

4 Getting Started with LVS

- [Launching LVS](#) 3-146
- [Input and Output Files](#) 3-147
- [User Interface](#) 3-149
- [Using LVS in Batch Mode](#) 3-185
- [Tutorial](#) 3-188

LVS stands for *layout versus schematic*. This netlist comparison tool compares two netlists to determine whether they describe the same circuit. When they do not, LVS works in conjunction with L-Edit to identify and correct errors.

LVS can be used to determine whether a schematic circuit matches a layout, or whether two different schematics or layouts implement the same circuit.

LVS has the following features:

- SPICE input format LVS accepts standard SPICE-format netlists, such as those produced by NetTran and L-Edit.
- Fragmentation identification When two netlists are not equivalent, LVS can identify unresolvable nodes and devices and assist in locating them on the original schematic or layout.
- Automorphism resolution LVS identifies *automorph classes*—sets of elements or nodes (such as devices in parallel) which cannot be distinguished from one another. To resolve automorph classes, LVS can either employ user-supplied *prematch* information or a detailed *trial matching* process.

- Parameter comparison
LVS uses *topological* (device types, number of connections), *parametric* (resistance, capacitance), and *geometric* (area, length, width) information to compare netlists. Slew rates can be defined to specify how different two values can be while still comparing equivalently. Different margins can be defined for parametric and geometric comparisons.
- Permuted class resolution
LVS can identify the switching of two elements in series.

Launching LVS

You can launch LVS by one of five methods:

- Clicking the **Start** button on the Windows toolbar and navigating to the application through the **Programs** menu
- Double-clicking the LVS icon on your desktop
- Double-clicking an LVS verification database (**.vdb**) file in Windows Explorer
- Double-clicking an LVS batch (**.bat**) file in Windows Explorer (see [Using LVS in Batch Mode on page 3-185](#))
- Invoking LVS from a DOS command prompt (see [LVS Command-Line Syntax on page 3-241.](#))

The LVS icon looks like this:



Input and Output Files

LVS requires two SPICE-format netlist files for input. SPICE files can be in either T-Spice or P-Spice format, but both input files must be of the same type. In T-Spice mode, LVS also accepts netlists in H-Spice (Berkeley Spice) format.

Optional input files include *prematch files* and *element description files*. A prematch file lists elements or nodes that may be identical, and their listing in the prematch file can prevent the formation of an automorph class. For further information on prematch files, see [Prematch File Format on page 3-266](#).

An element description file defines nonprimitive SPICE devices present in the netlists. For further information on element description files, see [Element Description File Format on page 3-261](#).

LVS writes the verification results to an optional output file with the extension **.out**. The program can also produce an optional node and element list with the extension **.lst**.

Information required for the verification run—input and output files, plus verification options—is referred to as a *verification setup*. LVS saves a setup to a verification database (**.vdb**) file. Multiple verification setups can be exported to a batch (**.bat**) file for subsequent invocation from a DOS command prompt.

File Locking

LVS reads in all input files, including layout and schematic netlist files, element description files, and prematch files, at the start of a verification run. You can open such files for reading at this time, but LVS will prevent you from editing them. The verification database (**.vdb**) file itself is locked throughout the verification, and you cannot edit it at any time during the run.

LVS also locks input files while processing each verification setup in a queue or batch file.

Backup Files

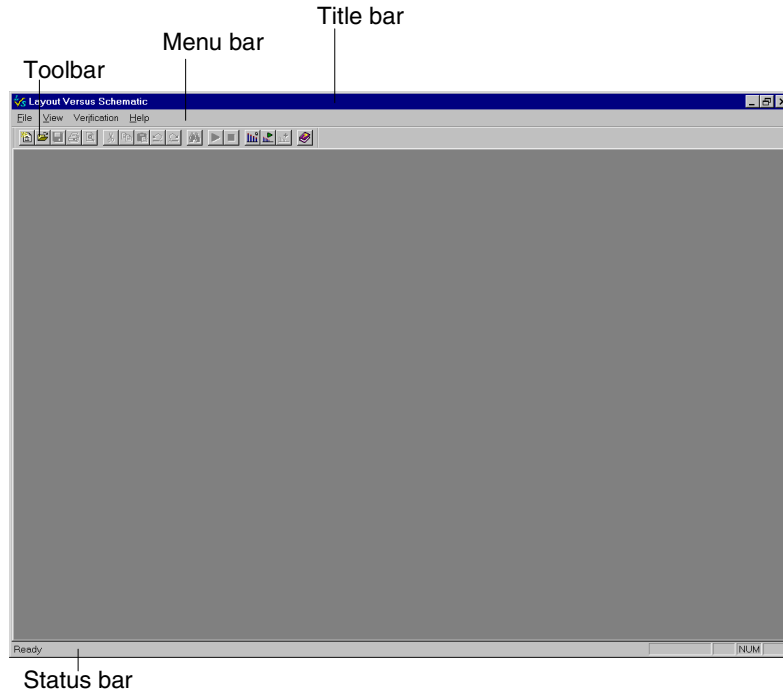
LVS automatically creates a backup of each verification database file, using the same filename as the original, but with a **.vdo** extension. Each time you modify the **.vdb** file, LVS will overwrite the previous backup of that file.

User Interface

The basic LVS user interface (shown below) consists of the following elements:

- Title bar
- Menu bar
- Toolbar (optional)
- Status bar (optional)

By default, the user interface will appear as in the following illustration:



The user interface may also contain the following other elements, depending on the types of files that are currently open and the task being performed:

- [Setup Window on page 3-156](#)
- [Text Window on page 3-171](#)
- [Verification Window on page 3-177](#)
- [Verification Queue on page 3-181](#)

For additional information on a particular interface element, see the appropriate section.

