NOTICE: This document contains references to Agilent Technologies. Agilent's former Test and Measurement business has become Keysight Technologies. For more information, go to **www.keysight.com.**

KEYSIGHT
TECHNOLOGIES

Agilent Technologies

Advanced Design System 2011

September 2011
LineCalc

© Agilent Technologies, Inc. 2000-2011
5301 Stevens Creek Blvd., Santa Clara, CA 95052 USA
No part of this documentation may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Mentor products and processes are registered trademarks of Mentor Graphics Corporation. * Calibre is a trademark of Mentor Graphics Corporation in the US and other countries. "Microsoft®, Windows®, MS Windows®, Windows NT®, Windows 2000® and Windows Internet Explorer® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Oracle and Java and registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC. FLEXlm is a trademark of Globetrotter Software, Incorporated. Layout Boolean Engine by Klaas Holwerda, v1.7 http://www.xs4all.nl/~kholwerd/bool.html . FreeType Project, Copyright (c) 1996-1999 by David Turner, Robert Wilhelm, and Werner Lemberg. QuestAgent search engine (c) 2000-2002, JObjects. Motif is a trademark of the Open Software Foundation. Netscape is a trademark of Netscape Communications Corporation. Netscape Portable Runtime (NSPR), Copyright (c) 1998-2003 The Mozilla Organization. A copy of the Mozilla Public License is at http://www.mozilla.org/MPL/ . FFTW, The Fastest Fourier Transform in the West, Copyright (c) 1997-1999 Massachusetts Institute of Technology. All rights reserved.

The following third-party libraries are used by the NlogN Momentum solver:

"This program includes Metis 4.0, Copyright © 1998, Regents of the University of Minnesota", http://www.cs.umn.edu/~metis , METIS was written by George Karypis (karypis@cs.umn.edu).

Intel@ Math Kernel Library, http://www.intel.com/software/products/mkl

SuperLU_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7-zip - 7-Zip Copyright: Copyright (C) 1999-2009 Igor Pavlov. Licenses for files are: 7z.dll: GNU LGPL + unRAR restriction, All other files: GNU LGPL. 7-zip License: This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful,but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. unRAR copyright: The decompression engine for RAR archives was developed using source code of unRAR program.All copyrights to original unRAR code are owned by Alexander Roshal. unRAR License: The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver. 7-zip Availability: http://www.7-zip.org/

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. AMD Availability: http://www.cise.ufl.edu/research/sparse/amd

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License

as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: http://www.cise.ufl.edu/research/sparse/umfpack UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at http://www.cise.ufl.edu/research/sparse . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at http://www.cise.ufl.edu/research/sparse . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. http://www.mathworks.com . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at http://www.netlib.org). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

Qt Version 4.6.3 - Qt Notice: The Qt code was modified. Used by permission. Qt copyright: Qt Version 4.6.3, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved. Qt License: Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. Qt Availability: http://www.qtsoftware.com/downloads Patches Applied to Qt can be found in the installation at: $HPEESOF_DIR/prod/licenses/thirdparty/qt/patches. You may also contact Brian Buchanan at Agilent Inc. at brian_buchanan@agilent.com for more information.

The HiSIM_HV source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code, is owned by Hiroshima University and/or STARC.

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at http://systemc.org/ . This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

Restricted Rights Legend U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Using LineCalc

LineCalc is an analysis and synthesis program for calculating electrical and physical parameters of single and coupled transmission lines.

LineCalc can communicate directly with the circuit simulators. You can send parameter data for selected circuit design elements, along with data on any associated substrates or walls, directly from the simulator to LineCalc. After the element parameter values are calculated, you can update the associated schematic or layout circuit design in the active simulator immediately with the LineCalc results. Or you can place a newly synthesized component into the Schematic window.

Using dialog boxes, you can make changes in parameter values of your transmission lines and see the results of those changes on the screen. You can use LineCalc like a spreadsheet, in the sense that a change in one value brings about a recalculation through all related values when you choose the appropriate **Calculate** button.

You can print results, save results to an element data file, or instantly update a schematic or layout in a circuit design.

Note
LineCalc cannot be used with the signal processing simulators.

The Basic LineCalc Process

- Select a LineCalc component.
- Select one or more independent parameters for calculation.
- Change default parameter values, as necessary.
- Perform the analysis or synthesis.
- Update the design, if applicable.

Key Terms

Analysis - Calculation of electrical parameters from physical data.

Synthesis - Calculation of physical parameters from electrical data.

Shared Parameters - Parameters common to a group of components and required for both analysis and synthesis.

LineCalc File (<filename>.lcs) - The data file created when results from a LineCalc analysis or synthesis of a component are saved. The file contains information about the component's parameters, associated shared parameters, frequency, and units to support design iteration.This data file is created when you save results from a LineCalc analysis or synthesis of a component. The file contains information about the component's parameters, associated shared parameters, frequency, and units to support design iteration.

Starting LineCalc

Before running LineCalc, you should be familiar with the basic operation of the ADS design environment. (For details, refer to *Schematic Capture and Layout* (usrguide).)

To run LineCalc from within the Design Environment:

- From the Schematic window, choose **Tools** > **LineCalc** > **Start LineCalc**.

To run LineCalc in stand-alone mode (UNIX mode):

- Type `linecalc` in terminal window

> **Note**
> Starting LineCalc in this manner assumes you have established a path to the ADS installation directory. If you have not, type `<install_dir>/bin/linecalc` , where `<install_dir>` represents your complete installation path. For details on establishing a path statement, refer to *Installing Advanced Design System on UNIX and Linux* (install).

When LineCalc is launched, the LineCalc window appears, as well as the Message/Status window. The Message/Status window displays messages about the status of the current process as well as warning messages.

The LineCalc Window

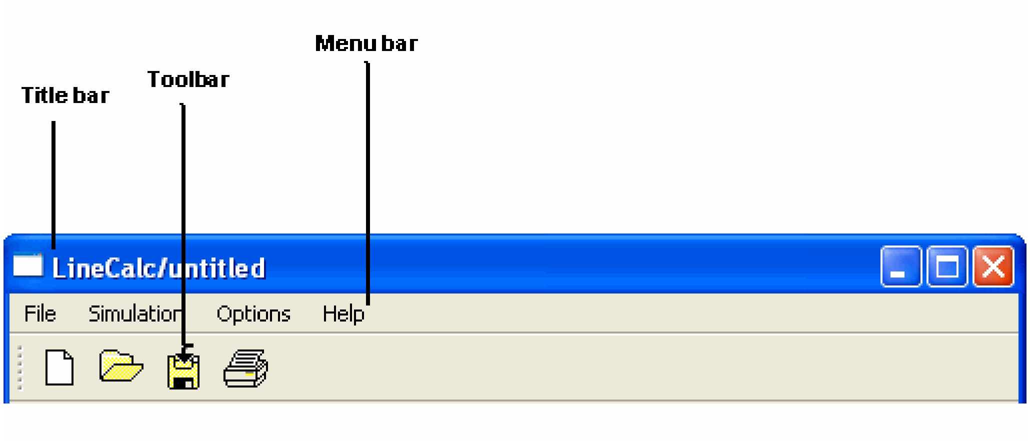
The LineCalc window is divided into several regions.

- Title Bar, Toolbar, and Menu Bar
- Component Display
- Shared Parameters Display
- Parameters Display
- Results Display
- Status Bar

The following sections describe these regions.

Title Bar, Toolbar, and Menu Bar

The *Title bar* displays the application name and the name of the active data file (or untitled if no data file is associated with the data being displayed).

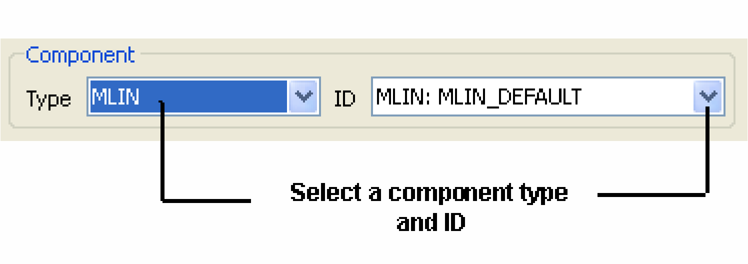


The *Toolbar* contains buttons for frequently-used commands.

The *Menu bar* displays the available menus and commands.

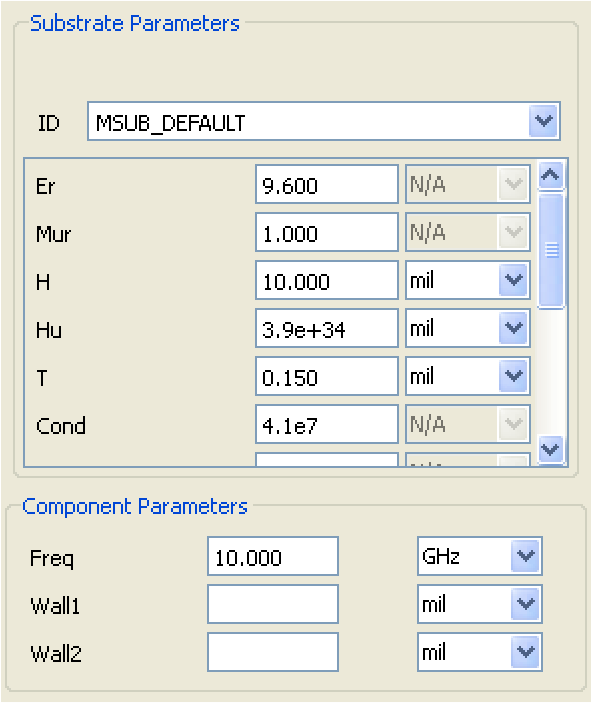
Component Display

The *Component* display lists the currently-selected component type and ID. You can choose a different component type and ID to analyze or synthesize, using the **Type** or **ID** drop-down list.



Shared Parameters Display

The *Shared Parameters* display consists of the *Substrate Parameters* and the *Component Parameters* sections. The parameters shared by all components with substrate parameters are listed in the Substrate Parameters area. The parameters shared by components with frequency or mwall parameters are listed in the Component Parameters area.



