*", Presented at the 1986 IEEE T&D Show, Anaheim, California, September, 1986.

14. M.F. McGranaghan and E.W. Gunther, "*Design of a PC-Based Harmonic Simulation Program*," 2nd International Conference on Harmonics in Power Systems, Winnipeg, Manitoba, October, 1986.

15. M.F. McGranaghan and E.W. Gunther, "*The Role of Simulations and Measurements in a Harmonic Standard*," PEA System Planning Committee Fall Meeting, Hershey, Pennsylvania, September 16-17, 1986.

16. D. T. Rizy, E. W. Gunther, M. F. McGranaghan, "*Transient and Harmonic Voltages Associated with Automated Capacitor Switching on Distribution Systems*", IEEE Transactions on Power Systems, Vol PWRS-2, No 3, pp. 713-723, August 1987.

17. *Electric Power System Harmonics Design Guide*, R. C. Dugan, M. F. McGranaghan, ORNL/Sub/81-95011/3, Oak Ridge Na-

tional Laboratory, Sept 1987. 2nd edition, Sept 1988. Also available from Cooper Power Systems.

18. “*Analyzing Harmonics in Commercial Power Systems*”*,* R. Dwyer, to be published by the Electric Power Research Institute.

19. "*A Program for Analysis of Power System Harmonics*," C. Gilker, R. V. Dwyer, R. C. Dugan, *IEEE Computer Applications in Power,* Vol. 2, No. 4, Oct 1989, pp 36-41.

20. R. W. Dwyer, E W. Gunther, R. Adapa, "*A Comparison of Solution Techniques for the Calculation of Harmonic Distortion Due to Adjustable Speed DC Drives*", Proceedings of the Fourth International Conference on Harmonics in Power Systems, October 4-6, 1990, Budapest, Hungary.

21. E. W. Gunther, "*An Integrated Computing Environment for the Simulation of Power System Harmonics*", Fifth International Conference on Harmonics in Power Systems, 1992, Atlanta Georgia.

22. “*Predicting Voltage Distortion in Systems with Multiple Random Harmonic Sources*”*,* S.Kaprielian, A. Emmanuel, R. Dwyer, et al, to be presented at the 1993 IEEE Power Engineering Society Summer Meeting.

23. "*Operational and Harmonic Concerns Associated with a Semiconverter*", 1994 IEEE IAS Rubber and Plastics Conference Record, Akron, Ohio, April 1994, C. Melhorn, Le Tang.

24. "Commercial Building Harmonics - Impact of Electronic Ballasts", 1994 IEEE PES Winter Power Meeting, Panel Session on Harmonics, New York, New York, January 1994, C. Melhorn, R. Schwabe.

25. “*Evaluation of Harmonic Impact from Compact Flourescent Lights on Distribution Systems*”, 1995 IEE PES Winter Power Meeting, Panel Session on Harmonics, New York, New York, January 1995, R. Dwyer, A. Khan, M. F. Mc Granaghan, Le Tang.

26. *Tutorial on Harmonics Modeling and Simulation*, IEEE Power Engineering Society Publication TP-125-0, 1998.

APPENDIX B

EXAMPLE DATA FILES

This appendix provides a listing of the example data files distributed with the SuperHarm installation.