Menus

The horizontal space at the top of the application interface contains the *Menu bar*. The availability of individual menus depends on the types of files you have open in LVS.

When no files are open, the LVS menu bar will look like this:

Commands for creating, opening, saving, and printing files

Commands for displaying or hiding elements of the user interface

Commands for starting, stopping, or setting up verification runs


Commands for accessing the online documentation



File View Verification Help

When the active window contains a setup file, the LVS menu bar will look like this:

Commands for selecting, tiling, or cascading windows



File View Verification Window Help

When the active window contains a text file, the LVS menu bar will look like this:

Commands for editing and searching in text files



The image shows a horizontal menu bar with five items: File, Edit, View, Window, and Help. Each item is underlined. The menu bar is set against a light gray background.

Toolbar

You can position the toolbar anywhere within the application window or dock it against one side of the application window. You can also display or hide the toolbar by selecting **View > Toolbar**.

The availability of toolbar buttons will depend on the type of file in the active window—setup or text—and whether or not a verification queue exists.

When a setup window or verification queue is active, the toolbar will look like this:

Create a new document

Open a file

Save a file



Run verification

Stop verification

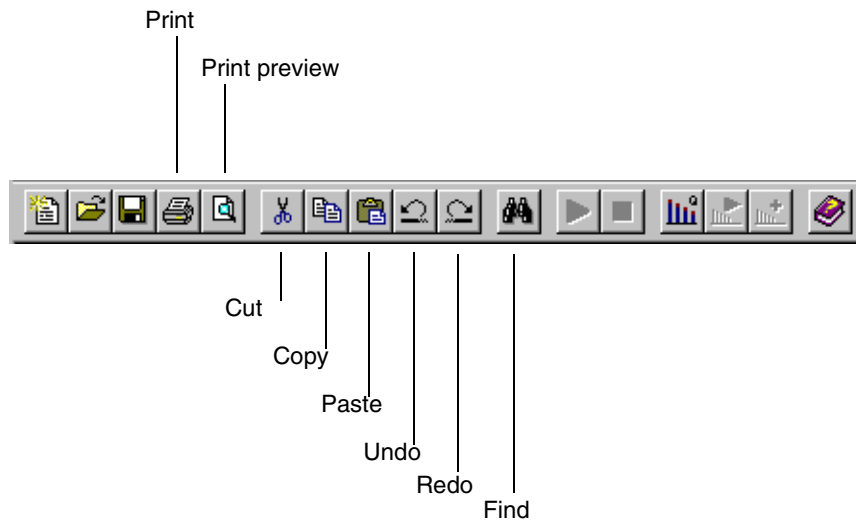
Open queue dialog

Run queue

Add setup to queue

Access online documentation

When a text or verification window is active, the toolbar will look like this:



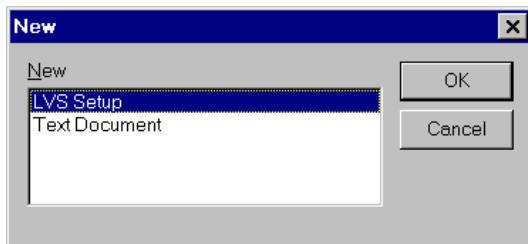
Status Bar

When a verification window is active and the cursor is in the output file section, the status bar displays the number of the line where the cursor is positioned. If a text window is active, the status bar displays both the line number and column number of the cursor position.

Setup Window

When one or more LVS setup files are open, the user interface will contain one setup window for every open setup file. The setup window contains five tabs, and it is used to specify input files and a variety of verification options.

To create a new setup window, select **File > New**. In the **New** dialog, select **LVS Setup** and click **OK**.



LVS will create a new setup window with the default name **Setup1**, which corresponds to the setup file **setup1.vdb**. When you save the setup file for the first time, you can either use the default name or supply your own, but you must use the **.vdb** extension (to ensure that Windows automatically recognizes the files as an LVS verification database). Subsequent new setup windows will receive similar names by default—**Setup2**, **Setup3**, etc.

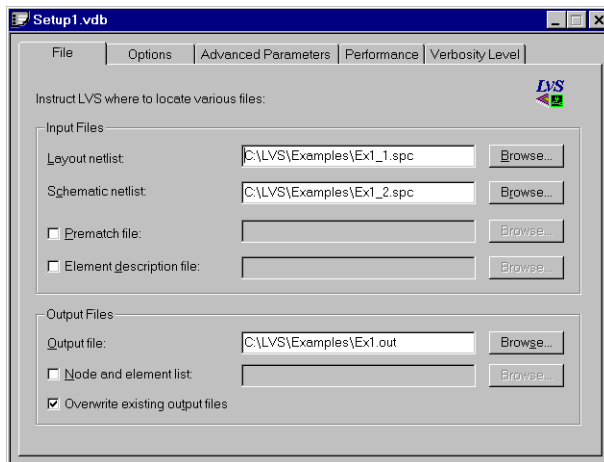
To open an existing setup window, select **File > Open**. In the **Open** dialog, select the appropriate **Verification Database File (*.vdb)** from the **Files of type** drop-down list and select an available file or enter its name in the **File name** field.

The following sections describe individual setup window tabs in detail.

- [Setup Window—File on page 3-158](#)
- [Setup Window—Options on page 3-161](#)
- [Setup Window—Advanced Parameters on page 3-164](#)
- [Setup Window—Performance on page 3-167](#)
- [Setup Window—Verbosity Level on page 3-169](#)

Setup Window—File

The **File** tab contains fields for specifying input and output files. You can type the correct filename in a field, or use a **Browse** button to navigate to an appropriate directory.