Substrate Parameters

ID MSUB_DEFAULT

Er	9.600	N/A
Mur	1.000	N/A
H	10.000	mil
Hu	3.9e+34	mil
T	0.150	mil
Cond	4.1e7	N/A

Component Parameters

Freq	10.000	GHz
Wall1		mil
Wall2		mil

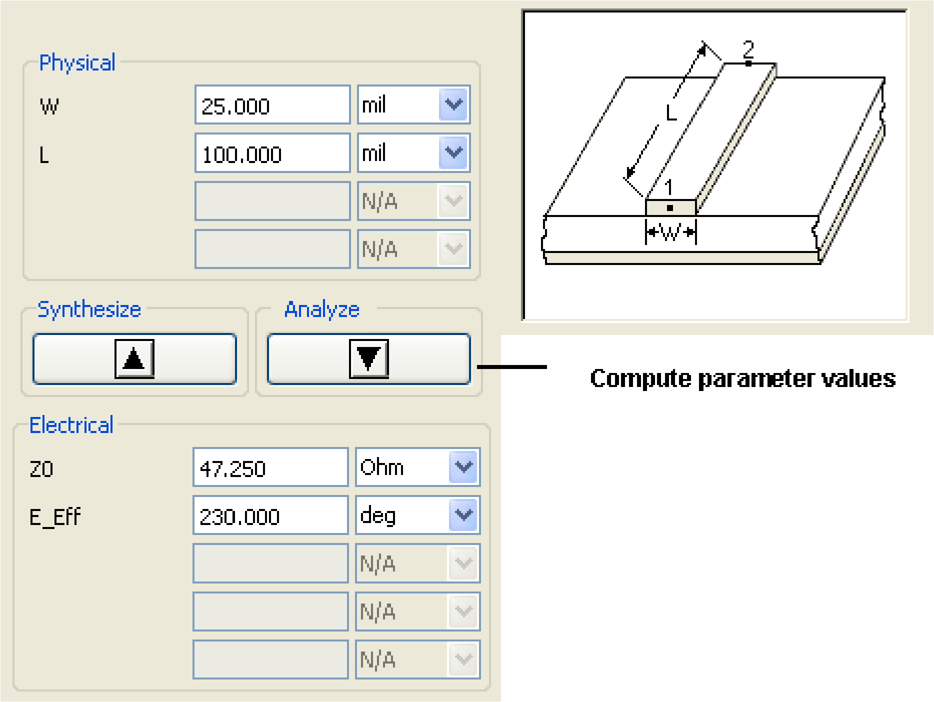
Parameters Display

The *Parameters* display lists the physical and electrical parameter values of the currently-selected component.

In the *Physical* section, the *Fix* button indicates the status of the associated physical parameter:

- If the button indicates Fixed, the adjacent parameter is fixed during synthesis
- If the button indicates Fix, the adjacent parameter is updated during synthesis

You can compute the physical parameter values from electrical data by clicking **Synthesize**. Clicking **Analyze** computes the electrical component parameters from physical data.



Physical

W 25.000 mil

L 100.000 mil

N/A

N/A

Electrical

Z0 47.250 Ohm

E_Eff 230.000 deg

N/A

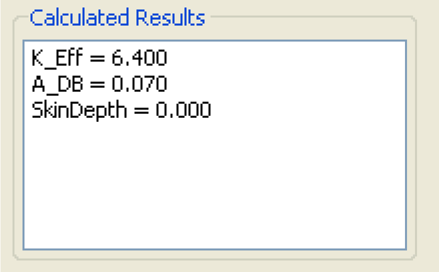
N/A

N/A

Compute parameter values

Results Display

The *Results* display shows the parameter values that change as a result of the synthesis or analysis. These are displayed as the calculated results.

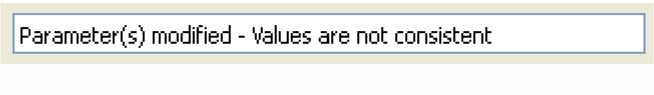


Calculated Results

K_Eff = 6.400
A_DB = 0.070
SkinDepth = 0.000

Status Bar

The *Status Bar* at the bottom of the window shows the status of your displayed data. When you have changed a parameter value without recalculation (synthesis or analysis), the message appears: *Parameter(s) modified - Values are not consistent.* After you perform the calculation, the message appears: *Values are consistent*.



Parameter(s) modified - Values are not consistent

LineCalc Menus and Commands

This section describes the LineCalc Menus and Commands. The LineCalc User Interface has the following menus:

- File Menu
- Simulation Menu
- Options Menu
- Help Menu

Changing the Default Data Directory

The LineCalc program reads the files from and save these to your program startup directory, unless you change the default directory.
To use an alternate directory during your current program session:

1. From the Synthesis/LineCalc window, choose **File** > **Change Dir.**
2. Use the displayed dialog box to specify a directory path (other than the program startup directory) for reading and saving files until you exit the program.

Resetting the Display

You must clear any currently active work before selecting or importing components for a new synthesis or analysis.

- To save your work, choose **File** > **Save**; then choose **File** > **New**.
- To cancel a design in progress and start over, choose **File** > **New**.

> **Note**
> Choosing *New* displays the default component.

Opening a Component File

After resetting the display, you can activate a component that you want to synthesize or analyze in one of these ways:

- By opening a previously-saved LineCalc component file, or
- By importing components from the active schematic or layout design

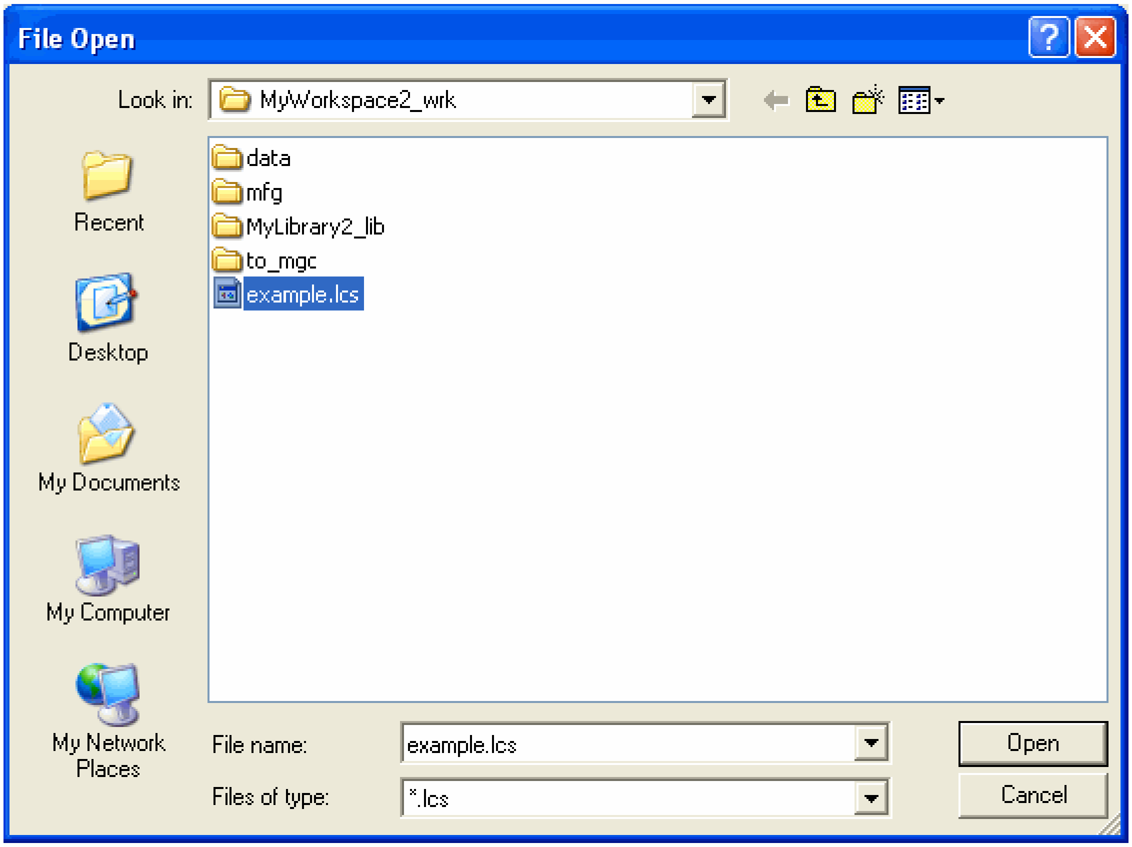
Note
When you activate LineCalc, the last-saved component is displayed. Clear the display before opening a component file (see Resetting the Display).

Opening a Previously-Saved LineCalc Component File

You can activate a component that you want to synthesize or analyze by opening a previously-saved LineCalc component file.

To open a previously-saved LineCalc component file:

1. Choose **File** > **Open** to display the File Open dialog box.



2. Specify the path to the directory where the component file is located.
3. Select the component file and click **Open**. The component data saved to that file becomes active in the LineCalc window.

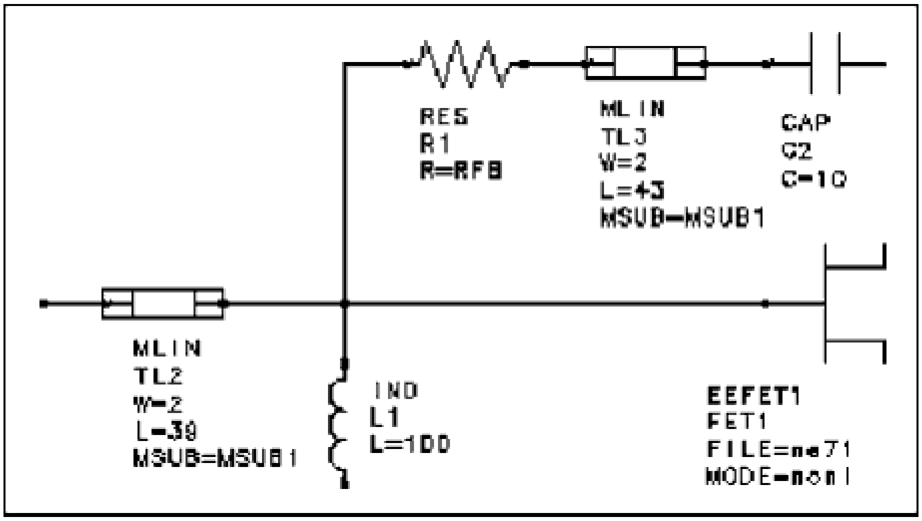
Importing Components from the Active Design

You can activate a component that you want to synthesize or analyze by importing components from the active schematic or layout design.