Page Intentionally Left Blank

Ex1harma.sha

```
TITLE TITLE1="Example 1 - 44 kV Distribution System"
      TITLE2="All Capacitors in except proposed bank for BUS6"
      TITLE3="With Load"
!
!
! Case A: All Capacitors in except proposed bank for BUS6
!         With Load
!
!
!         Harmonic simulation case
!
!
!
!
!
!
!
! Utility source
!
VSOURCE NAME=VSRC BUS=SRCV MAG=25400
!
!
!
! Harmonic Source for Cases with Rectifier Modeled
!
!   2.5 MVA Rectifier, 6 Pulse, 4.16 kV, 6% commutation reactance
!
!
NONLINEARLOAD NAME=SRC1 BUS=RECT KVA=858.1599 KV=2.4018 DF=0.97107
TABLE=
{
{  1,     357.30050,  166.19},  //  Commas separate items
{  5,      65.16396,  110.51},
{  7,      42.53477,   82.08},
{ 11,      20.48435,   22.63},
{ 13,      14.32440,   -9.39},
{ 17,       6.91028,  -82.42},
{ 19,       4.96316, -125.51},
{ 23,       3.35249,  141.58},
{ 25,       3.07114,   98.55},
{ 29,       2.54135,   20.73},
{ 31,       2.21511,  -16.98},
{ 35,       1.57854,  -96.41},
{ 37,       1.34501, -139.45}   //  Ensure that you do not put a comma here
}
!
!
!  Metering element in series with the rectifier load
!
BRANCH  NAME=rectif  FROM=bus7l  TO=rect  R=1.0  X=0.0
!
!
! Source Equivalent at 44KVBUS
!   Note: All Impedance Values Given in Ohms at 60 Hz
!
BRANCH NAME=equiv FROM=44kvbus TO=SRCV R=2.46 X=5.65
!
```

```
! Capacitor Banks on the System
!   All the Capacitors are in Service except a new bank
!   proposed for BUS6
!
CAPACITOR NAME=bank01 FROM=44kvbus R=0.0 KV=46.0 MVA=6.00
CAPACITOR NAME=bank02 FROM=bus2l   R=0.0 KV=13.8 MVA=1.05
CAPACITOR NAME=bank03 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank04 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank05 FROM=bus4l   R=0.0 KV=13.8 MVA=0.75
!CAPACITOR NAME=bank06 FROM=bus6    R=0.0 KV=46.0 MVA=3.60  // proposed bank
!
!
! Loads
!
LINEARLOAD NAME=LOAD1 FROM=bus1  KVA=4500.0 KV=44.00 DF=0.99018
LINEARLOAD NAME=LOAD2 FROM=bus2l KVA=3100.0 KV=13.00 DF=0.94606
LINEARLOAD NAME=LOAD3 FROM=bus3l KVA=1100.0 KV=12.50 DF=0.85749
LINEARLOAD NAME=LOAD4 FROM=bus4l KVA=1300.0 KV=12.50 DF=0.97823
LINEARLOAD NAME=LOAD6 FROM=bus6  KVA= 600.0 KV=44.00 DF=0.85749
LINEARLOAD NAME=LOAD7 FROM=bus7l KVA=2000.0 KV= 4.16 DF=0.90000
!
!
! Transformers from 44 kV to Customer Buses
!
TRANSFORMER       NAME = B2TRAN  MVAb.HX = 100.0     MVA=10
 H.1  = BUS2       X.1  = BUS2L
kV.H  = 44.00    kV.X  = 13.00
%R.HX = 3.20     %X.HX = 86.80

TRANSFORMER       NAME = B3TRAN  MVAb.HX = 100.0     MVA=10
 H.1  = BUS3       X.1  = BUS3L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 12.10    %X.HX = 130.40

TRANSFORMER       NAME = B4TRAN  MVAb.HX = 100.0     MVA=10
 H.1  = BUS4       X.1  = BUS4L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 29.20    %X.HX = 214.5

TRANSFORMER       NAME = B7TRAN  MVAb.HX = 7.5       MVA=10
 H.1  = BUS7       X.1  = BUS7L
kV.H  = 44.00    kV.X  = 4.16
                 %X.HX = 4.6
!
!
! 44 kV Distribution Lines
!   Note: Line capacitance not included due to short line lengths
!         Impedances are in Ohms at 44 kV
!
BRANCH NAME=LINE1 FROM=44KVBUS TO=BUS1  R=0.0581  X=1.2778
BRANCH NAME=LINE2 FROM=44KVBUS TO=BUS2  R=3.678   X=8.0150
BRANCH NAME=LINE3 FROM=44KVBUS TO=BUS3  R=0.987   X=2.9810
BRANCH NAME=LINE4 FROM=44KVBUS TO=BUS4  R=1.084   X=1.7040
BRANCH NAME=LINE5 FROM=BUS4    TO=BUS6  R=3.717   X=7.8410
BRANCH NAME=LINE6 FROM=BUS6    TO=BUS7  R=0.620   X=1.3340
!
!
!
!
!
```

```
!
!
!
! Save currents for harmonic mode cases
!
RETAIN CURRENTS=YES
!
!
! End of Input File
!
....
```