Options include:

Layout netlist

One of the input files to be compared. This file must be in SPICE format (T-Spice/H-Spice or P-Spice syntax). The default filename extension is **.sp**. (See [SPICE File Format on page 3-270](#).)

Schematic netlist

One of the input files to be compared. This file must be in SPICE format (T-Spice/H-Spice or P-Spice syntax). The default filename extension is **.sp**. (See [SPICE File Format on page 3-270](#).)

Prematch file

An optional input file that specifies equivalent elements and nodes for the iterative matching process. The default filename extension is **.pre**. (See [Prematch File Format on page 3-266](#).)

Element description file

An optional input file containing instructions on how to deal with custom devices. The default filename extension is **.elm**. (See [Element Description File Format on page 3-261](#).)

Output file

The output file containing verification results. This file must be identified in order to run the verification process. The default filename extension is **.out**.

Node and element list

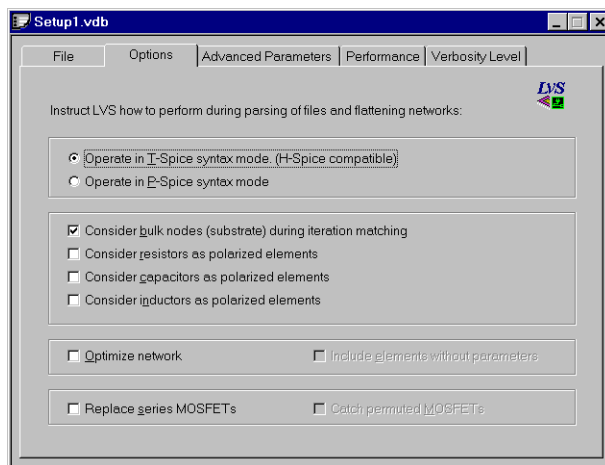
An optional output file containing a node and element list. The default filename extension is **.lst**.

Overwrite existing output files

Causes automatic overwriting of the named output and node and element list files even if these files already exist.

Setup Window—Options

The **Options** tab allows you to set options for parsing files and flattening networks:



Options include:

Operate in T-Spice syntax mode

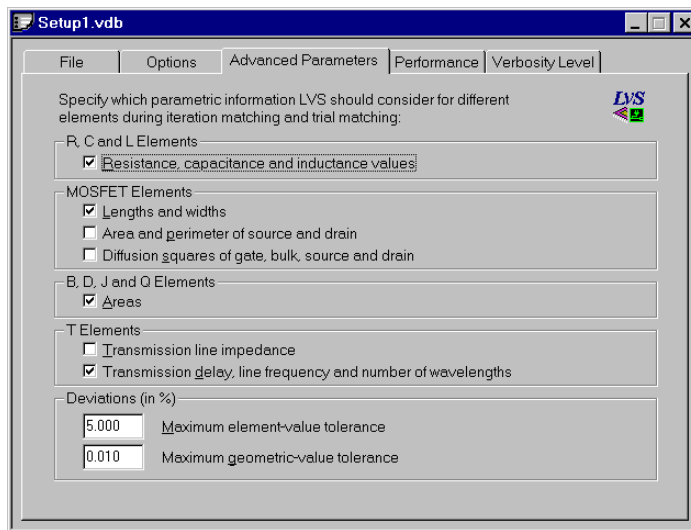
Assumes T-Spice syntax (which is H-Spice compatible). This is the default mode.

Operate in P-Spice syntax mode	Assumes P-Spice syntax. This mode must be selected if the input files were produced by P-Spice.
Consider bulk nodes (substrate) during iteration matching	Whether to consider or ignore bulk nodes on devices during verification. The default is to consider.
Consider resistors as polarized elements	Whether to regard resistors as polarized elements. The default is polarized.
Consider capacitors as polarized elements	Whether to regard capacitors as polarized elements. The default is polarized.
Consider inductors as polarized elements	Whether to regard inductors as polarized elements. The default is polarized.
Optimize network	Optimizes the two netlists for parallel and series resistors and capacitors and parallel MOSFETs. The default is not to optimize.
Include elements without parameters	If Optimize network is checked, device parameters can optionally be considered. The default is to consider.
Replace series MOSFETs	Replaces series chain MOSFETs with equivalent components, which reduces the processing required. The default is not to replace.

Catch permuted MOSFETs If **Replace series MOSFETs** is checked, permuted MOSFETs can be detected and identified. The default is not to identify.

Setup Window—Advanced Parameters

The **Advanced Parameters** tab allows you to specify which parametric information LVS should consider for different elements during iteration matching and detailed trial matching.



Options include:

R, C, and L Elements

Takes into account parametric information on resistance, capacitance, and inductance elements.

MOSFET Elements

Takes into account parametric information on MOSFET lengths and widths, source/drain areas and perimeters, and gate/bulk/source/drain diffusion squares.

B, D, J, and Q Elements

Takes into account parametric information on non-MOSFET semiconductor device areas.

T Elements

Takes into account parametric information on transmission line impedance, delay, frequency, and wavelength number.