To import components from an active design:

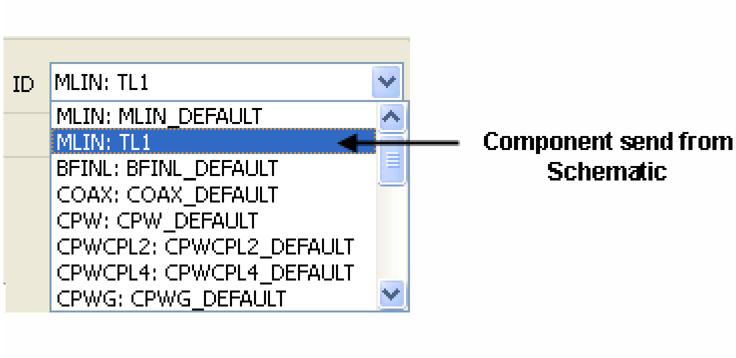
1. In the Schematic or Layout window, select the component you want to send to LineCalc. In the illustration, a portion of a schematic example, MLIN:TL2 is selected.

   Note
   You can send only one component at a time.



2. Choose **Tools** > **LineCalc** > **Send Selected Components to LineCalc**. The component data for all selected components is sent to LineCalc.

In the LineCalc window, each component that you import from the Schematic or Layout window, along with its ID name, is listed in the *Component ID Selection* dialog box. In the illustration the two MLINs (TL2 and TL3) that were sent to LineCalc from the Schematic window are now displayed in the Component ID Selection list and can be selected for evaluation.


ID MLIN: TL1

MLIN: MLIN_DEFAULT
MLIN: TL1
BFINL: BFINL_DEFAULT
COAX: COAX_DEFAULT
CPW: CPW_DEFAULT
CPWCPL2: CPWCPL2_DEFAULT
CPWCPL4: CPWCPL4_DEFAULT
CPWG: CPWG_DEFAULT

Only one component/ID calculation can be performed at a time. For example, select and perform any calculations on MLIN:TL3 before you attempt to select and modify MLIN:TL2.

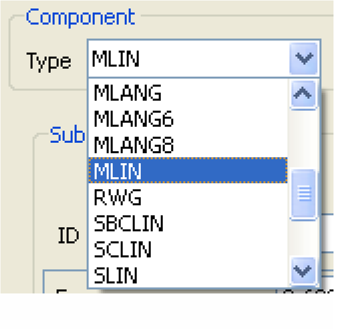
Selecting a Component

Note
Before selecting or importing components to begin a new synthesis or analysis, be sure to clear any currently active work. See Resetting the Display".

To select a component:

1. Select a component type using the **Component Type** drop-down list. The LineCalc window changes to display the active Component ID, parameters, and other data.

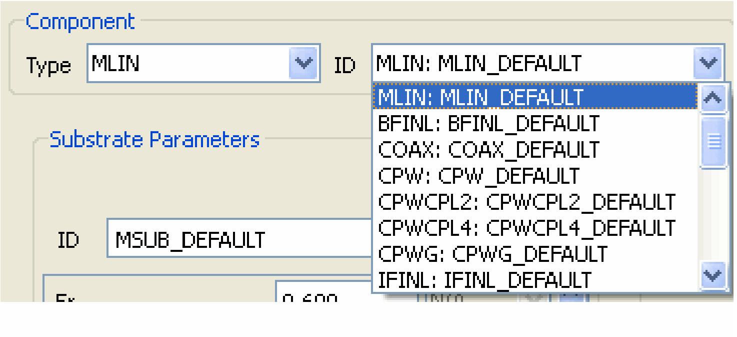
Select a Component Type



Component

Type MLIN

Select a Component ID



Component

Type MLIN

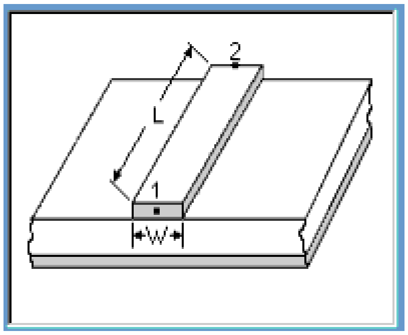
ID MLIN: MLIN_DEFAULT

Substrate Parameters

ID MSUB_DEFAULT

Displaying Component Help

To display detailed information for a component, select **Help** > **What's This?** and place the question mark on the component illustration.

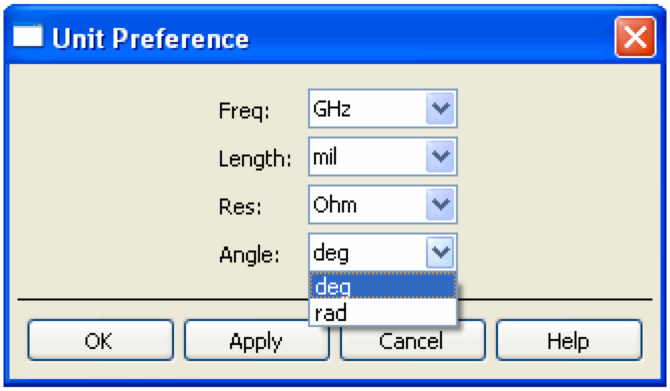


Changing Default Units

You can modify program units or frequencies before calculating the results by setting the Frequency, Length, Resistance, and Angle units.

To modify default units or frequencies:

1. In the Menu bar, select **Options > Preference**. Change the default unit preferences by selecting from the appropriate Unit drop-down lists.



Freq: GHz
Length: mil
Res: Ohm
Angle: deg

The following table lists the unit, the options available for each unit, and the default option.

Unit	Options	Default
Frequency	GHz, MHz, kHz, Hz	GHz
Length	in, mil, m, cm, mm, um	mil
Resistance	kohm, ohm	ohm
Angle	deg, rad	deg

2. In the **Units Preference** dialog box, modify the settings and click **OK** to accept your changes. The newly-chosen unit is reflected when a new component is chosen.

Changing Parameter Values

To change any values in the Physical or Electrical parameter fields:

- Select the unit and type a new value in the field.

To change Shared parameter values:

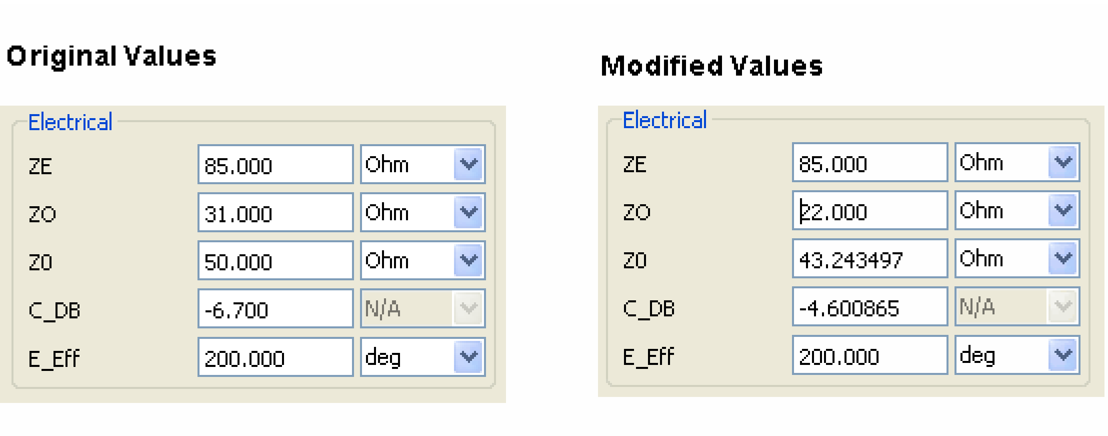
- Select the default value and type a new value in the field.
- To change the units associated with each shared parameter, select the units button next to each parameter field.

Note
See also: Interdependent Values for Electrical Parameters and Fixed Values for Physical Parameters

Interdependent Values for Electrical Parameters

For certain components, some of the electrical parameter values depend on the current values assigned to other parameters. In the following illustration, note the beginning values for ZO and Z0 in the Electrical portion of the Parameters display for the MCLIN component.

To observe the values of related parameters change automatically, change the value of ZO to 22 and press Enter (or click one of the other fields containing interdependent parameters).



Original Values

Electrical

ZE	85.000	Ohm
ZO	31.000	Ohm
Z0	50.000	Ohm
C_DB	-6.700	N/A
E_Eff	200.000	deg

Modified Values

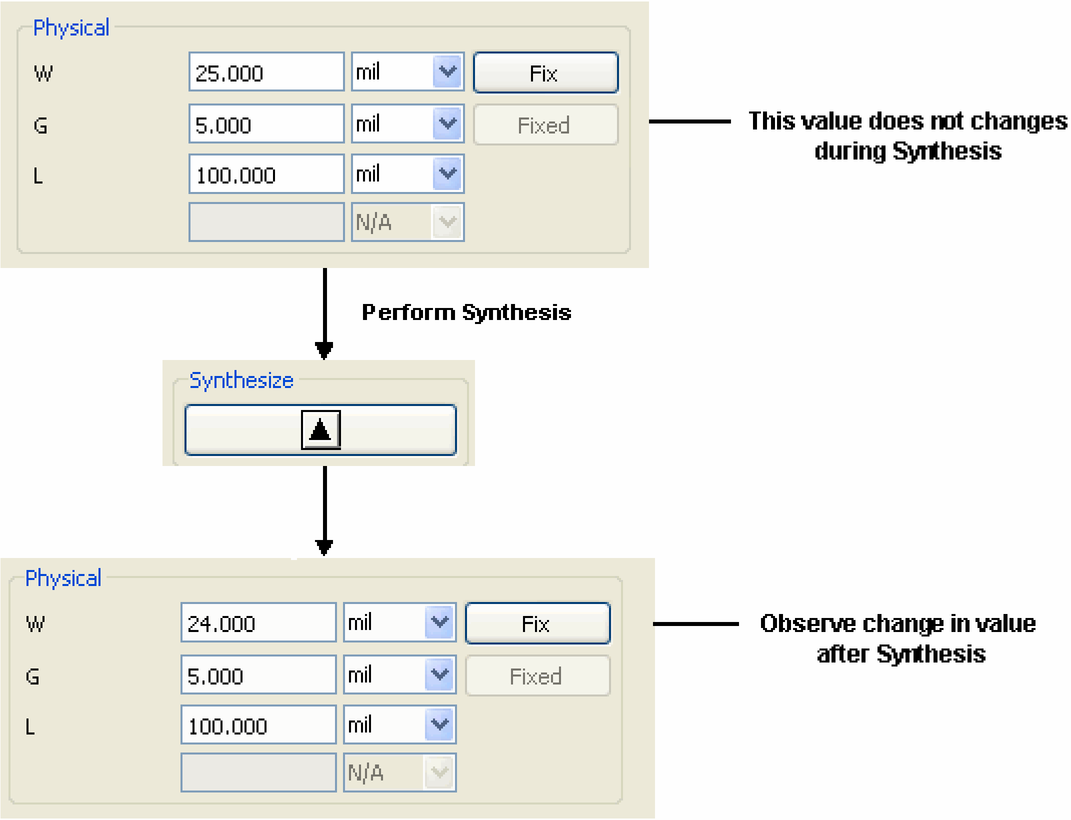
Electrical

ZE	85.000	Ohm
ZO	22.000	Ohm
Z0	43.243497	Ohm
C_DB	-4.600865	N/A
E_Eff	200.000	deg

The display updates to show that the corresponding Z0 value changed from 48.815 to 38.099. Also, the C_DB value changed from −10.676 to −6.022. Generally, the last two visited fields are used to compute the other parameters.