Ex1harmb.sha

```
TITLE TITLE1="Example 1 - 44 kV Distribution System"
      TITLE2="All Capacitors in except one 1800 kVAr bank at BUS3L"
      TITLE3="With Load"
!
!
! Case A: All Capacitors in except one 1800 kVAr bank at BUS3L
!         With Load
!
!
!         Harmonic simulation case
!
!
!
!
!
!
!
! Utility source
!
VSOURCE NAME=VSRC BUS=SRCV MAG=25400
!
!
!
! Harmonic Source for Cases with Rectifier Modeled
!
!   2.5 MVA Rectifier, 6 Pulse, 4.16 kV, 6% commutation reactance
!
!
NONLINEARLOAD NAME=SRC1 BUS=RECT KVA=858.1599 KV=2.4018 DF=0.97107
TABLE=
{
{  1,     357.30050,  166.19},  //  Commas separate items
{  5,      65.16396,  110.51},
{  7,      42.53477,   82.08},
{ 11,      20.48435,   22.63},
{ 13,      14.32440,   -9.39},
{ 17,       6.91028,  -82.42},
{ 19,       4.96316, -125.51},
{ 23,       3.35249,  141.58},
{ 25,       3.07114,   98.55},
{ 29,       2.54135,   20.73},
{ 31,       2.21511,  -16.98},
{ 35,       1.57854,  -96.41},
{ 37,       1.34501, -139.45}   //  Ensure that you do not put a comma here
}
!
!
!  Metering element in series with the rectifier load
!
BRANCH  NAME=rectif  FROM=bus7l  TO=rect  R=1.0  X=0.0
!
!
! Source Equivalent at 44KVBUS
!   Note: All Impedance Values Given in Ohms at 60 Hz
!
BRANCH NAME=equiv FROM=44kvbus TO=SRCV R=2.46 X=5.65
!
```

```
! Capacitor Banks on the System
!   All the Capacitors are in Service except a new bank
!   proposed for BUS6
!
CAPACITOR NAME=bank01 FROM=44kvbus R=0.0 KV=46.0 MVA=6.00
CAPACITOR NAME=bank02 FROM=bus2l   R=0.0 KV=13.8 MVA=1.05
CAPACITOR NAME=bank03 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
!CAPACITOR NAME=bank04 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank05 FROM=bus4l   R=0.0 KV=13.8 MVA=0.75
CAPACITOR NAME=bank06 FROM=bus6    R=0.0 KV=46.0 MVA=3.60  // proposed bank
!
!
! Loads
!
LINEARLOAD NAME=LOAD1 FROM=bus1  KVA=4500.0 KV=44.00 DF=0.99018
LINEARLOAD NAME=LOAD2 FROM=bus2l KVA=3100.0 KV=13.00 DF=0.94606
LINEARLOAD NAME=LOAD3 FROM=bus3l KVA=1100.0 KV=12.50 DF=0.85749
LINEARLOAD NAME=LOAD4 FROM=bus4l KVA=1300.0 KV=12.50 DF=0.97823
LINEARLOAD NAME=LOAD6 FROM=bus6  KVA= 600.0 KV=44.00 DF=0.85749
LINEARLOAD NAME=LOAD7 FROM=bus7l KVA=2000.0 KV= 4.16 DF=0.90000
!
!
! Transformers from 44 kV to Customer Buses
!
TRANSFORMER       NAME = B2TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS2      X.1  = BUS2L
kV.H  = 44.00    kV.X  = 13.00
%R.HX = 3.20     %X.HX = 86.80

TRANSFORMER       NAME = B3TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS3      X.1  = BUS3L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 12.10    %X.HX = 130.40

TRANSFORMER       NAME = B4TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS4      X.1  = BUS4L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 29.20    %X.HX = 214.5

TRANSFORMER       NAME = B7TRAN  MVAb.HX = 7.5      MVA=10
 H.1  = BUS7      X.1  = BUS7L
kV.H  = 44.00    kV.X  = 4.16
                 %X.HX = 4.6
!
!
! 44 kV Distribution Lines
!   Note: Line capacitance not included due to short line lengths
!         Impedances are in Ohms at 44 kV
!
BRANCH NAME=LINE1 FROM=44KVBUS TO=BUS1  R=0.0581  X=1.2778
BRANCH NAME=LINE2 FROM=44KVBUS TO=BUS2  R=3.678   X=8.0150
BRANCH NAME=LINE3 FROM=44KVBUS TO=BUS3  R=0.987   X=2.9810
BRANCH NAME=LINE4 FROM=44KVBUS TO=BUS4  R=1.084   X=1.7040
BRANCH NAME=LINE5 FROM=BUS4    TO=BUS6  R=3.717   X=7.8410
BRANCH NAME=LINE6 FROM=BUS6    TO=BUS7  R=0.620   X=1.3340
!
!
!
!
!
```

```
!
!
!
! Don't save currents for scans
!
RETAIN CURRENTS=YES
!
!
! End of Input File
!
....
```