Deviations (in %)

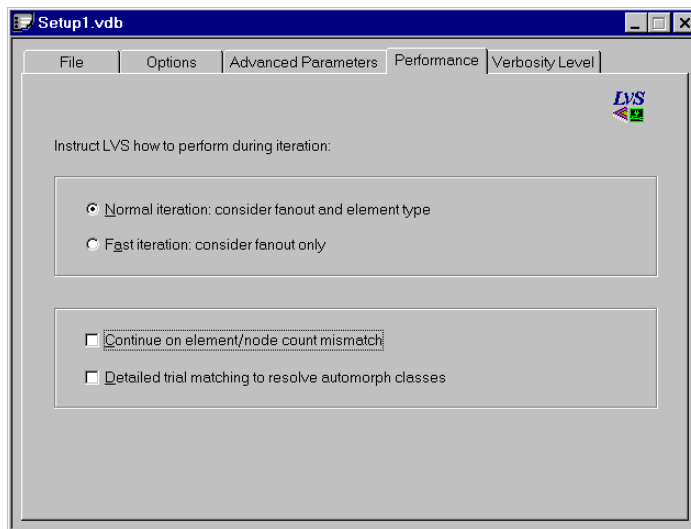
The **Maximum element-value tolerance** is the maximum amount (as a percentage of the larger value) by which two parameter values may differ and still compare as equal. The default is 5%. (This tolerance only applies to resistances, capacitances, inductances, lengths, widths, transmission delays, line frequencies, and wavelength numbers.)

The **Maximum geometric-value tolerance** is the maximum amount (as a percentage of the larger value) by which two geometric shapes may differ and still compare as equal. The default is 0.01%. (This tolerance only applies to source/drain areas and perimeters, gate/source/bulk/drain diffusion squares, B/D/J/Q element areas, and transmission line impedances.)

For further information, see [Parameter Matching on page 3-205](#).

Setup Window—Performance

The **Performance** tab contains options that instruct LVS how to perform during iteration.



Options include:

Normal iteration

Considers both fanout and element type. This is the default.

Fast iteration

Considers only fanout.

Continue on element/node count mismatch

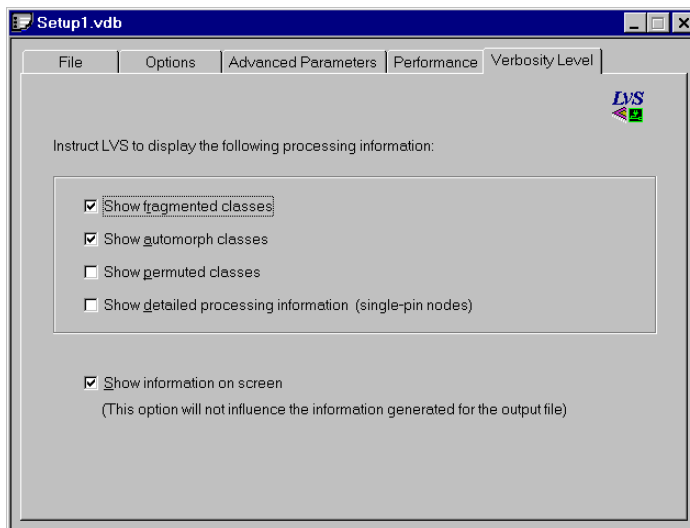
Continues to run even on element or node count mismatch. If this option is not selected, LVS issues a prompt when an element/node count mismatch occurs. In batch or queuing mode, this option is automatically selected.

Detailed trial matching to resolve automorph classes

Instructs LVS to perform detailed trial-matching when automorphed element or node classes occur without first issuing a prompt. In batch or queuing mode, this option is automatically selected.

Setup Window—Verbosity Level

The **Verbosity Level** tab contains options that specify which processing information LVS will display during the verification run.



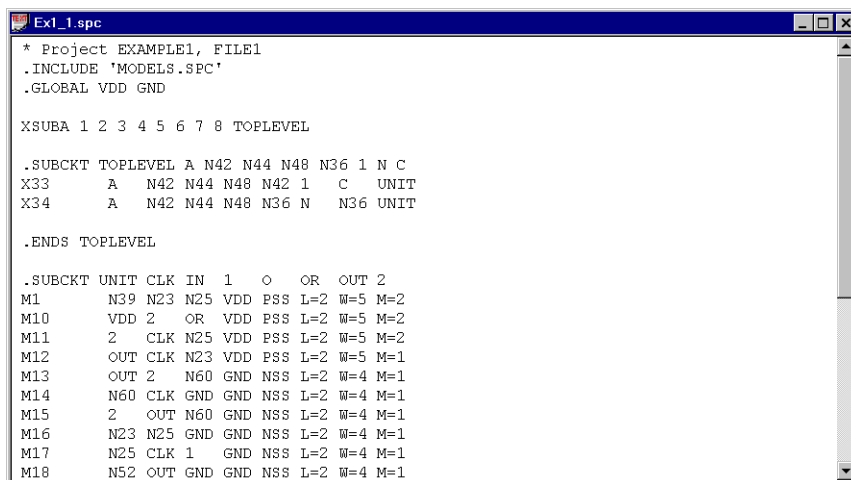
Options include:

Show fragmented classes	Shows information on fragmented classes.
Show automorph classes	Shows information on automorph classes.
Show permuted classes	Shows information on permuted classes.
Show detailed processing information	Displays single connection nodes. If a prematch file is used, prematched elements are displayed, as well as elements that LVS attempts to postmatch. If an element description file is included, non-flattened special elements are identified.
Show information on screen	Displays results in the verification window and writes them to the output file. If this option is not checked, results are written to the output file only.

For further information on individual netlist classes, see also [Resolving Fragmented Classes on page 3-199](#), [Resolving Automorph Classes on page 3-202](#) and [Permuted Classes in Digital Designs on page 3-207](#).