Fixed Values for Physical Parameters

Certain components require the value of one of the physical parameters to be fixed (not modifiable) during synthesis. For these components, one parameter must be assigned a fixed value before you can perform a synthesis. For example, observe how the W and G parameters of the CPW component are interrelated. By default, when you select the CPW component, the value of G is fixed and the value of W is not. This default condition is shown by the label on the **Fixed** button, which is dimmed.



W	25.000	mil	Fix
G	5.000	mil	Fixed
L	100.000	mil	
		N/A	

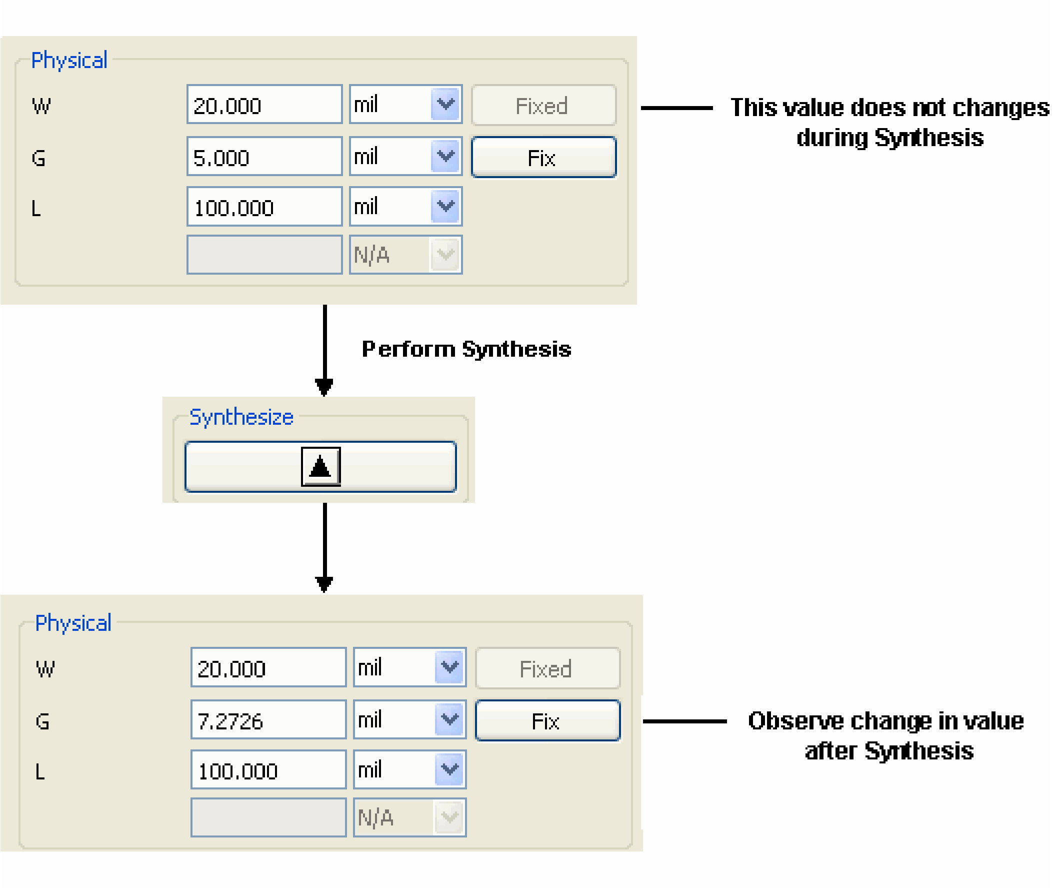
This value does not changes during Synthesis

Perform Synthesis

W	24.000	mil	Fix
G	5.000	mil	Fixed
L	100.000	mil	
		N/A	

Observe change in value after Synthesis

If you want to change the value of G, you must click the **Fix** button for W. The new condition is shown by the label on the button which changes to read Fixed and is dimmed. An active **Fix** button on the G value indicates that value is currently not fixed and will change during synthesis.



W 20.000 mil
G 5.000 mil
L 100.000 mil
N/A

This value does not changes during Synthesis

Perform Synthesis

W 20.000 mil
G 7.2726 mil
L 100.000 mil
N/A

Observe change in value after Synthesis

Performing a Synthesis

A synthesis computes the physical parameter values based on the electrical parameter values.

To synthesize a revised component:

- Select **Simulate** > **Compute Physical Params** or click **Calculate** (Up button) in the Parameters display. The values for physical parameters (shown in the upper portion of the display) are computed from the electrical values and the values in the Results display are updated.

Performing an Analysis

An analysis computes the electrical component parameters based on the physical parameter values.

To analyze a revised component:

- Select **Simulate** > **Compute Electrical Params** or click **Calculate** (Down button) in the Parameters display. The values for electrical parameters (shown in the lower portion of the display) are computed from the physical values and the values in the Results display are updated.

Updating Schematics with LineCalc Results

After performing a synthesis or analysis, you can place the component to the Schematic window.

In the Schematic window, select **Tools** > **Linecalc** > **Place New Synchronized Component.**

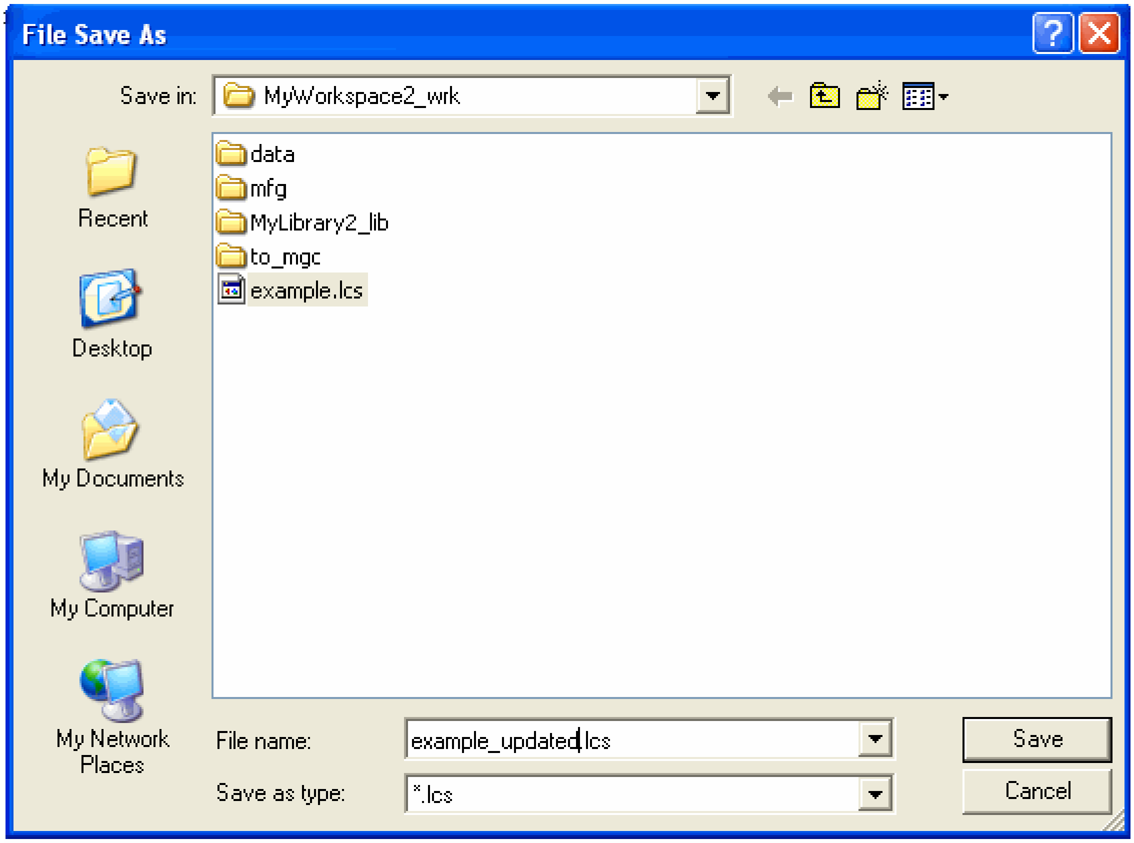
If you start LineCalc from the Schematic and send selected components to LineCalc for synthesis or analysis, you can update your schematic when your calculations are complete.

In the Schematic window, select the component to be updated. Then select **Tools > LineCalc > Update Selected Component From LineCalc**.

Saving Results

To save a LineCalc component file:

- Choose **File** > **Save** from the Synthesis/LineCalc window. The program automatically saves the filename with an *.lcs* extension. Unless you change the default data directory, the data file is saved in the program startup directory.



File name: example_updated.lcs

Save as type: *.lcs

To save a LineCalc component file to a different data file:

1. In the Synthesis/LineCalc window, choose **File** > **Save As**.
2. In the dialog box, enter a filename and specify a directory path.

To save a design that was modified by LineCalc calculations:

- In the Schematic or Layout window, choose **File** > **Save**.

Printing

You can print the data for a component:

- To specify a printing configuration, choose **File** > **Print Setup**.
- To print the data associated with the currently displayed component, choose **File** > **Print.**

The printed data format is:

product information

date information

component type

component ID

units information

frequency value

shared parameters

physical parameters

electrical parameters

result parameters

> **Note**
> For detailed information on print and print setup options, refer to your operating system documentation.

Exiting the Program

To exit LineCalc:

1. Choose **File** > **Exit.** If any unsaved changes are detected, you are prompted, *Do you want to save the design?*
2. Select **Yes** or **No** , as appropriate.