Ex1harmc.sha

```
TITLE TITLE1="Example 1 - 44 kV Distribution System"
      TITLE2="All Capacitors in except one 1800 kVAr bank at BUS3L"
      TITLE3="With Load, 1200 kVAr Fifth Harmonic Filter added at BUS7L"
!
!
! Case A: All Capacitors in except one 1800 kVAr bank at BUS3L
!         With Load
!         1200 kVAr Fifth Harmonic Filter added at Harmonic Source
!
!         Harmonic simulation case
!
!
!
!
!
!
!
! Utility source
!
VSOURCE NAME=VSRC BUS=SRCV MAG=25400
!
!
!
! Harmonic Source for Cases with Rectifier Modeled
!
!   2.5 MVA Rectifier, 6 Pulse, 4.16 kV, 6% commutation reactance
!
!
NONLINEARLOAD NAME=SRC1 BUS=RECT KVA=858.1599 KV=2.4018 DF=0.97107
TABLE=
{
{  1,      357.30050,  166.19},  //  Commas separate items
{  5,       65.16396,  110.51},
{  7,       42.53477,   82.08},
{ 11,       20.48435,   22.63},
{ 13,       14.32440,   -9.39},
{ 17,        6.91028,  -82.42},
{ 19,        4.96316, -125.51},
{ 23,        3.35249,  141.58},
{ 25,        3.07114,   98.55},
{ 29,        2.54135,   20.73},
{ 31,        2.21511,  -16.98},
{ 35,        1.57854,  -96.41},
{ 37,        1.34501, -139.45}   //  Ensure that you do not put a comma here
}
!
!
!  Metering element in series with the rectifier load
!
BRANCH  NAME=rectif  FROM=bus7l  TO=rect  R=1.0  X=0.0
!
!
! Source Equivalent at 44KVBUS
!   Note: All Impedance Values Given in Ohms at 60 Hz
!
BRANCH NAME=equiv FROM=44kvbus TO=SRCV R=2.46 X=5.65
!
```

```
! Capacitor Banks on the System
!   All the Capacitors are in Service except a new bank
!   proposed for BUS6
!
CAPACITOR NAME=bank01 FROM=44kvbus R=0.0 KV=46.0 MVA=6.00
CAPACITOR NAME=bank02 FROM=bus2l   R=0.0 KV=13.8 MVA=1.05
CAPACITOR NAME=bank03 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
!CAPACITOR NAME=bank04 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank05 FROM=bus4l   R=0.0 KV=13.8 MVA=0.75
CAPACITOR NAME=bank06 FROM=bus6    R=0.0 KV=46.0 MVA=3.60  // proposed bank
!
!
! Loads
!
LINEARLOAD NAME=LOAD1 FROM=bus1  KVA=4500.0 KV=44.00 DF=0.99018
LINEARLOAD NAME=LOAD2 FROM=bus2l KVA=3100.0 KV=13.00 DF=0.94606
LINEARLOAD NAME=LOAD3 FROM=bus3l KVA=1100.0 KV=12.50 DF=0.85749
LINEARLOAD NAME=LOAD4 FROM=bus4l KVA=1300.0 KV=12.50 DF=0.97823
LINEARLOAD NAME=LOAD6 FROM=bus6  KVA= 600.0 KV=44.00 DF=0.85749
LINEARLOAD NAME=LOAD7 FROM=bus7l KVA=2000.0 KV= 4.16 DF=0.90000
!
!
! Transformers from 44 kV to Customer Buses
!
TRANSFORMER       NAME = B2TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS2      X.1  = BUS2L
kV.H  = 44.00    kV.X  = 13.00
%R.HX = 3.20     %X.HX = 86.80

TRANSFORMER       NAME = B3TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS3      X.1  = BUS3L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 12.10    %X.HX = 130.40

TRANSFORMER       NAME = B4TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS4      X.1  = BUS4L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 29.20    %X.HX = 214.5

TRANSFORMER       NAME = B7TRAN  MVAb.HX = 7.5      MVA=10
 H.1  = BUS7      X.1  = BUS7L
kV.H  = 44.00    kV.X  = 4.16
                 %X.HX = 4.6
!
!
! 44 kV Distribution Lines
!   Note: Line capacitance not included due to short line lengths
!         Impedances are in Ohms at 44 kV
!
BRANCH NAME=LINE1 FROM=44KVBUS TO=BUS1  R=0.0581  X=1.2778
BRANCH NAME=LINE2 FROM=44KVBUS TO=BUS2  R=3.678   X=8.0150
BRANCH NAME=LINE3 FROM=44KVBUS TO=BUS3  R=0.987   X=2.9810
BRANCH NAME=LINE4 FROM=44KVBUS TO=BUS4  R=1.084   X=1.7040
BRANCH NAME=LINE5 FROM=BUS4    TO=BUS6  R=3.717   X=7.8410
BRANCH NAME=LINE6 FROM=BUS6    TO=BUS7  R=0.620   X=1.3340
!
!
!  Filter
!
SERIESFILTER NAME=FILTER1 CAPBUS=BUS7L MIDBUS=FLTMID INDBUS=GROUND
```

```
HARMONIC=4.8 KVA=1200 KV=4.16 XRRATIO=20.0
!
!
! Don't save currents for scans
!
!RETAIN CURRENTS=YES
!
!
! End of Input File
!
....
```