Text Window

When a SPICE file, or other text-format file, is open, the user interface will contain one or more text windows.



```

Ex1_1.spc
* Project EXAMPLE1, FILE1
.INCLUDE 'MODELS.SPC'
.GLOBAL VDD GND

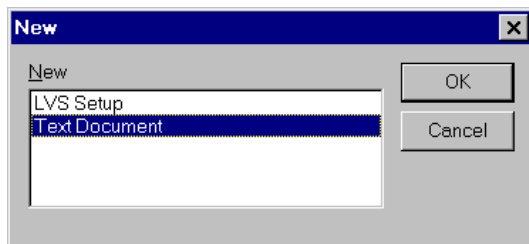
XSUBA 1 2 3 4 5 6 7 8 TOPLEVEL

.SUBCKT TOPLEVEL A N42 N44 N48 N36 1 N C
X33      A  N42 N44 N48 N42 1  C  UNIT
X34      A  N42 N44 N48 N36 N  N36 UNIT
.ENDS TOPLEVEL

.SUBCKT UNIT CLK IN 1 0 OR OUT 2
M1      N39 N23 N25 VDD PSS L=2 W=5 M=2
M10     VDD 2  OR  VDD PSS L=2 W=5 M=2
M11     2  CLK N25 VDD PSS L=2 W=5 M=2
M12     OUT CLK N23 VDD PSS L=2 W=5 M=1
M13     OUT 2  N60 GND NSS L=2 W=4 M=1
M14     N60 CLK GND GND NSS L=2 W=4 M=1
M15     2  OUT N60 GND NSS L=2 W=4 M=1
M16     N23 N25 GND GND NSS L=2 W=4 M=1
M17     N25 CLK 1  GND NSS L=2 W=4 M=1
M18     N52 OUT GND GND NSS L=2 W=4 M=1
  
```

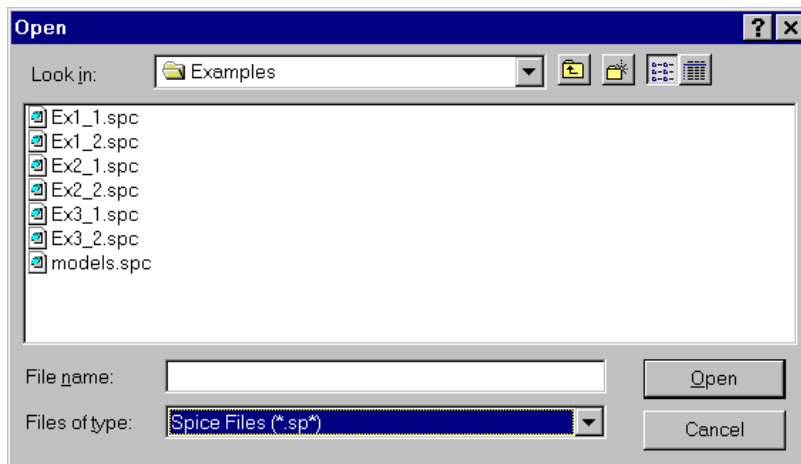
The text window is used to edit text-format files, such as SPICE, prematch, or element description files.

To create a new text file (and open a new text window), select **File > New**. In the **New** dialog, select **Text Document** and click **OK**.



LVS will create a new text window with the default name **LVS1**. When you save the text file, you can either use the default name or supply your own, but you must supply an appropriate filename extension, such as **.spc** or **.elm**. Subsequent new text windows will receive similar names by default—**LVS2**, **LVS3**, etc.

To open an existing text file, select **File > Open**. In the **Open** dialog, select the appropriate file type from the **Files of type** drop-down list and select an available file or enter its name in the **File name** field.



Within text files, LVS provides a full range of standard Windows editing capabilities. You can cut, copy, and paste text as you would in any standard text editor. You can undo successive text edits by selecting **Edit > Undo**, and you can reverse successive **Undo** commands by selecting **Edit > Redo**.

Using Find and Replace

You can search a text file for any string or replace all instances of a particular string with other input you supply. To initiate a search operation, select **Edit > Find**. LVS will display the **Find** dialog:



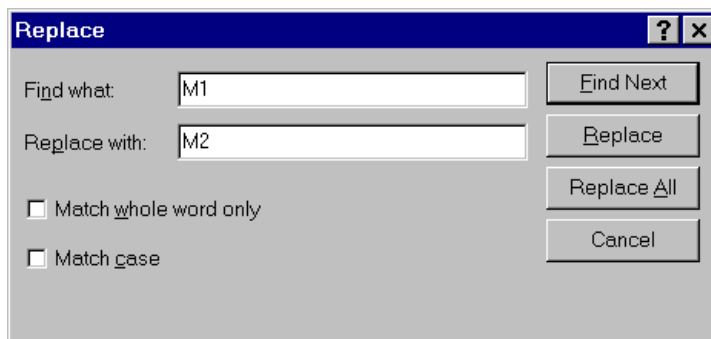
Enter a string in the **Find what** field and click **Find Next**. LVS will search the file in the active text window for the specified string and highlight it, if found. If you check the option **Match whole word only**, LVS will search only for whole words that match the specified search string—not just portions of a word. If you check the option **Match case**, LVS will perform a case-sensitive search.

Note:

LVS will search the entire text file from the current cursor location to the end of the file. When LVS reaches the end of the file, it will prompt you via a dialog box for permission to resume the search at the top of the file.

The **Find** dialog will remain open until you close it, but you will not be able to edit the text file. If you want to search for a string, edit it, then search for another instance of the same string, close this dialog and select **Edit > Find Next** (or press **F3**). LVS will continue searching the active text file for instances of the specified string until the last one has been found.

To perform a search-and-replace operation, select **Edit > Replace**. LVS will display the **Replace** dialog:



Enter a string in the **Find what** field and click **Find Next**. LVS will search the file in the active text window for the specified string and highlight it, if found. Next, enter a replacement string in the **Replace with** field. To replace a single instance of the search string with the replacement string, click **Replace**. To replace all instances of the search string with the replacement string, click **Replace All**.