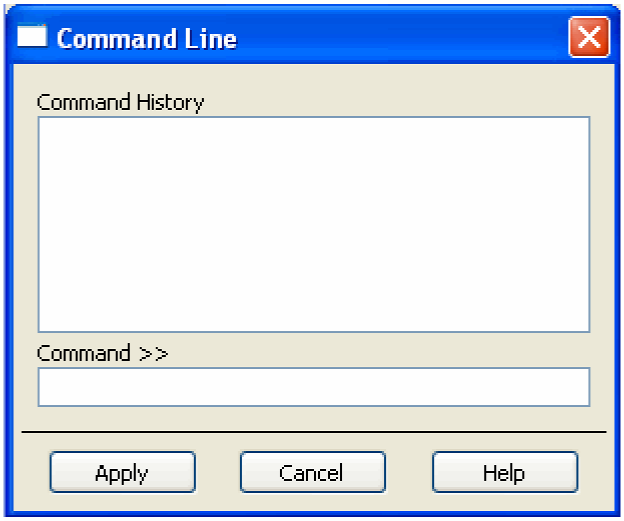
   If you select **Yes** , and the currently active file had not been saved previously (that is, labeled *untitled* at the top of the window), the Save As dialog box appears, prompting you for a filename.

Using AEL Commands

Using the program's Applications Extension Language (AEL), you can create entries for user-defined components, customize the contents of library and palette menus, and write customized commands. For configuration details on using AEL, see *AEL* (ael).

To execute AEL commands:

1. Choose **Options** > **Command Line** to open the Command Line dialog.



2. Type the command(s) in the **Command >>** field or select one of the commands in the Command History list. Click Apply to execute the command. You can double-click a command in the list to select and execute it.

> **Note**
> All commands entered in the **Command >>** field must be in correct AEL format.

Using Macros to Automate Tasks

Creating a Macro File

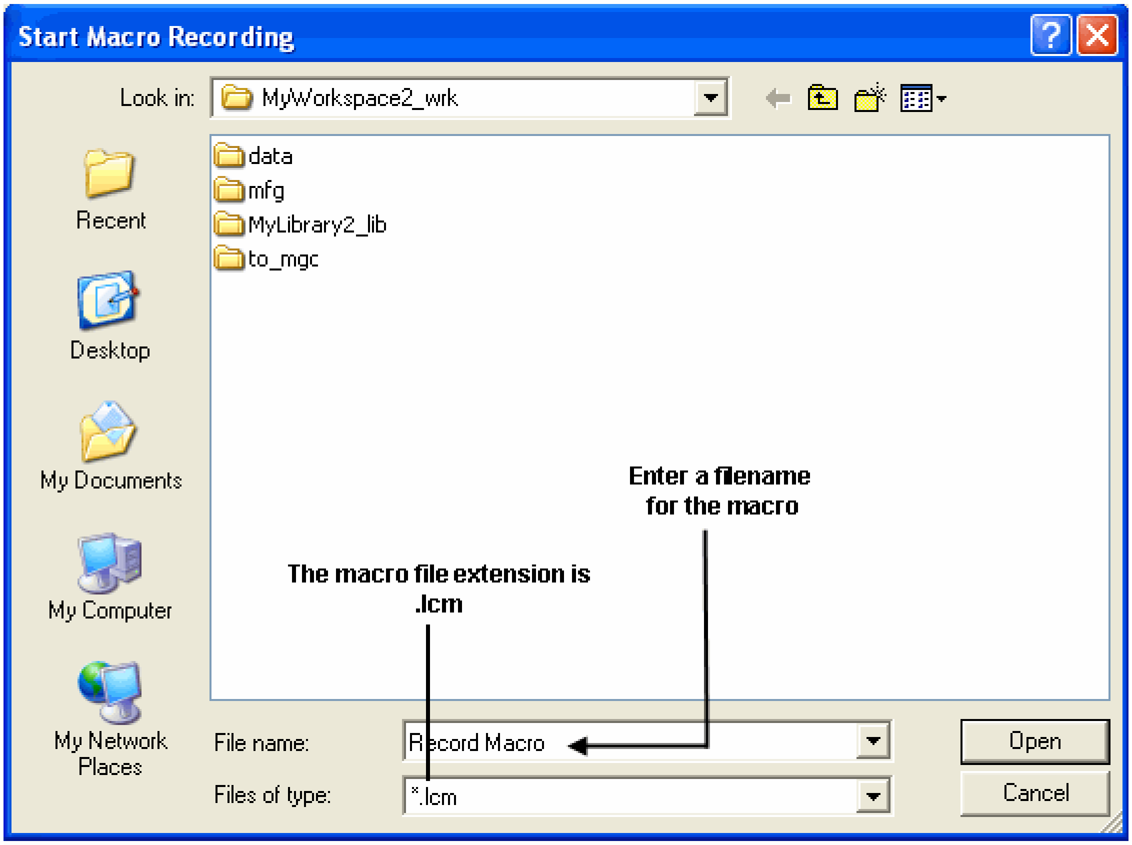
To create a macro file:

Using any text editor, create a file in one of your workspace directories and type the lines containing the AEL functions you want to execute. The filename must have the extension *.ael*.

Recording a Macro

To record a macro:

1. Choose **Options** > **Command Line** to open the Command Line dialog.
2. Select **Options** > **Start Record Macro**. In the dialog, enter a filename for the macro file you are going to record and click **OK** to close the dialog.


Start Macro Recording

Look in: MyWorkspace2_wrk

File name: Record Macro

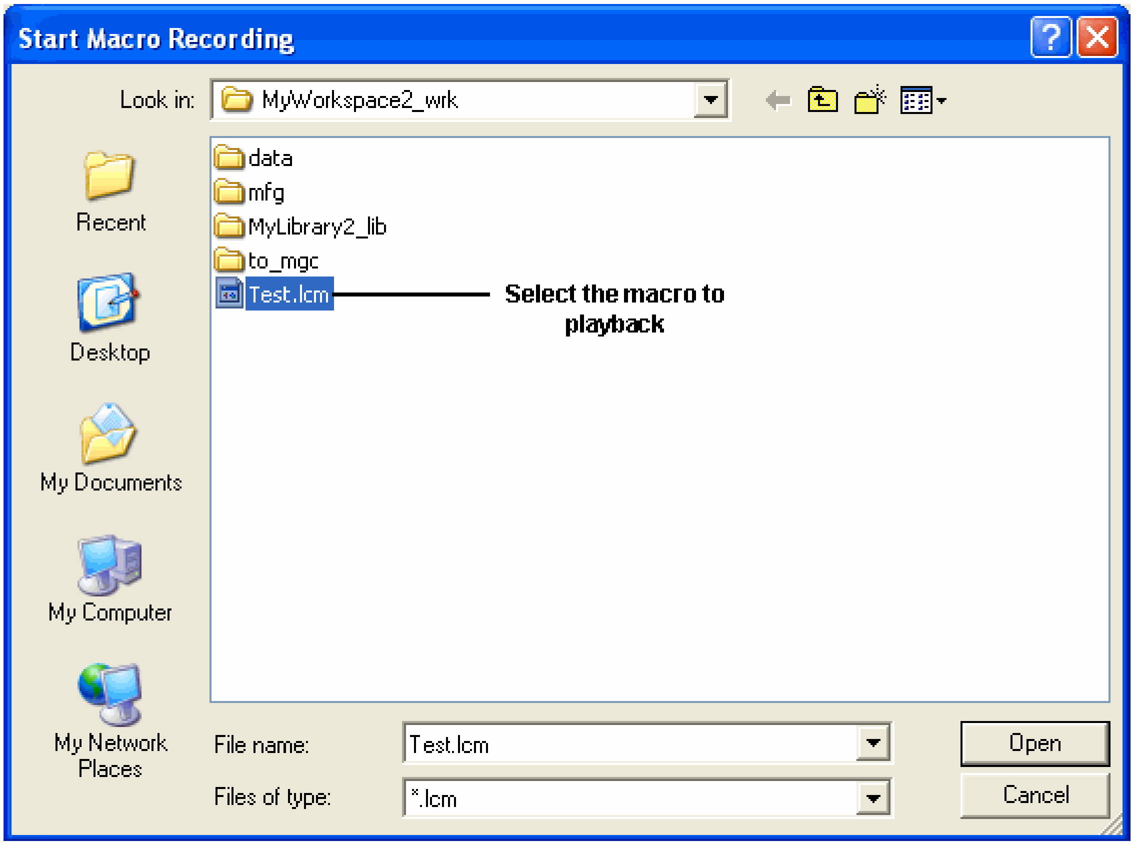
Files of type: *.lcm

3. In the Command Line dialog, the Command History displays all actions taken in the LineCalc window during the recording. Alternatively, enter the AEL functions you want to record in the **Command >>** field.
4. Select **Options** > **Stop Record Macro** to stop recording.

Running a Macro from the Design Environment

To play back a macro:

1. Choose **Options** > **Playback Macro**. The Start Macro Recording dialog displays all the *.lcm* files in the current workspace directory. You can browse to display macros from another directory.



Look in: MyWorkspace2_wrk

File name: Test.lcm

Files of type: *.lcm

Select the macro to playback

2. Select the macro you want to play back and click **Open** to execute the selected macro.

Running a Macro from the UNIX Command Line

To run a macro file from the UNIX command line:

- Change directories (cd) to your working directory and type:

```
$ linecalc -m <filename>.lcm
```

or

```
$ linecalc -m synthesize.lcm
```

LineCalc Components

Components and Parameters

Components and Associated Parameters lists the LineCalc-supported components and any associated parameters that can be modified or calculated. Components are grouped by substrates and details regarding their electrical and physical values, references, and other computed results are given.