Ex1scana.sha

```
TITLE TITLE1="Example 1 - 44 kV Distribution System"
      TITLE2="All Capacitors in except proposed bank for BUS6"
      TITLE3="With Load"
!
!
! Case A: All Capacitors in except proposed bank for BUS6
!         With Load
!
!
!         Scan case
!
!         Solution will be for 1 Amp Injection at RECT
!         over a range of frequencies (60 Hz to 1200 Hz)
!
SCAN NAME=SCAN1 BUS=RECT FMIN=60 FMAX=1200 FINC=1 ANG=0.0
!
!
! Utility source
!
VSOURCE NAME=VSRC BUS=SRCV MAG=25400
!
!
!
! Harmonic Source for Cases with Rectifier Modeled
!
!   2.5 MVA Rectifier, 6 Pulse, 4.16 kV, 6% commutation reactance
!
!
NONLINEARLOAD NAME=SRC1 BUS=RECT KVA=858.1599 KV=2.4018 DF=0.97107
TABLE=
{
{  1,     357.30050,  166.19},  //  Commas separate items
{  5,      65.16396,  110.51},
{  7,      42.53477,   82.08},
{ 11,      20.48435,   22.63},
{ 13,      14.32440,   -9.39},
{ 17,       6.91028,  -82.42},
{ 19,       4.96316, -125.51},
{ 23,       3.35249,  141.58},
{ 25,       3.07114,   98.55},
{ 29,       2.54135,   20.73},
{ 31,       2.21511,  -16.98},
{ 35,       1.57854,  -96.41},
{ 37,       1.34501, -139.45}   //  Ensure that you do not put a comma here
}
!
!
!  Metering element in series with the rectifier load
!
BRANCH  NAME=rectif  FROM=bus7l  TO=rect  R=1.0  X=0.0
!
!
! Source Equivalent at 44KVBUS
!   Note: All Impedance Values Given in Ohms at 60 Hz
!
BRANCH NAME=equiv FROM=44kvbus TO=SRCV R=2.46 X=5.65
!
```

```
! Capacitor Banks on the System
!   All the Capacitors are in Service except a new bank
!   proposed for BUS6
!
CAPACITOR NAME=bank01 FROM=44kvbus R=0.0 KV=46.0 MVA=6.00
CAPACITOR NAME=bank02 FROM=bus2l   R=0.0 KV=13.8 MVA=1.05
CAPACITOR NAME=bank03 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank04 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank05 FROM=bus4l   R=0.0 KV=13.8 MVA=0.75
!CAPACITOR NAME=bank06 FROM=bus6    R=0.0 KV=46.0 MVA=3.60  // proposed bank
!
!
! Loads
!
LINEARLOAD NAME=LOAD1 FROM=bus1  KVA=4500.0 KV=44.00 DF=0.99018
LINEARLOAD NAME=LOAD2 FROM=bus2l KVA=3100.0 KV=13.00 DF=0.94606
LINEARLOAD NAME=LOAD3 FROM=bus3l KVA=1100.0 KV=12.50 DF=0.85749
LINEARLOAD NAME=LOAD4 FROM=bus4l KVA=1300.0 KV=12.50 DF=0.97823
LINEARLOAD NAME=LOAD6 FROM=bus6  KVA= 600.0 KV=44.00 DF=0.85749
LINEARLOAD NAME=LOAD7 FROM=bus7l KVA=2000.0 KV= 4.16 DF=0.90000
!
!
! Transformers from 44 kV to Customer Buses
!
TRANSFORMER       NAME = B2TRAN  MVAb.HX = 100.0  MVA=10
 H.1  = BUS2       X.1  = BUS2L
kV.H  = 44.00    kV.X  = 13.00
%R.HX = 3.20     %X.HX = 86.80

TRANSFORMER       NAME = B3TRAN  MVAb.HX = 100.0  MVA=10
 H.1  = BUS3       X.1  = BUS3L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 12.10    %X.HX = 130.40

TRANSFORMER       NAME = B4TRAN  MVAb.HX = 100.0  MVA=10
 H.1  = BUS4       X.1  = BUS4L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 29.20    %X.HX = 214.5

TRANSFORMER       NAME = B7TRAN  MVAb.HX = 7.5    MVA=10
 H.1  = BUS7       X.1  = BUS7L
kV.H  = 44.00    kV.X  = 4.16
                 %X.HX = 4.6
!
!
! 44 kV Distribution Lines
!   Note: Line capacitance not included due to short line lengths
!         Impedances are in Ohms at 44 kV
!
BRANCH NAME=LINE1 FROM=44KVBUS TO=BUS1  R=0.0581  X=1.2778
BRANCH NAME=LINE2 FROM=44KVBUS TO=BUS2  R=3.678   X=8.0150
BRANCH NAME=LINE3 FROM=44KVBUS TO=BUS3  R=0.987   X=2.9810
BRANCH NAME=LINE4 FROM=44KVBUS TO=BUS4  R=1.084   X=1.7040
BRANCH NAME=LINE5 FROM=BUS4    TO=BUS6  R=3.717   X=7.8410
BRANCH NAME=LINE6 FROM=BUS6    TO=BUS7  R=0.620   X=1.3340
!
!
!
!
!
```

```
!
RETAIN VOLTAGELIST={ RECT, 44KVBUS, BUS2L, BUS3L, BUS4L, BUS7L }
!
!
! Don't save currents for scans
!
!RETAIN CURRENTS=YES
!
!
! End of Input File
!
....
```

