cādence®

Connectivity-to-Schematic User Guide

Product Version ICADVM20.1
October 2020

Contents

Preface

The Connectivity-to-Schematic tool is used to generate digital and analog schematic views from netlist views. This user guide describes how to use the Connectivity-to-Schematic tool. It is aimed at the designers of digital and analog circuits and assumes that you are familiar with:

■ The Virtuoso design environment and application infrastructure mechanisms designed to support consistent operations between all Cadence tools.

■ The applications used to design and develop integrated circuits in the Virtuoso design environment, notably Virtuoso Schematic Editor.

■ Virtuoso technology data.

This manual is aimed at designers of integrated circuits and assumes that you are familiar with:

■ The Virtuoso design environment and application infrastructure mechanisms designed to support consistent operations between all Cadence tools.

■ The applications used to design and develop integrated circuits in the Virtuoso design environment, notably Virtuoso Layout Suite and Virtuoso Schematic Editor.

■ Component Description Format (CDF), which lets you create and describe your own components for use with ADE.

This preface contains the following topics:

■ Scope

■ Licensing Requirements

■ Related Documentation

■ Additional Learning Resources

■ Customer Support

■ Feedback about Documentation

■ Typographic and Syntax Conventions

Scope

Unless otherwise noted, the functionality described in this guide can be used in both mature node (for example, IC6.1.8) and advanced node (for example, ICADVM18.1) releases.

Label	Meaning
(ICADVM18.1 Only)	Features supported only in the ICADVM18.1 advanced nodes and advanced methodologies release.
(IC6.1.8 Only)	Features supported only in mature node releases.

Licensing Requirements

Digital Schematic View

In this view, the Connectivity-to-Schematic tool searches for the following licenses in the specified order and checks out one of them:

■ 95100 Virtuoso® Schematic Editor L

■ 95300 Virtuoso® Layout Suite L

■ 95115 Virtuoso® Schematic Editor XL

■ 95310 Virtuoso® Layout Suite XL

■ 4 tokens of 95320 Virtuoso® Layout Suite - GXL

Analog Schematic View

In this view, the Connectivity-to-Schematic tool searches for the following licenses in the specified order and checks out one of them:

■ 95115 Virtuoso® Schematic Editor XL

■ 95310 Virtuoso® Layout Suite XL

■ 4 tokens of 95320 Virtuoso® Layout Suite - GXL

For information on licensing in the Virtuoso design environment, see *Virtuoso Software Licensing and Configuration Guide*.

Related Documentation

What's New and KPNS

■ *Connectivity-to-Schematic What's New*

■ *Connectivity-to-Schematic Known Problems and Solutions*

Installation, Environment, and Infrastructure

■ *Cadence Installation Guide*.

■ *Virtuoso Design Environment User Guide*.

■ *Cadence Application Infrastructure User Guide*.

SKILL and Technology Information

■ *HDL Import and Netlist-to-Schematic Conversion SKILL Reference*

■ *Virtuoso Technology Data User Guide* and the *Virtuoso Technology Data ASCII Files Reference*

■ *Virtuoso Technology Data SKILL Reference*

Virtuoso Tools

IC6.1.8 Only

■ *Virtuoso Layout Suite L User Guide*

■ *Virtuoso Layout Suite XL User Guide*

■ *Virtuoso Layout Suite GXL Reference*

ICADVM18.1 Only

■ *Virtuoso Layout Viewer User Guide*

- *Virtuoso Layout Suite XL: Basic Editing User Guide*

- *Virtuoso Layout Suite XL: Connectivity Driven Editing Guide*

- *Virtuoso Layout Suite EXL Reference*

- *Virtuoso Concurrent Layout User Guide*

- *Virtuoso Design Planner User Guide*

- *Virtuoso Multi-Patterning Technology User Guide*

- *Virtuoso Placer User Guide*

- *Virtuoso Simulation Driven Interactive Routing User Guide*

- *Virtuoso Width Spacing Patterns User Guide*

- *Virtuoso RF Solution Guide*

- *Virtuoso Electromagnetic Solver Assistant User Guide*

IC6.1.8 and ICADVM18.1

- *Virtuoso Abstract Generator User Guide*

- *Virtuoso Custom Digital Placer User Guide*

- *Virtuoso Design Rule Driven Editing User Guide*

- *Virtuoso Electrically Aware Design Flow Guide*

- *Virtuoso Floorplanner User Guide*

- *Virtuoso Fluid Guard Ring User Guide*

- *Virtuoso Interactive and Assisted Routing User Guide*

- *Virtuoso Layout Suite SKILL Reference*

- *Virtuoso Module Generator User Guide*

- *Virtuoso Parameterized Cell Reference*

- *Virtuoso Pegasus Interactive User Guide*

- *Virtuoso Space-based Router User Guide*

- *Design Data Translator's Reference*.

- *Verilog In for Virtuoso Design Environment User Guide and Reference*

■ *VHDL In for Virtuoso Design Environment User Guide and Reference*

Additional Learning Resources

Video Library

The Video Library on the Cadence Online Support website provides a comprehensive list of videos on various Cadence products.

To view a list of videos related to a specific product, you can use the *Filter Results* feature available in the pane on the left. For example, click the *Virtuoso Layout Suite* product link to view a list of videos available for the product.

You can also save your product preferences in the Product Selection form, which opens when you click the *Edit* icon located next to *My Products*.

Virtuoso Videos Book

You can access certain videos directly from Cadence Help. To learn more about this feature and to access the list of available videos, see Virtuoso Videos.

Rapid Adoption Kits

Cadence provides Rapid Adoption Kits that demonstrate how to use Virtuoso applications in your design flows. These kits contain design databases and instructions on how to run the design flow.

In addition, Cadence offers the following training course on schematic views:

■ Virtuoso Schematic Editor

Cadence also offers the following training courses on the SKILL programming language, which you can use to customize, extend, and automate your design environment:

■ SKILL Language Programming Introduction

■ SKILL Language Programming

The courses listed above are available in North America. For specific information about the courses available in your region, visit Cadence Training or write to training_enroll@cadence.com.

Note: The links in this section open in a separate web browser window when clicked in Cadence Help.

Help and Support Facilities

Virtuoso offers several built-in features to let you access help and support directly from the software.

■ The Virtuoso *Help* menu provides consistent help system access across Virtuoso tools and applications. The standard Virtuoso *Help* menu lets you access the most useful help and support resources from the Cadence support and corporate websites directly from the CIW or any Virtuoso application.

■ The Virtuoso Welcome Page is a self-help launch pad offering access to a host of useful knowledge resources, including quick links to content available within the Virtuoso installation as well as to other popular online content.

   The Welcome Page is displayed by default when you open Cadence Help in standalone mode from a Virtuoso installation. You can also access it at any time by selecting *Help – Virtuoso Documentation Library* from any application window, or by clicking the *Home* button on the Cadence Help toolbar (provided you have not set a custom home page).

For more information, see <u>Getting Help</u> in *Virtuoso Design Environment User Guide.*

Customer Support

For assistance with Cadence products:

■ Contact Cadence Customer Support

   Cadence is committed to keeping your design teams productive by providing answers to technical questions and to any queries about the latest software updates and training needs. For more information, visit <u>https://www.cadence.com/support</u>.

■ Log on to Cadence Online Support

   Customers with a maintenance contract with Cadence can obtain the latest information about various tools at <u>https://support.cadence.com</u>.

Feedback about Documentation

You can contact Cadence Customer Support to open a service request if you:

■   Find erroneous information in a product manual

■   Cannot find in a product manual the information you are looking for

■   Face an issue while accessing documentation by using Cadence Help

You can also submit feedback by using the following methods:

■   In the Cadence Help window, click the *Feedback* button and follow instructions.

■   On the Cadence Online Support Product Manuals page, select the required product and submit your feedback by using the *Provide Feedback* box.