Components and Associated Parameters

Component	Electrical Parameters	Physical Parameters	Substrate Parameters	Calculated Values
COAX	Z0 E_Eff	Di Do L	Er TanD Rho Sigma	A_DB
RWG	Z0 E_Eff	A B L	Er Rho TanD Mur TanM Sigma	K_Eff A_DB
Components using the CPWSUB substrate (CPWSUB = *)				
CPW	Z0 E_Eff	W G L	Er Mur H Hu T Cond TanD Rough	K_Eff A_DB SkinDepth
CPWCPL2	ZE ZO Z0 C_DB E_EffComponents using the SSUB substrate (SSUB = *)	W G S L	Er Mur H Hu T Cond TanD Rough	KE KO AE_DB AO_DB SkinDepth
CPWCPL4	ZE ZO Z0 C_DB E_Eff	W = Wi G S = Si L	Er Mur H Hu T Cond TanD Rough	K_Eff KE KO AE_DB AO_DB SkinDepth
CPWG	Z0 E_Eff	W G L	Er Mur H Hu T Cond TanD Rough	K_Eff A_DB SkinDepth
Components using the FSUB substrate (FSUB = *)				
BFINL	Z0 E_Eff	D L	Er Cond Fa Fb Fdw	K_Eff A_DB SkinDepth
IFINL	Z0 E_Eff	D L	Er Cond Fa Fb Fdw	K_Eff A_DB SkinDepth
UFINL	Z0 E_Eff	D L	Er Cond Fa Fb Fdw	K_Eff A_DB SkinDepth
Components using the MSUB substrate (MSUB = *)				
MCLIN	ZE ZO Z0 C_DB E_Eff	W S L	Er Mur H Hu T Cond TanD Rough	KE KO AE_DB AO_DB SkinDepth
MLANG	ZE ZO Z0 C_DB E_Eff	W S L	Er Mur H Hu T Cond TanD Rough	KE KO AE_DB AO_DB SkinDepth
MLANG6	ZE ZO Z0 C_DB E_Eff	W S L	Er Mur H Hu T Cond TanD Rough	KE KO AE_DB AO_DB SkinDepth
MLANG8	ZE ZO Z0 C_DB E_Eff	W S L	Er Mur H Hu T Cond TanD Rough	KE KO AE_DB AO_DB SkinDepth
MLIN	Z0 E_Eff	W L	Er Mur H Hu T Cond TanD Rough	K_Eff A_DB SkinDepth

Components using the SSUB substrate (SSUB = *)				
SBCLIN	ZE ZO Z0 C_DB E_Eff	W S L	Er Mur B T Cond TanD	AE_DB AO_DB SkinDepth
SCLIN	ZE ZO Z0 C_DB E_Eff	W S L	Er Mur B T Cond TanD	AE_DB AO_DB SkinDepth
SLIN	Z0 E_Eff	W L	Er Mur B T Cond TanD	A_DB SkinDepth
SLINO	Z0 E_Eff	W S L	Er Mur B S T Cond TanD	A_DB SkinDepth
SOCLIN	ZE ZO Z0 C_DB E_Eff	W WO S L	Er Mur B T Cond TanD	AE_DB AO_DB SkinDepth
Components using the SSSUB substrate (SSSUB = *)				
SSCLIN	ZE ZO Z0 C_DB E_Eff	W S L	Er Mur H Hu Hl T Cond TanD Rough	KE KO AE_DB AO_DB SkinDepth
SSLIN	Z0 E_Eff	W L	Er Mur H Hu Hl T Cond TanD Rough	K_Eff A_DB SkinDepth

References/Shared Parameters

References/Shared Parameters lists the LineCalc-supported references/shared parameters.

References/Shared Parameters

References/Shared Parameters	**Parameters**	
	Parmeter Name	**Default Value**
CPWSUB	Er H T Cond Rough TanD Hu Mur COND	10.0 25.0 0.150 4.1e7 0.0 0.0 3.9e+07 mil 1.0 Cond
FSUB	Er Fdw Fa Fb Cond	2.2 62.5 mil 900.0 mil 400.0 mil 5.8e7
MSUB	Er H T Cond Cond1 Cond2 Rough TanD Hu Mur Diel1 Diel2 Hole Res	9.6 10 mil 0.150 4.1e7 Cond Cond1 0.0 0.0 3.9e+34m 1.0 Diel Diel1 hole resi
SSSUB	Er H T Cond Rough HU HL TanD Mur COND	10.0 25.0 0.150 4.1e7 0.0 100.0 100.0 0.0 1.0 Cond
SSUB	Er B T Cond Cond1 Cond2 TanD Mur	2.5 62.5 0.150 5.8e7 Cond Con1 0.0 1.0
Note: The default MSUB Cond value has changed to be 4.1e7 (the conductivity of gold).		

Parameter Definitions

Parameter	Definition
A	Width of rectangular waveguide
A_DB	Total attenuation of the structure in dB
AE_DB	Total even mode attenuation of coupled section in dB
AO_DB	Total odd mode attenuation of coupled section in dB
B	Ground plane spacing from top to bottom (stripline) or height of rectangular waveguide
C_DB	Coupling factor in dB, i.e., C(dB) = 20log10 [(ZE-ZO)/(ZE+ZO)]
D	Gap width in length units
DI	Inner diameter (COAX)
DO	Outer diameter (COAX)
E	Electrical length of line or coupled section (angle units)
E_EFF	Effective electrical length of line or coupled section (angle units)
E_MEAN	Mean electrical length of coupled microstrip or coplanar waveguide section (angle units)
ER	Substrate relative dielectric constant
G	Gap between conductor and ground planes (CPW, CPWG)
H	Substrate thickness in length units
HL	Lower ground plane to substrate spacing in length units
HU	Upper ground plane to substrate spacing in length units
HC	Distance from top cover to top of substrate in length units
KE	Even-mode effective dielectric constant: KE=[C/Vp(even mode)]2
K_EFF	Effective dielectric constant: K_EFF=(C/Vp)2
KO	Odd-mode effective dielectric constant: KO=[C/Vp(odd mode)]2
L	Length of line or coupled section
MUR	Relative permeability
Rough	RMS surface roughness in length units
S	Spacing between lines
SI	Spacing between inner center conductors (CPWCPL4)
SIGM	Dielectric conductivity value in Siemens per meter
T	Metal thickness
TAND	Dielectric loss tangent
TANM	Magnetic loss tangent
TEMP	Temperature; can not be modified by the user
W	Width of line
WI	Width of inner center conductors (CPWCPL4)
WO	Offset of coupled lines for offset coupled stripline
Ze	Impedance for even-mode
Zo	Impedance for odd-mode
Z0	For coupled-line components, such as SCLIN or MCLIN, Z0=sqrt(Ze × Zo)

Changing Parameter Defaults

The default parameters that appear in the LineCalc window for editing are set in a file called *lcuiinit* in the directory *$HPEESOF_DIR/linecalc/lib* (where *$HPEESOF_DIR* represents the complete path for your installation). You can make a local copy of this file and change the default parameters so that less editing is required whenever you use LineCalc.

To modify the *lcuiinit* file:

1. Copy the file *lcuiinit* to a directory of your choosing. Using any text editor, make the desired changes. Save the file.
2. Now you need to tell the program where to look for your customized file. Make a local copy of the LineCalc configuration file *linecalc.cfg* found in *$HPEESOF_DIR/config.* Copy this file to *$HOME/hpeesof/config*.
3. Using any text editor, locate the variable *LCALCUI_INIT_FILE* and modify the path to point to the location of your modified *lcuiinit* file.

> **Note**
> On the PC, *$HOME* represents the path you specified as the Home Folder during installation (*C:\users\default* by default); $ *HPEESOF_DIR* represents the path you specified as your Program Folder during installation (*C:\AdvDesSys*).

The component definition format is:

```
ELEMENT <element_name> <element_id> <# of ports> <units> <physical>
[<physical_fixed> <physical>] <subst> <tand> <sigma> <temp> <electrical>
[<electrical_fixed><electrical>]<result>
```

For example:

```
ELEMENT COAX COAX_DEFAULT 4 UNITS PHYSICAL PHYSICAL_FIXED
PHYSICAL SUBST ELECTRICAL RESULT UNITS UNITS_DEFAULT
FREQ_VAL=10.00 FREQ=GHz LENGTH=mil RES=ohm ANGLE=deg
PHYSICAL DI=37.00
PHYSICAL_FIXED DO=90.00
PHYSICAL L=650.00
SUBST COAXSUB=COAXSUB_DEFAULT ER=2.10 TAND=0.0003 RHO=1.0
ELECTRICAL Z0=35.00 E=90.00
RESULT A_DB=0.01
```

LineCalc Error Messages

This section describes the possible error messages and how to handle each one of them.

Syntax error in AEL file

Comment: An error has occurred in an AEL file.

Category: Syntax error in AEL file

Resolution: Check the AEL file, confirming that all the parentheses are matched, there are no missing ";", etc.

AEL files not loaded

Comment: The AEL files are not loaded due to error(s) occurred.

Category: Syntax error in AEL file

Resolution: Check the AEL file, confirming that all the parentheses are matched, there are no missing ";", etc.

Invoke process failed

Comment: Unable to invoke process (could be the LineCalc Engine, Help Server, etc.)

Category: Error in the initialization phase

Resolution: Confirm that all the necessary tools (LineCalc Engine, Status Server, Help Server, and Hardcopy Server) are available and functional.

Unable to connect to the Status Server

Comment: The LineCalc User Interface cannot connect to the Status Server process.

Category: Error in the initialization phase.

Resolution: Confirm that all the necessary tools (LineCalc Engine, Status Server, Help Server, and Hardcopy Server) are available and functional.

AEL command not found

Comment: The specified AEL command is not found

Category: Invalid AEL command.

Resolution: Confirm that the command is spelled properly and is indeed a supported function.

Incomplete config file

Comment: Some information in the configuration file, linecalc.cfg, is missing.

Category: Error in configuration file.

Resolution: Search the *linecalc.cfg* file to find the required information that is missing and replace it. Without this information, the program cannot function.

Init file not found

Comment: The initialization file, lcuiinit, is not found.

Category: Error in the initialization file.

Resolution: The initialization file must be placed where the program can read it. This is usually *$HPEESOF_DIR/lib/linecalc* . Confirm that all the information is there: the users can add more information to it, but the user should never delete any information from the original initialization file.

Syntax error in init file

Comment: A syntax error in the init file, *lcuiinit* , has been found.

Category: Error in the initialization file.

Resolution: The initialization file must be placed where the program can read it. This is usually *$HPEESOF_DIR/lib/linecalc* . Confirm that all the information is there: the users can add more information to it, but the user should never delete any information from the original initialization file.

Missing units information

Comment: Units information is not found.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing substrate information

Comment: Substrate information is not found.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing tand information

Comment: Tand information is not found.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing mcover information

Comment: Mcover information is not found.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing mwall information

Comment: Mwall information is not found.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing perm information

Comment: Perm information is not found.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing sigma information

Comment: Sigma information is not found.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing temp information

Comment: Temp information is not found.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing parameters information

Comment: Some parameters are missing.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Missing number of nodes information

Comment: The number of nodes information is missing.

Category: Error in data file.

Resolution: Confirm that all information exists for a given component in the data file.

Save element type failed

Comment: Unable to save the component type.