Ex1scanb.sha

```
TITLE TITLE1="Example 1 - 44 kV Distribution System"
      TITLE2="All Capacitors in except one 1800 kVAr bank at BUS3L"
      TITLE3="With Load"
!
!
! Case A: All Capacitors in except one 1800 kVAr bank at BUS3L
!         With Load
!
!
!         Scan case
!
!         Solution will be for 1 Amp Injection at RECT
!         over a range of frequencies (60 Hz to 1200 Hz)
!
SCAN NAME=SCAN1 BUS=RECT FMIN=60 FMAX=1200 FINC=10 ANG=0.0
!
!
! Utility source
!
VSOURCE NAME=VSRC BUS=SRCV MAG=25400
!
!
!
! Harmonic Source for Cases with Rectifier Modeled
!
!   2.5 MVA Rectifier, 6 Pulse, 4.16 kV, 6% commutation reactance
!
!
NONLINEARLOAD NAME=SRC1 BUS=RECT KVA=858.1599 KV=2.4018 DF=0.97107
TABLE=
{
{  1,      357.30050,  166.19},  //  Commas separate items
{  5,       65.16396,  110.51},
{  7,       42.53477,   82.08},
{ 11,       20.48435,   22.63},
{ 13,       14.32440,   -9.39},
{ 17,        6.91028,  -82.42},
{ 19,        4.96316, -125.51},
{ 23,        3.35249,  141.58},
{ 25,        3.07114,   98.55},
{ 29,        2.54135,   20.73},
{ 31,        2.21511,  -16.98},
{ 35,        1.57854,  -96.41},
{ 37,        1.34501, -139.45}   //  Ensure that you do not put a comma here
}
!
!
!  Metering element in series with the rectifier load
!
BRANCH  NAME=rectif  FROM=bus7l  TO=rect  R=1.0  X=0.0
!
!
! Source Equivalent at 44KVBUS
!   Note: All Impedance Values Given in Ohms at 60 Hz
!
BRANCH NAME=equiv FROM=44kvbus TO=SRCV R=2.46 X=5.65
!
```

```
! Capacitor Banks on the System
!   All the Capacitors are in Service except a new bank
!   proposed for BUS6
!
CAPACITOR NAME=bank01 FROM=44kvbus R=0.0 KV=46.0 MVA=6.00
CAPACITOR NAME=bank02 FROM=bus2l   R=0.0 KV=13.8 MVA=1.05
CAPACITOR NAME=bank03 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
!CAPACITOR NAME=bank04 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank05 FROM=bus4l   R=0.0 KV=13.8 MVA=0.75
CAPACITOR NAME=bank06 FROM=bus6    R=0.0 KV=46.0 MVA=3.60  // proposed bank
!
!
! Loads
!
LINEARLOAD NAME=LOAD1 FROM=bus1  KVA=4500.0 KV=44.00 DF=0.99018
LINEARLOAD NAME=LOAD2 FROM=bus2l KVA=3100.0 KV=13.00 DF=0.94606
LINEARLOAD NAME=LOAD3 FROM=bus3l KVA=1100.0 KV=12.50 DF=0.85749
LINEARLOAD NAME=LOAD4 FROM=bus4l KVA=1300.0 KV=12.50 DF=0.97823
LINEARLOAD NAME=LOAD6 FROM=bus6  KVA= 600.0 KV=44.00 DF=0.85749
LINEARLOAD NAME=LOAD7 FROM=bus7l KVA=2000.0 KV= 4.16 DF=0.90000
!
!
! Transformers from 44 kV to Customer Buses
!
TRANSFORMER       NAME = B2TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS2      X.1  = BUS2L
kV.H  = 44.00    kV.X  = 13.00
%R.HX = 3.20     %X.HX = 86.80

TRANSFORMER       NAME = B3TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS3      X.1  = BUS3L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 12.10    %X.HX = 130.40

TRANSFORMER       NAME = B4TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS4      X.1  = BUS4L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 29.20    %X.HX = 214.5

TRANSFORMER       NAME = B7TRAN  MVAb.HX = 7.5      MVA=10
 H.1  = BUS7      X.1  = BUS7L
kV.H  = 44.00    kV.X  = 4.16
                 %X.HX = 4.6
!
!
! 44 kV Distribution Lines
!   Note: Line capacitance not included due to short line lengths
!         Impedances are in Ohms at 44 kV
!
BRANCH NAME=LINE1 FROM=44KVBUS TO=BUS1  R=0.0581  X=1.2778
BRANCH NAME=LINE2 FROM=44KVBUS TO=BUS2  R=3.678   X=8.0150
BRANCH NAME=LINE3 FROM=44KVBUS TO=BUS3  R=0.987   X=2.9810
BRANCH NAME=LINE4 FROM=44KVBUS TO=BUS4  R=1.084   X=1.7040
BRANCH NAME=LINE5 FROM=BUS4    TO=BUS6  R=3.717   X=7.8410
BRANCH NAME=LINE6 FROM=BUS6    TO=BUS7  R=0.620   X=1.3340
!
!
!
!
!
```

```
!
!
!
! Don't save currents for scans
!
!RETAIN CURRENTS=YES
!
!
! End of Input File
!
....
```