Typographic and Syntax Conventions

The following typographic and syntax conventions are used in this manual.

text	Indicates names of manuals, menu commands, buttons, and fields.
`text`	Indicates text that you must type exactly as presented. Typically used to denote command, function, routine, or argument names that must be typed literally.
`z_argument`	Indicates text that you must replace with an appropriate argument value. The prefix (in this example, `z_`) indicates the data type the argument can accept and must not be typed.
`|`	Separates a choice of options.
`{ }`	Encloses a list of choices, separated by vertical bars, from which you **must** choose one.
`[ ]`	Encloses an optional argument or a list of choices separated by vertical bars, from which you **may** choose one.
`[ ?argName t_arg ]`	
	Denotes a *key argument*. The question mark and argument name must be typed as they appear in the syntax and must be followed by the required value for that argument.
`...`	Indicates that you can repeat the previous argument.
	Used with brackets to indicate that you can specify zero or more arguments.
	Used without brackets to indicate that you must specify at least one argument.
`,...`	Indicates that multiple arguments must be separated by commas.
`=>`	Indicates the values returned by a Cadence® SKILL® language function.
`/`	Separates the values that can be returned by a Cadence SKILL language function.

If a command-line or SKILL expression is too long to fit within the paragraph margins of this document, the remainder of the expression is moved to the next line and indented. In code excerpts, a backslash (\ ) indicates that the current line continues on to the next line.

1

Introducing Connectivity-to-Schematic

The Connectivity-to-Schematic tool (conn2sch) is used to generate schematic views from netlist views. A netlist view contains only connectivity information for a design. The Connectivity-to-Schematic tool generates a fully placed and routed schematic view using the connectivity information. You can create both, digital and analog schematic views using this tool. The Connectivity-to-Schematic tool provides different placement and routing engines to generate digital and analog schematic views. A schematic view can be opened, viewed, and edited using Virtuoso® Schematic Editor.

This chapter describes how the Connectivity-to-Schematic tool works. The chapter also describes different methods of running the tool.

Connectivity-to-Schematic — How it Works

The Connectivity-to-Schematic tool:

1. Accepts an OpenAccess (OA) netlist view as input and generates an OA schematic view by placing each component and wiring the connections with pins and nets.

    Note: This tool accepts only one cellview at a time. If you want to create a schematic view for more than one cellview or an entire library, you should call the `conn2Sch` function or conn2sch executable iteratively for each cellview.

2. Retains all cellview properties of source cellview in the destination schematic view.

3. Retains all effective CDF properties and other instance properties.

The following figure shows the files that the Connectivity-to-Schematic tool uses and creates while generating a schematic view:
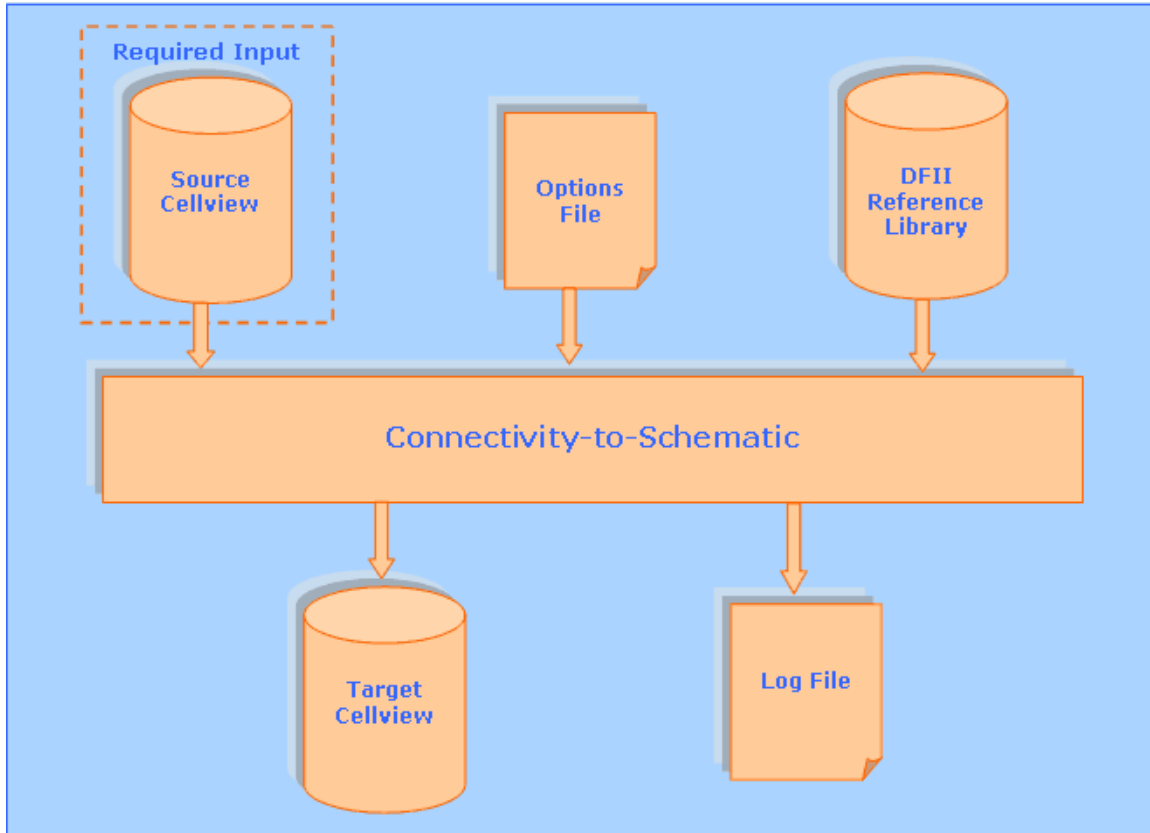
Note: You need to specify a valid OA cellview as a source to generate a schematic view.

Running the Connectivity-to-Schematic Tool

You can run the Connectivity-to-Schematic tool using:

■   Graphical User Interface

■   SKILL Functions

■   Standalone Mode

Graphical User Interface

The graphical user interface (GUI) of the Connectivity-to-Schematic tool can be accessed from Virtuoso® Design Environment workbench. To access the Connectivity-to-Schematic GUI:

1. Click *File* and point to *Import*. A list of import tools is displayed as shown in the following figure.
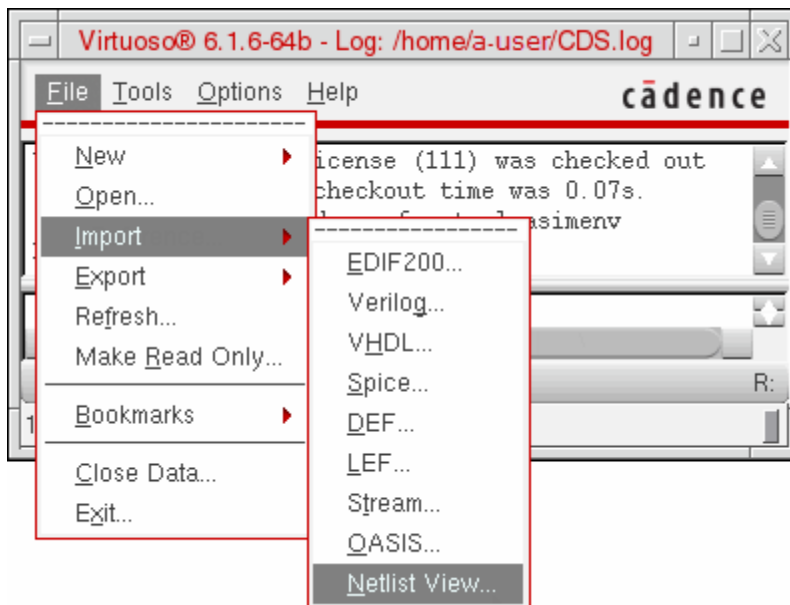
2. Click *Netlist View* to launch the Connectivity-to-Schematic tool. The following figure shows the GUI of the Connectivity-to-Schematic tool:
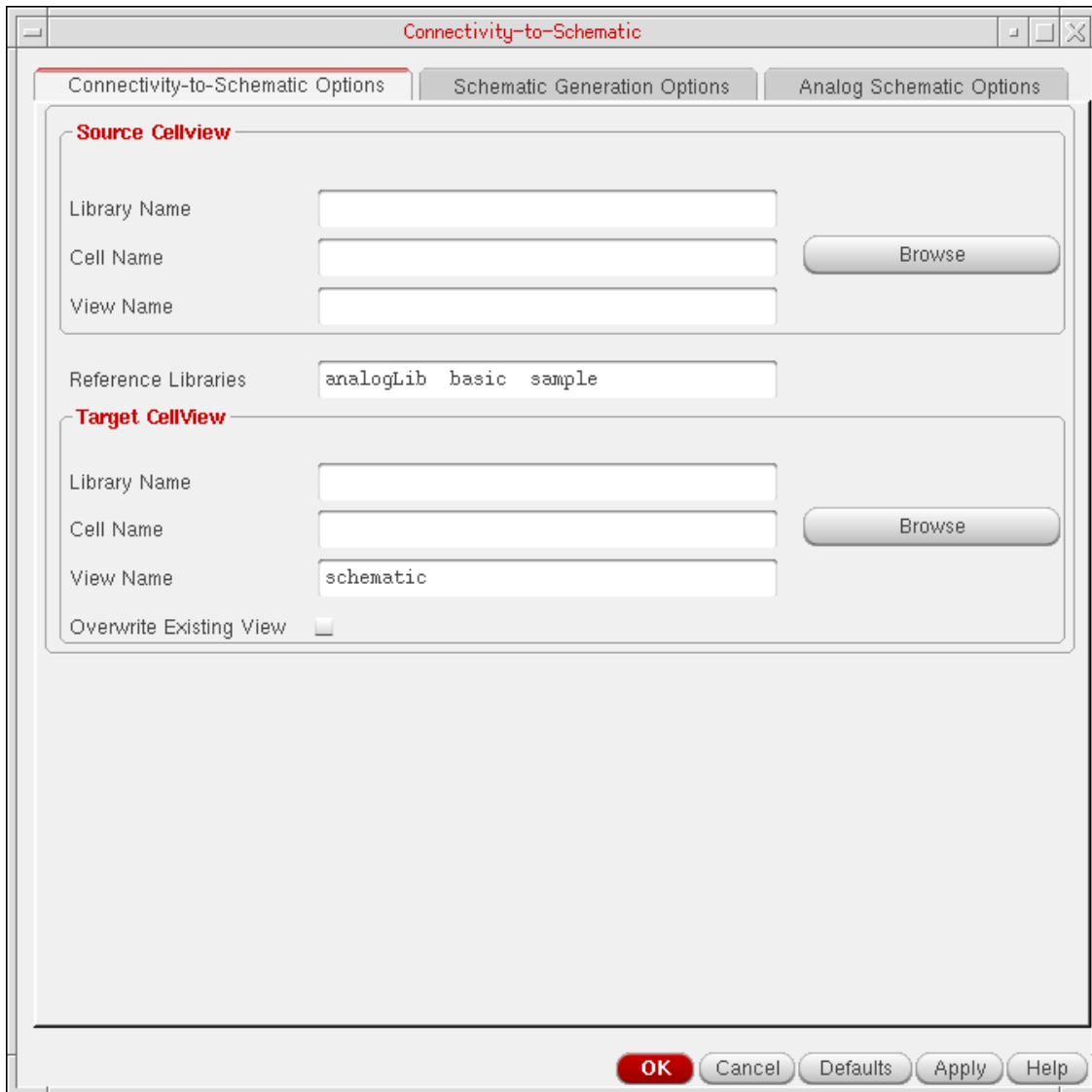
Alternatively, you can use the `Alt+f i n` menu access keys (press `Alt` and `f` keys simultaneously and then press the `i` key followed by the `n` key) to run the Connectivity-to-Schematic tool from CIW.

For information about using the Connectivity-to-Schematic GUI, see Chapter 2, "Working with Connectivity-to-Schematic".

SKILL Functions

The Connectivity-to-Schematic tool supports SKILL functions that enable you to create schematic views and launch GUI of the tool. These functions are run in Virtuoso® Design Environment workbench.

To know more about these SKILL functions, see *HDL Import and Netlist-to-Schematic Conversion SKILL Reference*.

Standalone Mode

You use the user interface of Connectivity-to-Schematic to specify the options and parameter settings. However, while working in the standalone mode, you can create files that contain all the options and parameter settings used by the tool to generate schematics. To know how to work in the standalone mode, see Chapter 3, "Working In Standalone Mode".

2

Working with Connectivity-to-Schematic

The Virtuoso Design Environment workbench enables you to access the Connectivity-to-Schematic tool using both, GUI and SKILL functions. This chapter explains the GUI of the Connectivity-to-Schematic tool. The chapter also explains how to generate digital and analog schematic views. In addition, the chapter describes environment variables and SKILL functions supported by the Connectivity-to-Schematic tool.

Introducing the User Interface

The user interface of Connectivity-to-Schematic consists of the following tabs:

■ Connectivity-to-Schematic Options

■ Schematic Generation Options

■ Analog Schematic Options

In addition, the user interface contains the following buttons, which are common for all tabs:

■ *OK*: Passes the information that you provide to generate a schematic to conn2sch executable and closes the user interface of Connectivity-to-Schematic.

■ *Cancel*: Closes the user interface of Connectivity-to-Schematic.

■ *Defaults*: Restores default settings on the tabs. If you change the default settings on any tab, the changes are retained during the workbench session, by default.

■ *Apply*: Passes the information that you provide to generate a schematic to conn2sch executable. Unlike the *OK* button, the user interface of Connectivity-to-Schematic does not close when you click the *Apply* button.

■ *Help*: Access help on how to use Connectivity-to-Schematic and get online support.

Connectivity-to-Schematic Options Tab

This tab contains options to specify source and target cellviews required to generate a schematic view. The following figure shows the *Connectivity-to-Schematic Options* tab:

Source Cellview

Contains fields to specify a source cellview, which you need to convert to a schematic view. A source cellview consists of an existing library, cell, and view. You can either type the library,

cell, and view names in the *Library Name*, *Cell Name*, and the *View Name* fields or select these using the *Browse* button.

Reference Libraries

Specifies a list of libraries that contain the symbol views for the instances present in the source cellview. The analogLib, basic, and sample libraries are displayed as reference libraries, by default. If a master instance of all other instances in the source cellview is not of the *dbcSchematicSymbol* type, then a symbol from the reference libraries is used. If a symbol is not found in any of the reference libraries, then a new symbol view is created in the destination library and this is used as an instance master.

If you want to generate a multi-sheet schematic, you can specify a reference library containing the sheet border and the index sheet symbols, such as the `US_8ths` library.

Target Cellview

Contains fields to specify the destination library name, cell name, and view name where you need to generate the schematic view. You can either type the library, cell, and view names in the *Library Name*, *Cell Name*, and the *View Name* fields or select these using the *Browse* button.

If you do not specify the target library name or the cell name, the source library name or the source cell name is used by default. The target view name is schematic, by default. However, you can change the view name to any other view name.

Browse

Opens Library Browser to enable you to select the target library, cell, and view names. If the target cell and view do not exist, you can use the *Browse* button to select only the library name. To specify new target cell and view names, you need to type them in the *Cell Name* and *View Name* fields.

The *Browser* button is available for the source and target cellviews.