The options **Match whole word only** and **Match case** have the same functionality as in the **Find** dialog. Note also that in the case of a **Replace** command, LVS will scan the text file as it would in the case of a **Find** command—from the current cursor location to the end of file, and then from the top of the file to the cursor location.

Using Go To

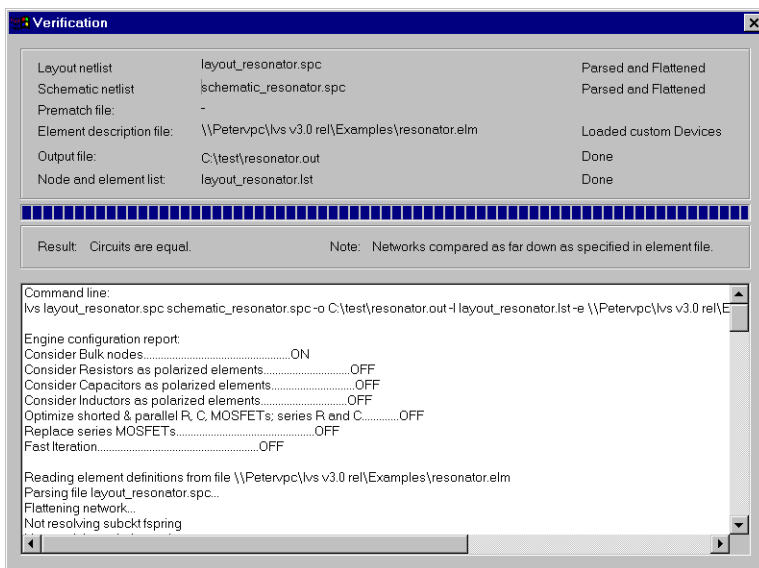
To move the cursor directly to a specific line in a text file, select **Edit > Go To**. LVS will display the following dialog:



Enter a line number and click **OK**. LVS will move the cursor to the beginning of the specified line.

Verification Window

LVS reports its progress and the results of a verification run in the verification window.



The verification results will appear in this window whenever you run a verification. You can also display the results of the last-performed verification by

selecting **View > Verification Window** anytime after the initial verification run. If you select this command before running a verification, the window will be empty.

The verification window is divided into three sections, or panes. The uppermost section lists the input and output files used in the verification, and the current status of each file. A graphic indicator also displays the progress LVS has made in processing the verification run as a whole.

The middle section of the verification window reports the result of a verification run, relevant notes, and any errors encountered during the verification.

The bottom section of the verification window contains an editable copy of the output file, which reports input files and options set in the setup window, plus verification progress statements written during the run. This section can be used to perform line edits on the output file using the following basic text editing commands. Note that in the verification window these commands are only available via the keyboard.

Ctrl+A	Select all.
Ctrl+C	Copy to clipboard.
Ctrl+F	Find.
Ctrl+G	Go to line.

Ctrl+S	Save to file.
Ctrl+V	Paste from clipboard.
Ctrl+X	Cut to clipboard.
Ctrl+Home	Position cursor on first line.
Ctrl+End	Position cursor on last line.
Page Up	Scroll up through file.
Page Down	Scroll down through file.
Delete	Delete line.

In the verification window ([page 2-25](#)), the bottom section shows the command line derived from entries made in the setup window. This command line is used to initiate the verification run:

Command line:

```
lvs layout_resonator.spc schematic_resonator.spc -o C:\test\resonator.out -l
layout_resonator.lst -e \\Petersvpc\lvs v3.0 rel\Examples\resonator.elm -nrcl -y2 -
vfpar
```

Beneath the command line, LVS displays the options chosen in the setup window:

Engine configuration report:

```
Consider Bulk nodes.....ON
```

```

Consider Resistors as polarized elements.....OFF
Consider Capacitors as polarized elements.....OFF
Consider Inductors as polarized elements.....OFF
Optimize shorted & parallel R, C, MOSFETs; series R and C.....OFF
Replace series MOSFETs.....OFF
Fast Iteration.....OFF

```

Next, the verification window displays the progress statements written during the verification run:

```

Reading element definitions from file \\Petersvpc\lvs v3.0 rel\Examples\resonator.el
Parsing file layout_resonator.spc...
Not resolving subckt fspring
Not resolving subckt comb
Not resolving subckt fspring
Not resolving subckt comb
Not resolving subckt plate4
Parsing file schematic_resonator.spc...

```

For further information on LVS verification output, see the [LVS Output Tutorial on page 3-214](#).

You can rerun the verification whose results are reported in the verification window by clicking on the appropriate setup window and pressing **F5**.

Verification Queue

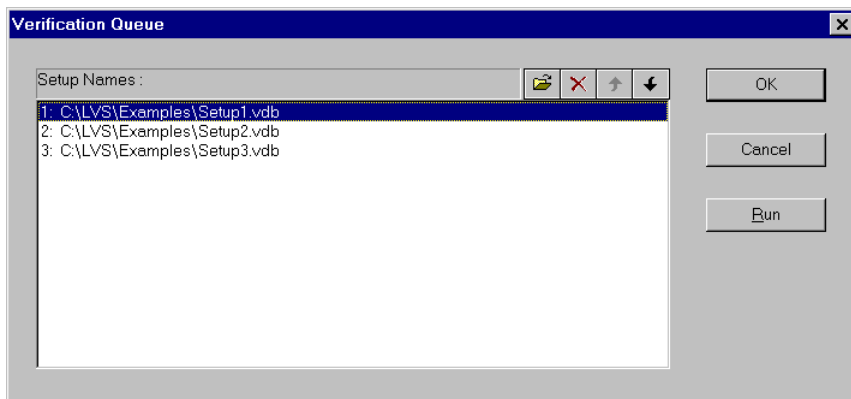
You can run several verifications consecutively by creating a verification queue. Once you have created such a queue, you can use it to run multiple verifications without additional user intervention.