Ex1scanc.sha

```
TITLE TITLE1="Example 1 - 44 kV Distribution System"
      TITLE2="All Capacitors in except one 1800 kVAr bank at BUS3L"
      TITLE3="With Load, 1200 kVAr Fifth Harmonic Filter added at BUS7L"
!
!
! Case A: All Capacitors in except one 1800 kVAr bank at BUS3L
!         With Load
!         1200 kVAr Fifth Harmonic Filter added at Harmonic Source
!
!         Scan case
!
!         Solution will be for 1 Amp Injection at RECT
!         over a range of frequencies (60 Hz to 1200 Hz)
!
SCAN NAME=SCAN1 BUS=RECT FMIN=60 FMAX=1200 FINC=10 ANG=0.0
!
!
! Utility source
!
VSOURCE NAME=VSRC BUS=SRCV MAG=25400
!
!
!
! Harmonic Source for Cases with Rectifier Modeled
!
!   2.5 MVA Rectifier, 6 Pulse, 4.16 kV, 6% commutation reactance
!
!
NONLINEARLOAD NAME=SRC1 BUS=RECT KVA=858.1599 KV=2.4018 DF=0.97107
TABLE=
{
{  1,     357.30050,  166.19},  //  Commas separate items
{  5,      65.16396,  110.51},
{  7,      42.53477,   82.08},
{ 11,      20.48435,   22.63},
{ 13,      14.32440,   -9.39},
{ 17,       6.91028,  -82.42},
{ 19,       4.96316, -125.51},
{ 23,       3.35249,  141.58},
{ 25,       3.07114,   98.55},
{ 29,       2.54135,   20.73},
{ 31,       2.21511,  -16.98},
{ 35,       1.57854,  -96.41},
{ 37,       1.34501, -139.45}   //  Ensure that you do not put a comma here
}
!
!
!  Metering element in series with the rectifier load
!
BRANCH  NAME=rectif  FROM=bus7l  TO=rect  R=1.0  X=0.0
!
!
! Source Equivalent at 44KVBUS
!   Note: All Impedance Values Given in Ohms at 60 Hz
!
```

```
BRANCH NAME=equiv FROM=44kvbus TO=SRCV R=2.46 X=5.65
!
! Capacitor Banks on the System
!   All the Capacitors are in Service except a new bank
!   proposed for BUS6
!
CAPACITOR NAME=bank01 FROM=44kvbus R=0.0 KV=46.0 MVA=6.00
CAPACITOR NAME=bank02 FROM=bus2l   R=0.0 KV=13.8 MVA=1.05
CAPACITOR NAME=bank03 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
!CAPACITOR NAME=bank04 FROM=bus3l   R=0.0 KV=13.8 MVA=1.80