Overwrite Existing View

Overwrites the target cellview if it exists. If you do not select this option and a target cellview of the same name exists, then the Connectivity-to-Schematic tool will exit without creating the schematic.

Default value: Off

Schematic Generation Options Tab

This tab contains options to manage the appearance of the schematic view, which Connectivity-to-Schematic generates. For example, you can specify if the schematic should be fully routed or extracted. The following figure shows the *Schematic Generation Options* tab:

⚠️ *Important*

> To create the symbol of the top cell of your input design, set the environment variable
> `generateTopCellSymbol` to `t`.

Sheet Symbol

Specifies the symbol that controls the size and page orientation of the sheet on which the schematic is to be made. Valid entries are the names of customer-designed sheet symbols and sheet symbols offered by Virtuoso® Design Environment.

A selection of sample sheet symbols is shipped with Virtuoso Design Environment in the `<install_dir>/tools/dfII/etc/cdslib/sheets/US_8ths` library. These symbols include metric sheet symbols `A0` through `A4`, and sheet symbols for the traditional `A`, `A.book` (vertical orientation), `B`, `C`, `D`, `E`, and `F` sizes.

The library containing the sheet symbol entered in this field must be in your `cds.lib` file for the Connectivity-to-Schematic tool to use the correct information.

If the *Sheet Symbol* is set to `none`, a single sheet schematic is generated. If a specific sheet symbol size is provided, and the schematic cannot fit on a single sheet, a multi-sheet schematic is generated. A multi-sheet schematic has one index sheet and many schematic sheets.

The cell name of the index sheet for the multi-sheet schematic has the same name as the target cell name. The schematic sheets are named as <target_cell_name>@sheet<nnn> where *nnn* is the sheet number. For example, if the target cell name is top, the index sheet is also named as top. If the top index sheet has three schematic sheets, these are named as top@sheet001, top@sheet002, and top@sheet003.

Default value: `none`

Maximum Number Of Rows

Specifies the maximum number of rows on each sheet. This option is used only on a multi-sheet schematic. The value must be an integer from `1` to `1024`. The maximum number of rows that can fit on a sheet is `1024`.

The number of rows that are displayed on a sheet might be less than the specified number of rows depending on the size of the instance symbols and the size of the sheet.

Default value: `1024`

Maximum Number Of Columns

Specifies the maximum number of columns on each sheet. This option is used only on a multi-sheet schematic. The value must be an integer from `1` to `1024`. The maximum number of columns that can fit on a sheet is `1024`.

The number of columns that are displayed on a sheet might be less than the specified number of columns depending on the size of the instance symbols and the size of the sheet.

Default value: `1024`

Font Height

Controls the size of the font used for pin, wire, and instance labels. The value must be a real value between `0.0375` and `0.125`. The wire and instance labels use the font size specified in *Font Height*. Pin labels are scaled down to 75 percent of the specified size.

Default value: `0.0625`

Line To Line Spacing

Specifies the spacing in inches between nets flowing in a channel for each sheet. The spacing for net segments connected to instance pins depends on the pins placed on the symbols of the instances. *Line To Line Spacing* is used for all other net segments. The value must be a decimal number in the range of `0.19` to `0.5` inches.

Default value: `0.2`

Line To Component Spacing

Specifies the spacing in inches between a component and the nearest net flowing in a channel. The value must be a decimal number in the range of `0.19` to `0.5` inches.

Default value: `0.5`

Component Density

Controls the density of a schematic. The value must be an integer from `0` to `100`, where `100` is the most dense and `0` is the least dense.

Default value: `0`

Pin Placement

Specifies the placement of pins on a symbol. Pin placement can be defined as any of the following two options:

- *Left and Right Sides*: Places pins on the left and right sides of a symbol, usually with input pins on the left and inout pins and output pins on the right.

- *All Sides:* Places pins on any side of a symbol.

- *Pin Placement File*: Places pins on specific sides of modules as defined in a file called pin placement file. When you select this option, you must provide the path of the pin placement file in the *Pin Placement File Name* field.

  Pins and their placement on modules are defined in the pin placement file in the following format:

  ```
  pin_placement := {moduleName, {left|right|top|bottom},comma-separated-pin-
  name-list}
  ```

  For example:

  ```
  pin_placement := inv, left,a,b,c,d
  pin_placement := inv, top,e,f
  pin_placement := test4, bottom, c, d
  pin_placement := test4, left, e
  ```

  The first line in the example given above specifies that module `inv` has pins `a`, `b`, `c`, and `d` oriented on the left side.

  Default value: `Left and Right Sides`

Text to Symbol Generator Files

Specifies a space-separated list of tsg files to be used by Connectivity-to-Schematic to generate symbols in the target library. Connectivity-to-Schematic internally runs the Text-to-Symbol generator, a tool that reads the symbol descriptions given in the tsg files to create symbol views.

For more details about the Text-to-Symbol generator and the tsg files, see Text-to-Symbol Generator in Virtuoso Schematic Editor L User Guide.

> ⚠️ *Important*
>
> If the tsg files are given, the direction of pins specified by the Pin Placement option is ignored. Pins are placed on symbols as defined in the tsg files.

Full Place and Route

Specifies whether all the connections should be made by wiring. The following figure shows a fully routed schematic.
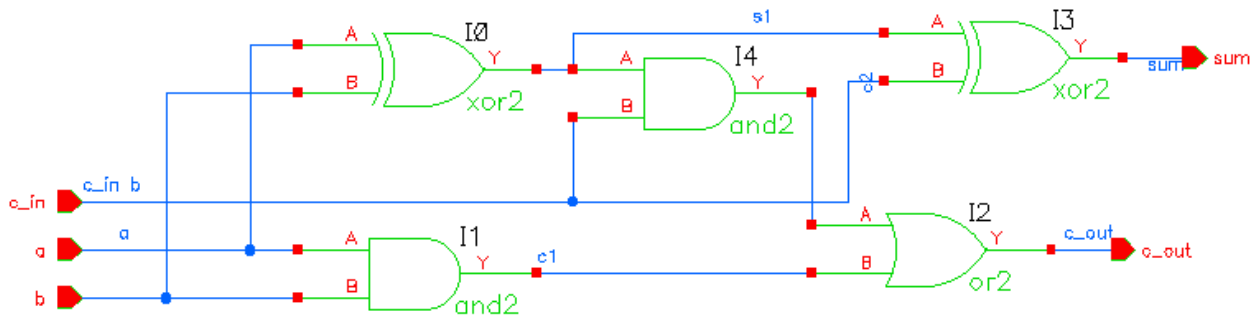
Turn this option off to generate a schematic in which the nets are not explicitly connected. In such a schematic the nets are not wired and the connectivity is indicated by name. This type of schematic is useful for importing large designs where the primary focus is in creating a netlist and simulation and not viewing the design. In addition, generating such a schematic is much faster than generating a completely routed schematic as a significant amount of run-time is spent during routing. The following figure shows a schematic in which the nets are not explicitly routed:
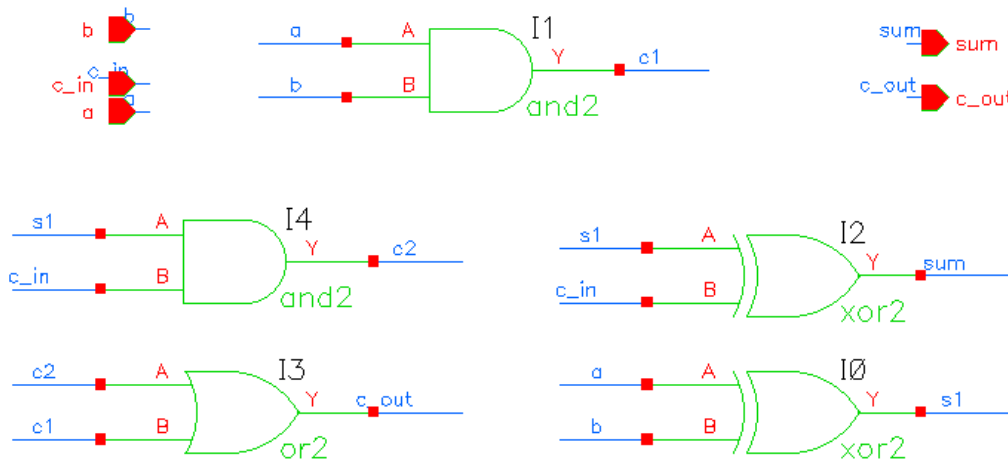
Default value: On

Note: You can also use the Fast Schematic Generation options to determine how nets must be connected when the design has a large number of instances and ports.