Category: Invalid component type to be saved.

Resolution: Confirm that the component type is one of the supported LineCalc components.

Save element ID failed

Comment: Unable to save the component ID.

Category: Invalid component ID to be saved.

Resolution: This is very unlikely to happen. However if it does happen, it is most likely due to unable to allocate more memory to store the information.

Save number of nodes information failed

Comment: Unable to save the number of nodes information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save units information failed

Comment: Unable to save the units information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save freq unit failed

Comment: Unable to save the frequency unit information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save length unit failed

Comment: Unable to save the length unit information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save res unit failed

Comment: Unable to save the resistor unit information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save angle unit failed

Comment: Unable to save the angle unit information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save substrate information failed

Comment: Unable to save the substrate information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save mwall information failed

Comment: Unable to save the mwall information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save shared parameters failed

Comment: Unable to save the shared parameters information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save physical parameters information

Comment: Unable to save the physical parameters information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save electrical parameters information

Comment: Unable to save the electrical parameters information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Save result parameters failed

Comment: Unable to save the result parameters information.

Category: Unable to save the specified component.

Resolution: If this error occurs, the program is probably unable to allocate more memory to save the information. Another issue which needs to be looked at is that the information should be in the proper format: if the format is expected to be a number, it should be a number; if the format is expected to be a string, it should be a string type.

Invalid element type

Comment: Invalid component type.

Category: Invalid component type.

Resolution: An invalid component type has been requested. Confirm that the component type is spelled correctly is indeed a LineCalc supported component.

Unable to write the data file

Comment: Unable to save to the specified data file.

Category: Unable to write the data file.

Resolution: If this error occurs, the current user probably cannot write to the specified data file; this can also occur if the directory is not writable by the current user.

Create new shared parameters failed

Comment: Unable to create a new shared parameters information.

Category: Unable to create a new shared parameters.

Resolution: When this error occurs, the program is probably unable to allocate more memory to store the data.

Invalid substrate

Comment: Invalid substrate.

Category: Invalid substrate type.

Resolution: Confirm the substrate type entered is supported by LineCalc.

Error Retrieving DONE NETLIST

Comment: Invalid parameters for analysis/synthesis.

Category: Invalid parameters for analysis/synthesis.

Resolution: Ensure the electrical, physical, and substrate parameter values are within the range of usage, per user guide.

Inconsistent Electrical Parameter Value. Ze and Zo must be of the same sign.

Comment: Ze and Zo must be of the same sign.

Category: Ze and Zo must be of the same sign.

Resolution: Use the same sign for both Ze and Zo.

LineCalc AEL Functions

This section describes the supported AEL functions for the LineCalc User Interface.

- *lcuiuser analyze()* (linecalc)
- *lcuiuser elemid select()* (linecalc)
- *lcuiuser elemid select apply()* (linecalc)
- *lcuiuser elemtype select ()* (linecalc)
- *lcuiuser elemtype select apply()* (linecalc)
- *lcuiuser file chdir ok()* (linecalc)
- *lcuiuser file exit()* (linecalc)
- *lcuiuser file new()* (linecalc)
- *lcuiuser file open ok()* (linecalc)
- *lcuiuser file print()* (linecalc)
- *lcuiuser file print setup()* (linecalc)
- *lcuiuser file save()* (linecalc)
- *lcuiuser file save as ok()* (linecalc)
- *lcuiuser option cmdline apply()* (linecalc)
- *lcuiuser option pbmacro ok()* (linecalc)
- *lcuiuser set angle unit()* (linecalc)
- *lcuiuser set elec res unit()* (linecalc)
- *lcuiuser set electrical parms()* (linecalc)
- *lcuiuser set freq unit()* (linecalc)
- *lcuiuser set freq value()* (linecalc)
- *lcuiuser set length unit()* (linecalc)
- *lcuiuser set mwall length unit()* (linecalc)
- *lcuiuser set physical parms()* (linecalc)
- *lcuiuser set physical parms all()* (linecalc)
- *lcuiuser set physical parms fix()* (linecalc)
- *lcuiuser set phys length unit()* (linecalc)
- *lcuiuser set res unit()* (linecalc)
- *lcuiuser set shared parms()* (linecalc)
- *lcuiuser set subst length unit()* (linecalc)
- *lcuiuser set subst parm value()* (linecalc)
- *lcuiuser simulation analyze()* (linecalc)
- *lcuiuser simulation interrupt()* (linecalc)
- *lcuiuser simulation stop()* (linecalc)
- *lcuiuser simulation synthesize()* (linecalc)
- *lcuiuser synthesize()* (linecalc)

lcuiuser_analyze()

Computes the electrical parameters from the physical parameter values. In addition, the result parameters are computed. Returns: None.
See also: *lcuiuser_synthesize* (linecalc)(), *lcuiuser_simulation_analyze* (linecalc)(), *lcuiuser_simulation_synthesize* (linecalc)()

Syntax:

```
defun lcuiuser_analyze()
```

Example

```
lcuiuser_analyze();
```

lcuiuser_elemid_select()

Displays the dataset belonging to the specified component ID, elemID. Returns: None. See also: *lcuiuser_elemtype_select* (linecalc)()

Syntax:

```
defun lcuiuser_elemid_select( elemID )
```

where
elemID is the component ID to be displayed.

Example

```
lcuiuser_elemid_select( "MLIN_DEFAULT");
```

lcuiuser_elemid_select_apply()

Obsolete; see *lcuiuser_elemid_select* (linecalc)().

lcuiuser_elemtype_select ()

Displays the dataset belonging to the specified component type, elemType. If there are multiple components in the memory for a given component type, the first component type found from the list is displayed. To display a component type that has a specified ID, select the component by ID by calling the *lcuiuser_elemid_select_apply* (linecalc) function.
Returns: None.
See also: *lcuiuser_elemid_select* (linecalc)()

Syntax:

defun lcuiuser_elemtype_select (elemType)

where
elemType is the component type to be displayed.

Example

```
lcuiuser_elemtype_select("MLIN");
```

lcuiuser_elemtype_select_apply()

Obsolete; see *lcuiuser_elemid_select* (linecalc).

lcuiuser_file_chdir_ok()

Performs a change directory command to the specified directory, dirName. If a local configuration file (linecalc.cfg) exists, the function reads in the information and, if needed, overwrites any default values. If the current component has been modified and has not been saved, a question dialog box opens asking if the current component should be saved prior to performing a change directory command. Returns: None.
See also: *lcuiuser_file_save* (linecalc)(), *lcuiuser_file_save_as_ok* (linecalc)()

Syntax:

```
defun lcuiuser_file_chdir_ok(dirName)
```

where
dirName is the directory to go to

Example

```
lcuiuser_file_chdir_ok("dirName1");
```

lcuiuser_file_exit()

Exits the LineCalc User Interface program. If the current component set is not saved, a dialog box opens asking if the data should be saved prior to exiting the program. If no data needs to be saved, a different dialog box opens asking to confirm the exit command.
Returns: None.
See also: *lcuiuser_file_save* (linecalc)(), *lcuiuser_file_save_as_ok* (linecalc)()

Syntax:

```
defun lcuiuser_file_exit()
```

Example

```
lcuiuser_file_exit();
```

lcuiuser_file_new()

Displays the default component. The default component is the first component being displayed when the LineCalc User Interface is invoked. If the component is not saved yet, a dialog box opens so that the component can be saved prior to switching to a different component. Returns: None.
See also: *lcuiuser_file_save* (linecalc)(), *lcuiuser_file_save_as_ok* (linecalc)()

Syntax:

```
defun lcuiuser_file_new()
```

Example

```
lcuiuser_file_new();
```

lcuiuser_file_open_ok()

Opens the data file specified by filename and displays the data properly. If the current component is not saved yet, a dialog box opens so that the data can be saved before moving on to a different component set. Returns: None.
See also: *lcuiuser_file_save* (linecalc)(), *lcuiuser_file_save_as_ok* (linecalc)()

Syntax:

```
defun lcuiuser_file_open_ok(filename)
```

where
filename is filename to be opened

Example

```
lcuiuser_file_open_ok("dirName1/linecalc1.lcs");
or
lcuiuser_file_open_ok("linecalc1.lcs");
```

lcuiuser_file_print()

Sends the displayed data to a printer using the program's *Hardcopy Server* . Returns: None.
See also: *lcuiuser_file_print_setup* (linecalc)()

Syntax:

defun lcuiuser_file_print()

Example

lcuiuser_file_print();

lcuiuser_file_print_setup()

Opens the print setup dialog box to set the printer selections using the program's *Hardcopy Server* . Returns: None.
See also: *lcuiuser_file_print* (linecalc)()

Syntax:

```
defun lcuiuser_file_print_setup()
```

Example

```
lcuiuser_file_print_setup();
```

lcuiuser_file_save()

See also: *lcuiuser_file_save_as_ok* (linecalc)()

Syntax:

defun lcuiuser_file_save()

> Saves the data to the current data filename. If there is no associated data file for the current component or if write permission is not set for the current data file, opens the File Selection dialog box to either select or enter the filename to save the data. Returns: None.

Example

```
lcuiuser_file_save();
```

lcuiuser_file_save_as_ok()

Saves the currently displayed component to a different file, specified by filename.
Returns: None.
See also: *lcuiuser_file_save* (linecalc)()

Syntax:

```
defun lcuiuser_file_save_as_ok(filename)
```

where
filename is filename to be saved

Example

```
lcuiuser_file_save_as_ok("filename1.lcs");
```

lcuiuser_option_cmdline_apply()

Executes the AEL command string, cmdString. Returns: None.
See also: *lcuiuser_option_pbmacro_ok* (linecalc)()

Syntax:

```
defun lcuiuser_option_cmdline_apply(cmdString)
```

where
cmdString is AEL command string to be executed

Example

```
lcuiuser_option_cmdline_apply("lcuiuser_file_exit()");
```

lcuiuser_option_pbmacro_ok()

Executes the specified AEL macro file, filename, which contains of a set of supported AEL functions. Returns: None.
See also: *lcuiuser_option_cmdline_apply* (linecalc)()

Syntax:

defun lcuiuser_option_pbmacro_ok(filename)

where
filename is AEL macro file to be executed

Example

lcuiuser_option_pbmacro_ok("macro1.ael");

lcuiuser_set_angle_unit()

Sets the angle unit to the specified string, angleUnitString. Note that angleUnitString must be one of the unit strings supported by LineCalc. The supported angle units are: deg, and rad. Returns: None.
See also: *lcuiuser_set_freq_unit* (linecalc)(), *lcuiuser_set_freq_value* (linecalc)(), *lcuiuser_set_length_unit* (linecalc)(), *lcuiuser_set_res_unit* (linecalc)()

Syntax:

```
defun lcuiuser_set_angle_unit(angleUnitString)
```

where
angleUnitString is the new angle unit string

Example

```
lcuiuser_set_angle_unit("deg");
```

lcuiuser_set_elec_res_unit()

Sets the unit of specified electrical parameter parmName, to the specified string, elecUnit. Note that elecUnit must be one of the unit strings supported by LineCalc. The supported units for electrical resistance parameters are: ohm and kohm. The supported units for electrical angle parameters are: deg, and rad. Returns: None.
See also: *lcuiuser_set_mwall_length_unit* (linecalc) (), *lcuiuser_set_phys_length_unit* (linecalc)(), *lcuiuser_set_subst_length_unit* (linecalc) ().