To create a verification queue or add a setup to an existing queue, select **Verification > Add to Queue** or click the following toolbar button:



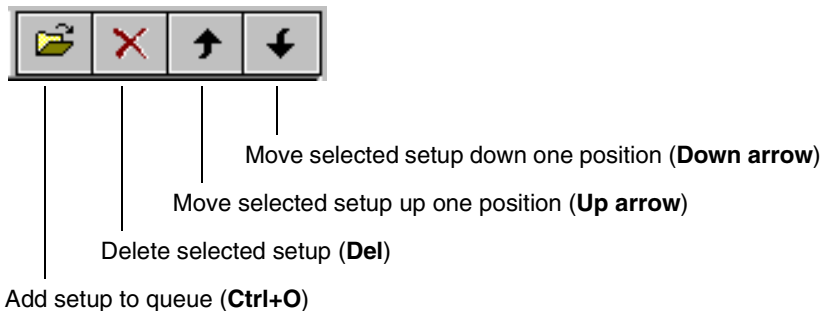
LVS will automatically add the active setup to the end of the existing queue.

To display a verification queue, select **Verification > Verification Queue (F8)** or click the **Open Queue Dialog** button. LVS will display a dialog like this:



Each setup is listed in the **Setup Names** field. You can make changes to the existing queue via keyboard shortcuts or a toolbar within the dialog. The toolbar

buttons, with their functions and corresponding keyboard shortcuts, are illustrated below:

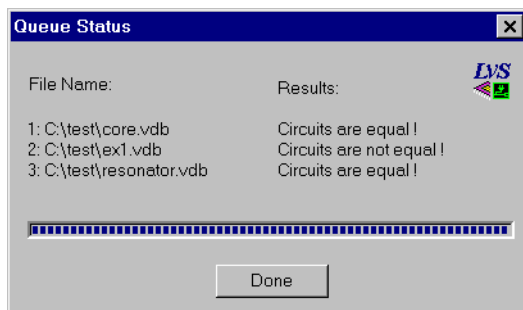


Clicking **OK** saves any changes you have made to the verification queue and closes the dialog.

Clicking **Run** initiates a verification for every setup listed in the verification queue. You can also initiate such a verification run with the verification queue closed by selecting **Verification > Start Verification Queue (F9)** or by clicking the **Run Queue** toolbar button:



When LVS has finished running the verification queue, it will display a dialog like the following:



Using LVS in Batch Mode

Using LVS in *batch mode* provides another way to run multiple verifications consecutively. To run LVS in batch mode, you must first create a *batch file*. A batch file is a text file containing one or more command-line invocations of LVS and appropriate setup information—input files and verification options—for each verification run.

Creating a Batch File

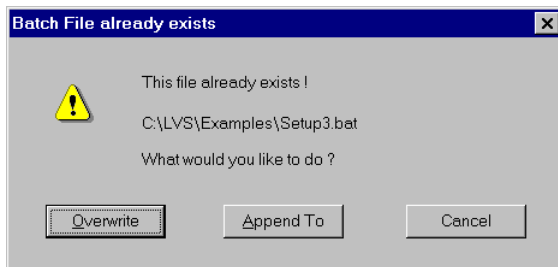
To create a batch file, select **File > Export Batch File (Ctrl+E)** with a setup window active. LVS will display the **Export Batch File as** dialog. Navigate to the correct directory if necessary and enter the name of the batch file in the **File name** field, then click **OK**. LVS will create a batch file with a single invocation of LVS and command-line options derived from parameters set in the active setup window.

A simple batch file will contain text like the following:

```
start /w lvs "C:\LVS\Examples\Ex2_1.spc"
"C:\LVS\Examples\Ex2_2.spc" -o "C:\LVS\Examples\Ex2.out"
-nrcl -c1245 -dv5.000 -dg0.010 -vfa -q
```

For further information on batch-file syntax, see [Batch-File Syntax](#) on page 3-243. For further information on command-line options, see [Options](#) on page 3-245.

To add verifications to an existing batch file, select it in the **Export Batch File as** dialog and click **OK**. LVS will prompt you with the following dialog:



To add to the current batch file, click **Append To**. LVS will append an appropriate command line to the existing batch file. If you click **Overwrite**, LVS will overwrite the existing batch file with a new command line using setup information derived from the current setup window.

You can further develop a batch file by editing it in the LVS text window. For example, you can change the names and paths of the SPICE files used in the verification, or set different command-line options.

To run an LVS batch file, double-click it in Windows Explorer or invoke it from a DOS command prompt. Note that LVS does not support launching of multiple instances—to run a batch file, you must first exit LVS for Windows.

For further information, see [Running a Batch File on page 3-244](#).

Tutorial

This brief tutorial will teach you how to perform the following basic tasks in LVS:

- Set up and run a verification
- Create a verification queue
- Create and run a verification batch file

For information on interpreting LVS output and resolving verification problems, see the [LVS Output Tutorial on page 3-214](#).

Creating a Verification Setup

In the following steps, you will learn to create and run an LVS verification database (**.vdb**) file. The file you will create is also provided as **C:\install_dir\Examples\ex1.vdb**.

- ☒ Start LVS, if it is not already running.
- ☒ Select **File > New** or click the **New** button in the toolbar.
- ☒ In the **New** dialog, select **LVS Setup** as the file type, and click **OK**.