Generate Square Schematics

Specifies whether to square the schematic. Turn this option off if you do not want the rows and columns of devices modified to make a rectangular schematic into a square one.

Default value: `On`

Minimize Crossovers

Specifies whether to minimize crossover of nets. Turn this option `on` to minimize crossovers of nets.

Default value: `Off`

Optimize Wire Label Locations

Specifies whether to override default label placement. Turn this option on to override default label placement, which keeps overlap of segments or labels to a minimum, in favor of fast placement, which places labels of segments at the midpoint without checking for minimum overlap.

Default value: `Off`

Extract Schematics

Specifies whether to extract the target cellview in order to report errors and warnings in the schematic. An extracted schematic has blinking markers for the errors or warnings in the schematic view.

Default value: `On`

Verbose

Specifies whether to print detailed status messages while the schematic is being partitioned and routed. Turn this option on to print detailed messages.
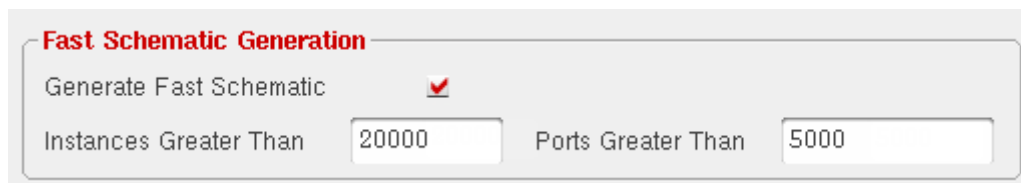
Default value: `Off`

Analog Schematic Generation

Specifies whether to generate an analog schematic using analog placement and routing engine. The options on the *Analog Schematic Options* tab become active only when you turn this option on. The Connectivity-to-Schematic tool generates digital schematic by default.

Default value: `Off`

Fast Schematic Generation

The options in this group box enable fast generation of the schematic when the design being imported contains a large number of instances or ports.

The following table lists the fields in the *Fast Schematic Generation* group box, with their default values and environment variables.

Field	Default Value	Environment Variable
Generate Fast Schematic	On	`generateFastSchematic`
Instances Greater Than	20000	`fastSchematicMaxInst`
Ports Greater Than	5000	`fastSchematicMaxPort`

For details on the environment variables, see "Setting Environment Variables" on page 39.

To use the fast schematic generation feature, select the *Generate Fast Schematic* check box and specify the number of instances and ports in their respective fields. If the number of instances or ports in the design exceeds the specified number, the tool generates a schematic in which the instances are placed in a two-dimensional array without any routing, and the connectivity of the nets is indicated by names.

Notes:

■ To enable the fast schematic generation feature without considering the number of instances or ports in the designs being imported, select *Generate Fast Schematic* and type `0` in the *Instances Greater Than* and *Ports Greater Than* fields.

■   If you disable the fast schematic generation feature, and the design has a large number of instances and ports, schematic generation can take significant time to place and route the instances and nets.

Cellview to Connect Floating Ports/Nets

The fields in this group box connect the floating nets ports in the schematic with a cellview when the source cellview is a netlist view.

The fields specify the library name, cell name, and view name of the cellview that you want to use for connecting to the floating ports and floating nets. For example, to use the `noconn` cellview to connect to the floating ports and nets, you can specify the library as `basic`, cell name as `noconn` and view names as `symbol`. You can either type the library, cell, and view names in the *Library Name*, *Cell Name*, and the *View Name* fields or select these using the *Browse* button.

Note: This feature will work only when the source cellview is a netlist view.

Alternatively, you can use the command-line option `noconn_symbol` or the .cdsenv variable `noconn_symbol`.

Analog Schematic Options Tab

This tab contains options to generate an analog schematic from an imported netlist. The information provided on this tab is passed to the conn2sch executable only if the *Analog*

Schematic Generation option, on the *Schematic Generation Options* tab, is turned on. The following figure shows the *Analog Schematic Options* tab:

Cellview To Be Used For Power Cell

Contains options to specify a cellview that you need to use as a master of a power cell instance. Select the values provided in the *Library Name*, *Cell Name*, and the *View Name* cyclic fields to specify a cellview. Alternatively, you can type the library name, cell name, and view name in these fields. The default values are:

Library Name: analogLib

Cell Name: vdd

View Name: symbol

If multiple instances of a power cellview exist, only a single instance is created while importing a design to ensure that the final schematic contains only one symbol instance of a power cell.

Cellview To Be Used For Ground Cell

Contains options to specify a cellview that you need to use as a master of a ground cell instance. Select the values provided in the *Library Name*, *Cell Name*, and the *View Name* cyclic fields to define a cellview. Alternatively, you can type the library name, cell name, and view name in these fields. The default values are:

Library Name: analogLib

Cell Name: gnd

View Name: symbol

If multiple instances of a ground cellview exist, only single instance is created while importing a design to ensure that the final schematic contains only one symbol instance of a ground cell.

Global Net

Contains options to specify power net and ground net names. Power and ground nets are used to vertically align the instances. Instances connected to a power net are placed towards top of a schematic and instances connected to a ground net are placed towards bottom of a schematic.

You can specify the power net and ground net names in the *Power Net Name* and the *Ground Net Name* fields. Default values for these fields are:

Power Net Name: vdd!

Ground Net Name: gnd!

Generating Schematics

Guidelines to Create and Edit Symbols

You must follow certain guidelines to create and edit symbols, which a schematic uses. If you do not follow these guidelines, the Connectivity-to-Schematic tool might not generate the schematic, or might generate a schematic, which is incorrect or off-grid. Also, nets might overlap symbols and symbol labels might overlap nets or net labels. The guidelines to create and edit symbols are:

■ To prevent extraction errors, make sure the snap spacing is 10 database units (dbu) or greater.

When creating schematics, the Connectivity-to-Schematic tool takes the snap spacing for schematics and symbols from the property, `xSnapSpacing` on the viewType, schematic in the target library. Ideally, snap spacing should be an even number of dbu.

■ To ensure that schematics are on-grid, check that the snap spacing of the target library and that of the symbols in the reference libraries are the same.

If this is not possible, make sure the symbols in the reference libraries have a snap spacing that is a multiple of the snap spacing of the target library.

■ To ensure that schematics are correct, check that the outer edges of all pin figures on each side of the symbol are on the same line (pin figures cannot be recessed from other pin figures) and are abutting the bounding box.

The bounding box is the smallest box enclosing all pin figures and all shapes on the Device-Drawing Layer Purpose pair. Also, make sure that no symbol figures are beyond the pin figures on either side of the symbol shape. You should also specify a unique pin access direction on the pins of the symbol; otherwise, a net might be connected to a pin from a wrong direction.

■ Snap spacing is set lower than 10 dbu in the `schematic` view type in the target library.

■ Snap spacing in the symbol is not a multiple of the snap spacing in the target.

Note: If the pins of a symbol are too close, the input/output pins of the cellview in which the symbol is placed may overlap. In such cases, the input/output pins are displaced and connected by their names to the connecting net.