Syntax:

```
defun lcuiuser_set_elec_res_unit ( parmName, elecUnit )

    where
    parmName is electrical parameter name of unit to be modified
    elecUnit is new electrical unit string
```

Example

```
lcuiuser_set_elec_res_unit ("E_Eff", "rad");
lcuiuser_set_elec_res_unit("Z0", "kohm");
```

lcuiuser_set_electrical_parms()

Sets the electrical parameter value of parameter name, parmName, to parmValue. parmName must be one of the valid electrical parameter for the currently displayed component. Returns: None.
See also: *lcuiuser_set_physical_parms* (linecalc)(), *lcuiuser_set_physical_parms_fix* (linecalc)()

Syntax:

```
defun lcuiuser_set_electrical_parms(parmName, parmValue)
```

where
parmName is electrical parameter name to be modified
parmValue is new electrical parameter value

Example

```
lcuiuser_set_electrical_parms("E_EFF", "360");
```

lcuiuser_set_freq_unit()

Sets the frequency unit string value, freqUnitString. Note that freqUnitString must be one of the unit strings supported by LineCalc. The supported frequency units are: GHz, MHz, kHz, and Hz. Returns: None.
See also: *lcuiuser_set_angle_unit* (linecalc)(), *lcuiuser_set_freq_value* (linecalc)(), *lcuiuser_set_length_unit* (linecalc)(), *lcuiuser_set_res_unit* (linecalc)()

Syntax:
defun lcuiuser_set_freq_unit(freqUnitString)

where
freqUnitString is the new frequency unit string

Example
lcuiuser_set_freq_unit("GHz");

lcuiuser_set_freq_value()

Sets the frequency unit value, freqUnitValue. It should be passed as a string value.
Returns: None.

See also: *lcuiuser_set_angle_unit()* (linecalc), *lcuiuser_set_freq_unit()* (linecalc), *lcuiuser_set_length_unit()* (linecalc), *lcuiuser_set_res_unit()* (linecalc)

Syntax:

```
defun lcuiuser_set_freq_value(freqUnitValue)
```

where
freqUnitValue is the new frequency unit value

Example

```
lcuiuser_set_freq_value("10.00");
```

lcuiuser_set_length_unit()

Sets the length unit string value, lengthUnitString. Note that lengthUnitString must be one of the unit strings supported by LineCalc. The supported length units are: in, mil, m, cm, mm, and um. Returns: None.
See also: *lcuiuser_set_angle_unit* (linecalc)(), *lcuiuser_set_freq_unit* (linecalc)(), *lcuiuser_set_res_unit* (linecalc)()

Syntax:

```
defun lcuiuser_set_length_unit(lengthUnitString)
```

where
lengthUnitString is the new length unit string

Example

```
lcuiuser_set_length_unit("mil");
```

lcuiuser_set_mwall_length_unit()

Sets the unit of specified mwall parameter parmName, to the specified string, mwallUnit. Note that mwallUnit must be one of the unit strings supported by LineCalc. The supported length units for mwall length parameters are: mil, um, mm, cm, meter, in, and ft.
Returns: None.
See also: *lcuiuser_set_phys_length_unit* (linecalc) (), *lcuiuser_set_subst_length_unit* (linecalc)(), *lcuiuser_set_elec_res_unit* (linecalc) ().

Syntax:

defun lcuiuser_set_mwall_length_unit (parmName, mwallUnit)

where
parmName is mwall parameter name of unit to be modified
mwallUnit is new mwall unit string

Example

```
lcuiuser_set_mwall_length_unit ( "Wall1", "mil");
lcuiuser_set_mwall_length_unit ("Wall2", "um");
```

lcuiuser_set_physical_parms()

Sets the physical parameter value of the specified parameter name, parmName, to parmValue. parmName must be one of the valid physical parameter for the currently displayed component. Returns: None.
See also: *lcuiuser_set_electrical_parms* (linecalc)(), *lcuiuser_set_physical_parms_fix* (linecalc)()

Syntax:

```
defun lcuiuser_physical_parms(parmName, parmValue)
```

where
parmName is physical parameter name to be modified
parmValue is new physical parameter value

Example

```
lcuiuser_set_physical_parms("W", "25.00");
```

lcuiuser_set_physical_parms_all()

Obsolete.

lcuiuser_set_physical_parms_fix()

Sets the physical parameter fixed value of the specified parameter name, parmName, to fixValue. fixValue should be either 0 or 1. parmName must be one of the valid physical parameter for the currently displayed component. Returns: None.
See also: *lcuiuser_set_electrical_parms* (linecalc)(), *lcuiuser_set_physical_parms* (linecalc)()

Syntax:

defun lcuiuser_set_physical_parms_fix(parmName, fixValue)

where
parmName is physical parameter name to be modified
fixValue is new fixed physical parameter(either 0 or 1)

Example

```
lcuiuser_set_physical_parms_fix("W", 0);
```

lcuiuser_set_phys_length_unit()

Sets the unit of specified physical parameter parmName, to the specified string, physUnit. Note that physUnit must be one of the unit strings supported by LineCalc. The supported length units for physical length parameters are: mil, um, mm, cm, meter, in, and ft.
Returns: None.
See also: *lcuiuser_set_mwall_length_unit* (linecalc) (), *lcuiuser_set_subst_length_unit* (linecalc)(), *lcuiuser_set_elec_res_unit* (linecalc)().

Syntax:

defun lcuiuser_set_phys_length_unit (parmName, physUnit)

where
parmName is physical parameter name of unit to be modified
physUnit is new physical unit string

Example

lcuiuser_set_phys_length_unit ("W", "mil");

lcuiuser_set_res_unit()

Sets the resistor unit string value, resUnitString. Note that it must be one of the unit strings supported by LineCalc. The supported resistor units are: kohm and ohm. Returns: None.
See also: *lcuiuser_set_angle_unit* (linecalc)(), *lcuiuser_set_freq_unit* (linecalc)(), *lcuiuser_set_length_unit* (linecalc)()

Syntax:

defun lcuiuser_set_res_unit(resUnitString)

where
resUnitString is the new resistor unit string

Example

```
lcuiuser_set_res_unit("ohm");
```

lcuiuser_set_shared_parms()

Obsolete; see *lcuiuser_set_subst_parm_value* (linecalc)().

lcuiuser_set_subst_length_unit()

Sets the unit of specified shared parameter parmName, to the specified string, substUnit. Note that substUnit must be one of the unit strings supported by LineCalc. The supported length units for shared length parameters are: mil, um, mm, cm, meter, in, and ft.
Returns: None.
See also: *lcuiuser_set_mwall_length_unit* (linecalc) (), *lcuiuser_set_phys_length_unit* (linecalc)(), *lcuiuser_set_elec_res_unit* (linecalc) ().

Syntax:

defun lcuiuser_set_subst_length_unit (parmName, substUnit)

where
parmName is shared parameter name of unit to be modified
substUnit is new shared unit string

Example

lcuiuser_set_subst_length_unit ("Hu", "mil");

lcuiuser_set_subst_parm_value()

Sets the shared parameter value of the parameter name, parmName, to parmValue. parmName must be one of the valid shared parameter for the currently displayed component. Shared parameters are substrate, tand, mcover, mwall, perm, and sigma parameters. In addition, shared parameters for COAX and RWG components include those listed in *References/Shared Parameters".* (linecalc) Returns: None.

Syntax:

```
defun lcuiuser_set_subst_parm_value(parmName, parmValue)
```

where
parmName is shared parameter name to be modified
parmValue is new shared parameter value

Example

```
lcuiuser_set_subst_parm_value("ER", "9.91");
```

lcuiuser_simulation_analyze()

Computes the electrical parameters from the physical parameter values. In addition to computing the electrical parameters, the result parameters are computed. Returns: None. See also: *lcuiuser_analyze* (linecalc)(), *lcuiuser_synthesize* (linecalc)(), *lcuiuser_simulation_synthesize* (linecalc)()

Syntax:

```
defun lcuiuser_simulation_analyze()
```

Example

```
lcuiuser_simulation_analyze();
```

lcuiuser_simulation_interrupt()

Interrupts the simulation. LineCalc remains available for future simulation usage. Returns: None.
See also: *lcuiuser_simulation_stop* (linecalc)()

Syntax:

defun lcuiuser_simulation_interrupt()

Example

lcuiuser_simulation_interrupt();

lcuiuser_simulation_stop()

Interrupts the simulation process and the LineCalc engine becomes unavailable. If a user requests another simulation to be done, the LineCalc engine is invoked automatically.
Returns: None.
See also: *lcuiuser_simulation_interrupt* (linecalc)()

Syntax:

```
defun lcuiuser_simulation_stop()
```

Example

```
lcuiuser_simulation_stop();
```

lcuiuser_simulation_synthesize()

Computes the physical parameters from the electrical parameter values. In addition to computing the physical parameters, the result parameters are computed. Returns: None. See also: *lcuiuser_analyze* (linecalc)(), *lcuiuser_synthesize* (linecalc)(), *lcuiuser_simulation_analyze* (linecalc)()

Syntax:

```
defun lcuiuser_simulation_synthesize()
```

Example

```
lcuiuser_simulation_synthesize();
```

lcuiuser_synthesize()

Computes the physical parameters from the electrical parameter values. In addition to computing the physical parameters, the result parameters are computed. Returns: None. See also: *lcuiuser_analyze* (linecalc)(), *lcuiuser_simulation_analyze* (linecalc)(), *lcuiuser_simulation_synthesize* (linecalc)()

Syntax:

```
defun lcuiuser_synthesize()
```

Example

```
lcuiuser_synthesize();
```