CAPACITOR NAME=bank05 FROM=bus4l   R=0.0 KV=13.8 MVA=0.75
CAPACITOR NAME=bank06 FROM=bus6    R=0.0 KV=46.0 MVA=3.60  // proposed bank
!
!
! Loads
!
LINEARLOAD NAME=LOAD1 FROM=bus1  KVA=4500.0 KV=44.00 DF=0.99018
LINEARLOAD NAME=LOAD2 FROM=bus2l KVA=3100.0 KV=13.00 DF=0.94606
LINEARLOAD NAME=LOAD3 FROM=bus3l KVA=1100.0 KV=12.50 DF=0.85749
LINEARLOAD NAME=LOAD4 FROM=bus4l KVA=1300.0 KV=12.50 DF=0.97823
LINEARLOAD NAME=LOAD6 FROM=bus6  KVA= 600.0 KV=44.00 DF=0.85749
LINEARLOAD NAME=LOAD7 FROM=bus7l KVA=2000.0 KV= 4.16 DF=0.90000
!
!
! Transformers from 44 kV to Customer Buses
!
TRANSFORMER       NAME = B2TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS2       X.1  = BUS2L
kV.H  = 44.00    kV.X  = 13.00
%R.HX = 3.20     %X.HX = 86.80

TRANSFORMER       NAME = B3TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS3       X.1  = BUS3L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 12.10    %X.HX = 130.40

TRANSFORMER       NAME = B4TRAN  MVAb.HX = 100.0    MVA=10
 H.1  = BUS4       X.1  = BUS4L
kV.H  = 44.00    kV.X  = 12.50
%R.HX = 29.20    %X.HX = 214.5

TRANSFORMER       NAME = B7TRAN  MVAb.HX = 7.5      MVA=10
 H.1  = BUS7       X.1  = BUS7L
kV.H  = 44.00    kV.X  = 4.16
                 %X.HX = 4.6
!
!
! 44 kV Distribution Lines
!   Note: Line capacitance not included due to short line lengths
!         Impedances are in Ohms at 44 kV
!
BRANCH NAME=LINE1 FROM=44KVBUS TO=BUS1  R=0.0581  X=1.2778
BRANCH NAME=LINE2 FROM=44KVBUS TO=BUS2  R=3.678   X=8.0150
BRANCH NAME=LINE3 FROM=44KVBUS TO=BUS3  R=0.987   X=2.9810
BRANCH NAME=LINE4 FROM=44KVBUS TO=BUS4  R=1.084   X=1.7040
BRANCH NAME=LINE5 FROM=BUS4    TO=BUS6  R=3.717   X=7.8410
BRANCH NAME=LINE6 FROM=BUS6    TO=BUS7  R=0.620   X=1.3340
!
!
!  Filter
```

```
!
SERIESFILTER NAME=FILTER1 CAPBUS=BUS7L MIDBUS=FLTMID INDBUS=GROUND
HARMONIC=4.8 KVA=1200 KV=4.16 XRRATIO=20.0
!
!
! Don't save currents for scans
!
!RETAIN CURRENTS=YES
!
!
! End of Input File
!
....
```

I N D E X