■ To ensure that schematics are on-grid, make sure all marked lengths are a multiple of the snap space, including the pin-to-pin distances and the origin-to-pin distance.

The Connectivity-to-Schematic tool places the origin of the Composer Symbol on the snap grid. If all marked lengths are not multiples of the snap space, the generated schematic will

be off-grid. Usually this is not a problem because the Symbol Editor automatically imposes this restriction. Problems might occur only if the snap space of the symbol has been changed during or after editing the symbol.

Generating Digital Schematics

The following figure shows the netlist view of the `eight_ip_cell` cell existing in the `sourceLib` library:
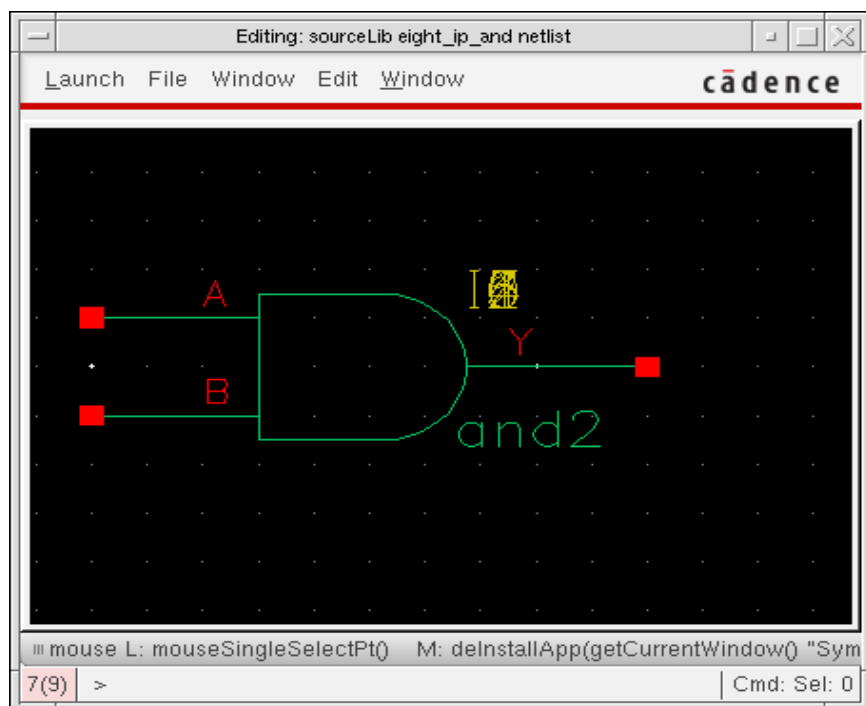
To generate a digital schematic view from the netlist view:

1. Click *File*, point to *Import*, and then click *Netlist View* to run the Connectivity-to-Schematic tool.

2. Specify the names of a source library, source cell, and source view in the appropriate fields on *Connectivity-to-Schematic Options* tab.

3. Specify the names of target library, target cell, and target view in the appropriate fields on the *Connectivity-to-Schematic Options* tab.

    If you do not specify the target library name, the source library is used as the target. Similarly, if you do not specify the target cell name, then the source cell name is used as the target cell name.

4. Click the *Schematic Generation Options* tab to modify schematic generation options.

5. Click *OK* or *Apply* on the *Connectivity-to-Schematic* tab to generate a digital schematic.

6. Wait until Connectivity-to-Schematic displays a completion message as shown in the following figure:
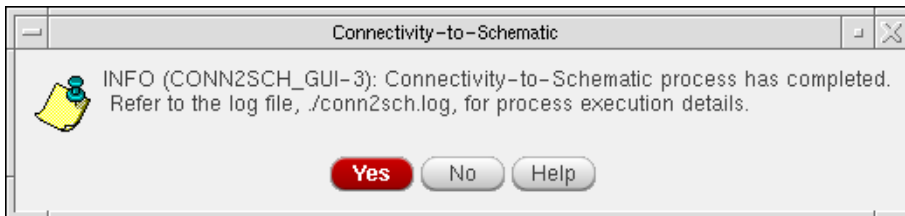
The completion message specifies that a schematic is generated in the specified target library.

7. Click *Yes* to view the log file. The following figure shows the log file contents:
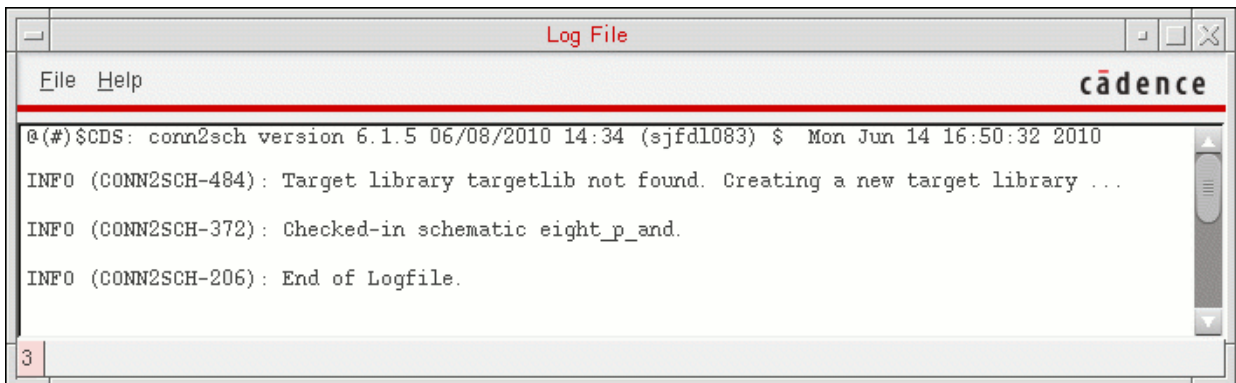
8. Select *File — Open* to open the *Open File* dialog box as shown in the following figure:
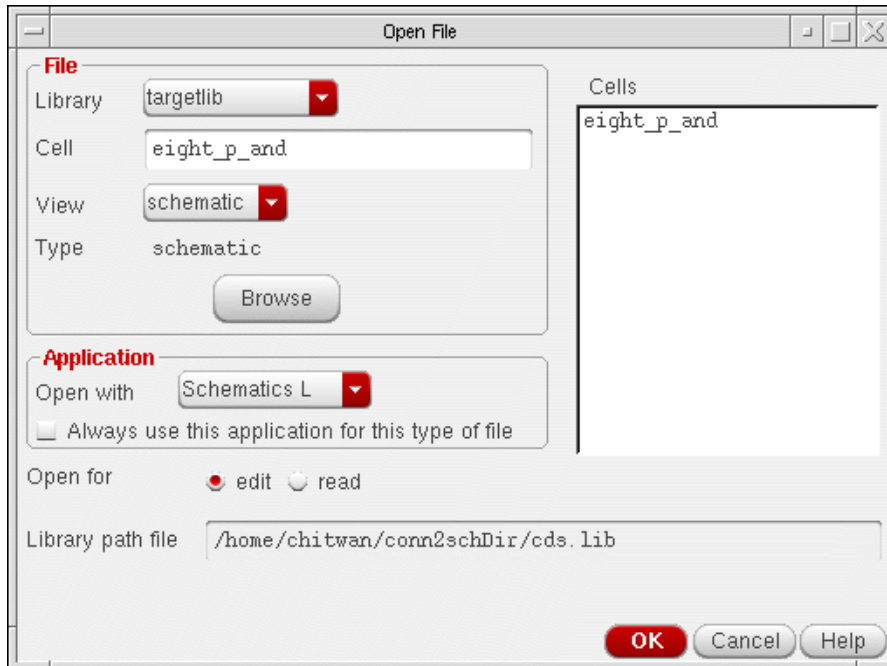
Alternatively, you can open the schematic view from Library Manager.

9. Select *digital_sch* from the *View* drop-down list box on the *Open File* dialog box. The schematic view generated by Connectivity-to-Schematic is displayed in the Virtuoso Schematic Editor L window.