- ☑ LVS will display a setup window. In the **File** tab, type the names of the following input and output files. You can also use the **Browse** buttons to navigate to these files.
 - The two netlists you wish to compare. For this tutorial, use the files **Ex1_1.spc** as the layout netlist and **Ex1_2.spc** as the schematic netlist. Both of these files are located in **C:\install_dir\Examples**.
 - In the **Output file** field, the name of the output file. For this tutorial, use **C:\install_dir\Examples\ex1.out**.
- ☑ Check the option **Overwrite existing output files** so that LVS will overwrite this output file on subsequent verification runs without prompting you.
- ☑ On the **Options** tab, check the following options:
 - **Consider bulk nodes (substrate) during iteration matching**
 - **Consider resistors as polarized elements**
 - **Consider capacitors as polarized elements**
 - **Consider inductors as polarized elements**
- ☑ On the **Advanced Parameters** tab, check **Resistance, capacitance, and inductance values**. Enter a **Maximum element-value tolerance** of **5.000**.
- ☑ On the **Performance** tab, select **Normal iteration: consider fanout and element type**.

- ☒ On the **Verbosity Level** tab, check the following options:
 - **Show fragmented classes**
 - **Show automorph classes**
 - **Show permuted classes**
 - **Show information on screen**
- ☒ Save your setup as **ex1.vdb**, or another name of your choice.
- ☒ Select **Verification > Run (F5)** or click the **Run Verification** button on the toolbar.
- ☒ This verification run will produce an element/node count mismatch. LVS will prompt you for permission to continue iteration. Click **Yes**.

LVS will complete the verification run and display the following results in the verification window:

```
0 perfectly matched element class(es)
1 fragmented element class(es)
3 perfectly matched node class(es)
12 fragmented node class(es)
```

This verification run results in one fragmented element class and 12 fragmented node classes. It is not necessary to resolve these fragmented classes now, but the correct procedure is described in [Resolving a Fragmented Class on page 3-231](#).

- ☑ Now create another verification setup using the following input files and parameters. Save your setup file as **ex2.vdb** (or use the setup file supplied with LVS).
- On the **File** tab, select **C:\install_dir\Examples\Ex1_2.spc** as the layout netlist file and **C:\install_dir\Examples\Ex2_1.spc** as the schematic netlist file. Name the output file **C:\install_dir\Examples\ex2.out**.
- On the **Options** tab, check the following options:
 - Consider bulk nodes (substrate) during iteration matching**
 - Consider resistors as polarized elements**
 - Consider capacitors as polarized elements**
 - Consider inductors as polarized elements**
- On the **Advanced Parameters** tab, check the following options:
 - Resistance, capacitance, and inductance values.** Then enter a **Maximum element-value tolerance** of **5.000**.
 - Areas (in B, D, J and Q Elements).** Then enter a **Maximum geometric-value tolerance** of **0.010**.
- On the **Performance** tab, select **Normal iteration: consider fanout and element type**.
- On the **Verbosity Level** tab, check the following options:

Show fragmented classes

Show automorph classes

Show permuted classes

Show information on screen

- ☑ Select **Verification > Run (F5)** or click the **Run Verification** button on the toolbar. LVS will complete the verification run and display the following results in the verification window:

```
32 perfectly matched element class(es)
18 perfectly matched node class(es)
4 fragmented node class(es)
```

- ☑ When you are finished viewing the results, close the verification window.

Creating a Verification Queue

In the following steps, you will learn how to create and run a verification queue.

- ☑ Select **ex2.vdb** as the active setup window.
- ☑ Select **Verification > Add to Queue (F7)** or click the **Add to queue button** on the toolbar.

- ☑ Select **Verification > Verification Queue (F8)**. LVS will display the **Verification Queue** dialog with the file **ex2.vdb** listed in the queue.
 - ☑ Within the verification queue dialog, click the **Add setup to queue** button. (**Ctrl+O**). LVS will open a standard Windows file browser.
 - ☑ Select the file **ex1.vdb** and click **Open**. LVS will add this setup file to the verification queue.
 - ☑ Repeat the last two steps and add file **ex3_1.vdb** to the queue.
 - ☑ If desired, rearrange the setups in the queue, using the arrow keys or appropriate buttons in the **Verification Queue** dialog.
 - ☑ To run the verification, click **Run** in the **Verification Queue** dialog, or click **OK** to close the dialog, then select **Verification > Start Verification Queue (F9)**. You can also click the **Run Queue** button in the application toolbar.
- LVS will run the verifications in the order listed and report the results in the **Queue Status** dialog.
- ☑ Close the **Verification Queue** dialog, if necessary.

In the next section, you will learn how to create and run a verification batch file.

Running LVS in Batch Mode

- ☑ Select **ex1.vdb** as the active setup window.
- ☑ Select **File > Export Batch File (Ctrl+E)**.
- ☑ LVS will display the **Export Batch File as** dialog. Select **ex1.bat** or type it in the **File name** field. Click **Save**.
- ☑ Select **ex2.vdb** as the active setup window.
- ☑ Select **File > Export Batch File (Ctrl+E)**. In the **Export Batch File as** dialog, type **ex1.bat** in the **File name** field and click **OK**.
- ☑ LVS will prompt you with the dialog **Batch file already exists**. Click the **Append To** button.
- ☑ Select **ex3_1.vdb** as the active setup window and repeat the last two steps to add this setup to the batch file.
- ☑ Exit LVS and open a DOS window. At the command prompt, type **ex1.bat**.

LVS will run the verifications listed in the batch file and write the results to the specified output file(s). You can view the results using the LVS text window or any other text editor.