The following figure shows the digital schematic view for the `eight_ip_cell` cell in the `targetlib` library:
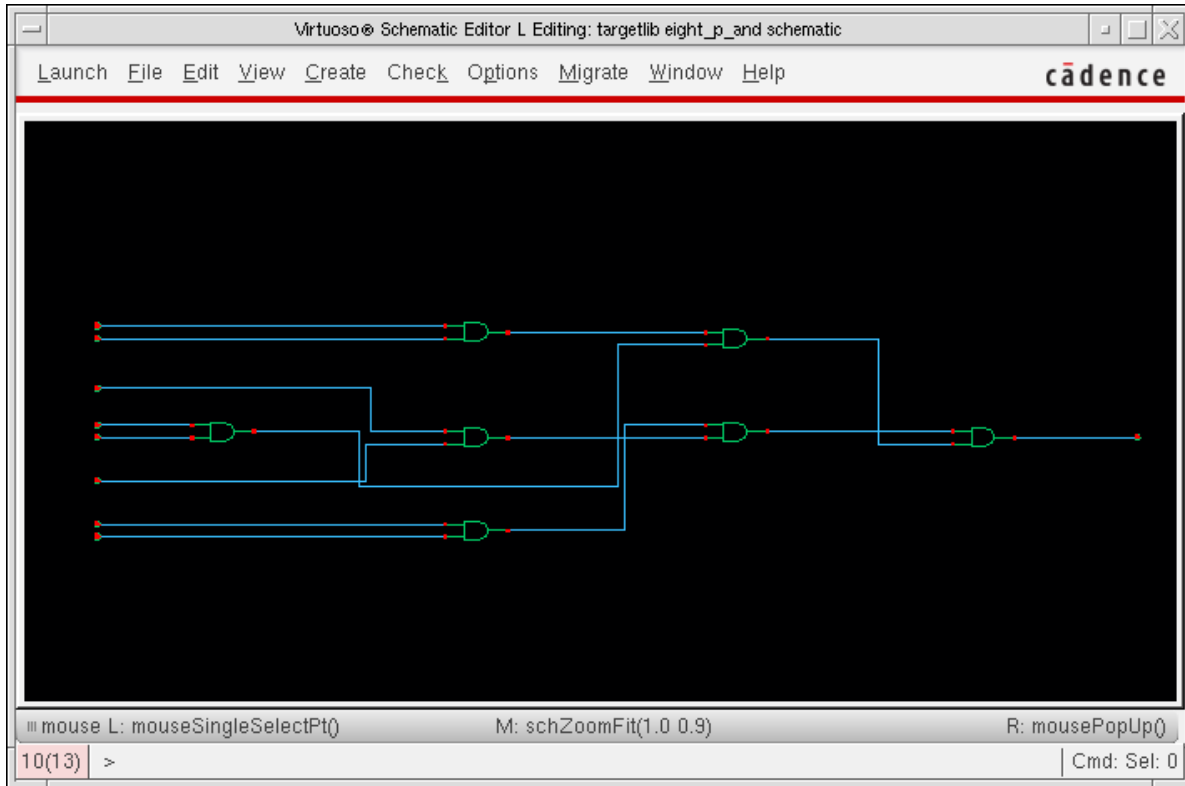
Generating Analog Schematics

The following figure shows the netlist view of the `amplifier` cell existing in the `srcLib` library:
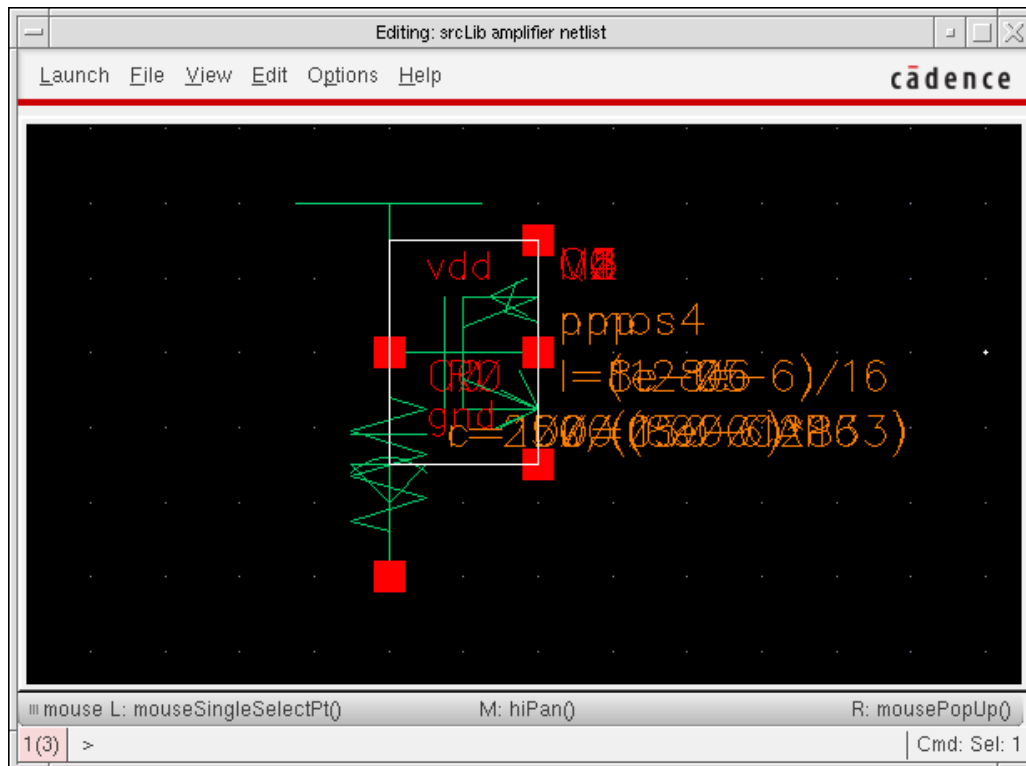
To generate an analog schematic from the netlist view:

1. Click *File*, point to *Import*, and then click *Netlist View* to run the Connectivity-to-Schematic tool.

2. Specify the names of a source library, source cell, and source view in the appropriate fields on *Connectivity-to-Schematic Options* tab.

3. Specify the names of target library, target cell, and target view in the appropriate fields on the *Connectivity-to-Schematic Options* tab.

    If you do not specify the target library name, the source library is used as the target. Similarly, if you do not specify the target cell name, then the source cell name is used as the target cell name.

4. Click the *Schematic Generation Options* tab to modify schematic generation options.

5. Select the Analog Schematic Generation option to activate the options on the *Analog Schematic Options* tab.

6. Click the *Analog Schematic Options* tab to modify analog schematic options.

7. Click *OK* or *Apply* on the *Connectivity-to-Schematic* tab to generate an analog schematic

8. Wait until Connectivity-to-Schematic displays the completion message that a schematic is generated in the specified target library.

9. Click *Yes* on the Connectivity-to-Schematic message box to view the log file.

10. Select *File — Open* to open the *Open File* dialog box.

11. Select `analogSch` from the *View* drop-down list box on the *Open File* dialog box. The schematic view generated by Connectivity-to-Schematic is displayed in the Virtuoso Schematic Editor L window.

The following figure shows the analog schematic view of the `amplifier` cell in the `tarLib` library:

🎞 Video
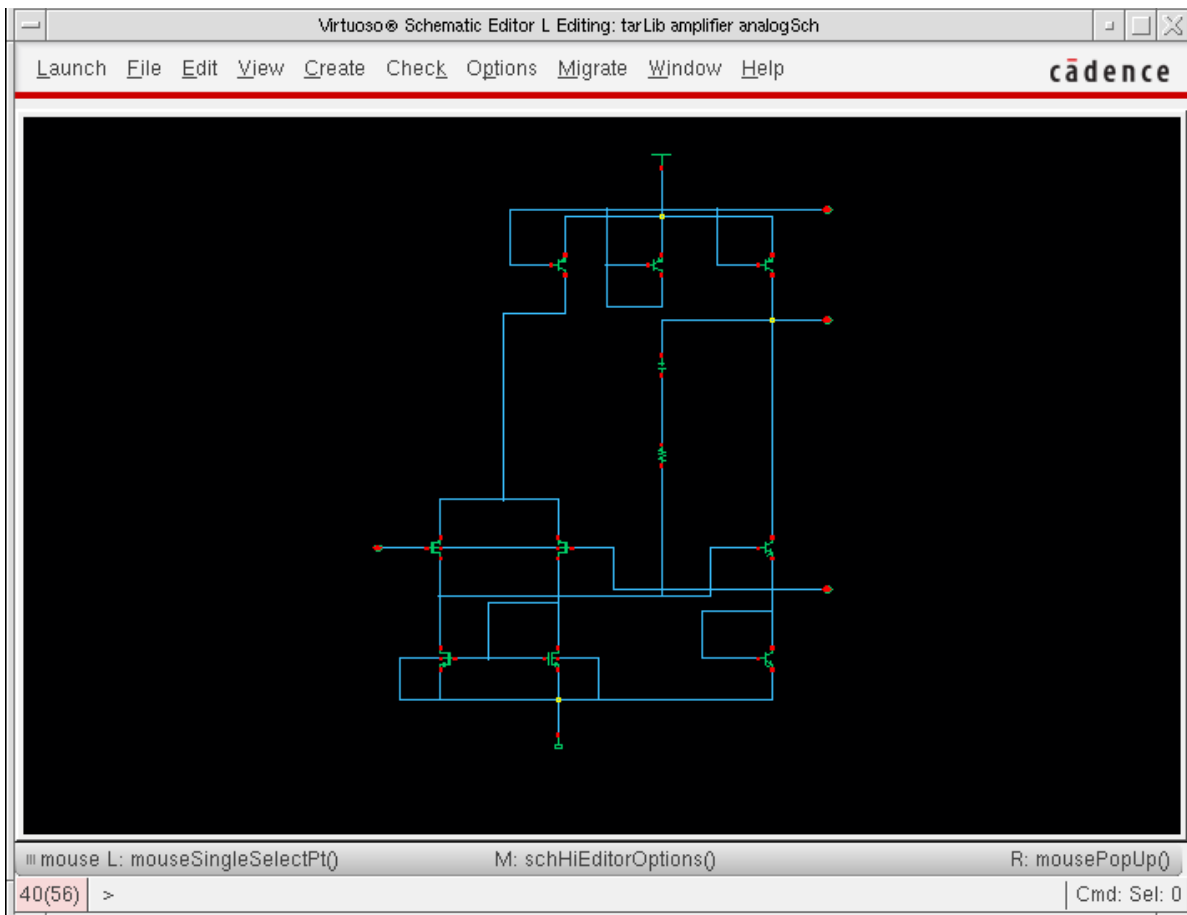
For information on the salient features of Analog Schematic Generator and how it is integrated with Spice In and Connectivity-to-Schematic, see _Analog Schematic Generation_ multimedia demonstration.

Setting Environment Variables

Environment variables are the global values or settings that determine the default behavior of the Connectivity-to-Schematic tool (`conn2sch`). You can change the values of these variables to customize the operation and behavior of the tool. Environment variables are specified in the `.cdsenv` file, which can be accessed from the following path:

```
<inst_dir>/tools/dfII/etc/tools/conn2sch/.cdsenv
```

The following table shows the entries in the `.cdsenv` file and the default values of the variables. Notice that each entry in the `.cdsenv` file begins with the tool name, followed by the environment variable name, variable type, the value, and the range of the variable.

Environment Variable Name	Type	Value	Range
asg_options	boolean	nil	-
density_level	int	0	(0 100)
dest_cell_name	string	""	-
dest_sch_lib	string	""	-
dest_symbol_view_name	string	"symbol"	-
dest_view_name	string	"schematic"	-
extract_schematic	boolean	t	-
fastSchematicMaxInst	int	20000	-
fastSchematicMaxPort	int	5000	-
full_place_and_route	boolean	t	-
generate_square_schematic	boolean	t	-
generateTopCellSymbol	boolean	nil	-
generateFastSchematic	boolean	t	-
ground_net	string	"gnd!"	nil
ground_symbol	string	"analogLib gnd symbol"	-
import_if_exists	int	0	-
label_height	float	0.0625	(0.0375 0.125)
line_component_spacing	float	0.500000	(0.19 0.5)
line_line_spacing	float	0.200000	(0.19 0.5)
log_file_name	string	"./conn2sch.log"	-
minimize_crossovers	boolean	nil	-
optimize_wire_label_locn	boolean	nil	-
page_col_limit	int	1024	(1 1024)
page_row_limit	int	1024	(1 1024)
pin_placement	cyclic	left_and_right_sides	nil, left_and_right_sides, all_sides, file, <file-name>

Environment Variable Name	Type	Value	Range
power_net	string	"vdd!"	nil
power_symbol	string	"analogLib vdd symbol"	–
ref_lib_list	string	"analogLib basic sample"	–
sheet_symbol	string	"none"	–
src_cell_name	string	""	–
src_sch_lib	string	""	–
src_view_name	string	""	–
verbose	boolean	nil	–
noconn_symbol	string	"basic noConn symbol"	–

Using SKILL Functions

The Connectivity-to-Schematic tool provides various SKILL APIs for tasks such as the following:

■ Generating a schematic view from an imported netlist view.

■ Invoking the graphical user interface of the tool.

For details, see *HDL Import and Netlist-to-Schematic Conversion SKILL Reference*.

3

Working In Standalone Mode

The Connectivity-to-Schematic tool can be run either from the Virtuoso® Design Environment workbench or in the standalone mode. This chapter describes the command to run the Connectivity-to-Schematic tool in the standalone mode, the parameter file options, and the command-line options.

Starting Connectivity-to-Schematic in Standalone Mode

The standalone executable, `conn2sch` can be accessed from the following directory:

```
<inst_dir>/tools/dfII/bin
```

The command to start the Connectivity-to-Schematic tool is:

```
conn2sch -lib <libName> -cell <cellName> -view <viewName>
[-param <paramFileName>] [-destlib <destLibName>]
[-destview <destViewName>] [-cdslib <cdsLibFileName]
[-f <optionsFileName] [-log <logFileName>] [+place_only]
[-fast_labels] [-help] [-version] [-min_crossovers]
[-nosquare] [-asg] [+noxtrsch] [-overdense] [-verbose]
```

The `-lib`, `-cell`, and `-view are` mandatory command-line options. The command-line options are not case-sensitive.

Using the Command-line Options

The following table describes the command-line options for the Connectivity-to-Schematic tool:

Option	Description
`-lib <libName>`	Specifies the source library name.

Option	Description
`-cell <cellName>`	Specifies the source cell name.
`-view <viewName>`	Specifies the source view name.
`-param <optionsFileName>`	Specifies the name of a `conn2sch` parameter file.
`-destlib <libName>`	Specifies the destination library name.
`-destview <viewName>`	Specifies the destination view name.
`-log <logFileName>`	Specifies the log file to which the warning and error messages are written. The *-log* option takes precedence over the `log_file_name` variable used in the parameter file.
`-f`	Specifies the file containing all the command line options. `conn2sch -f conn2sch_cmd_opts` where `conn2sch_cmd_opts` contains all the valid command line options.
`-help`	Prints a description for the command line options.
`-version / -v`	Prints the version number.
`-over_dense`	Generates high-density schematics.
`+place_only`	Generates a quick unrouted schematic with connectivity indicated by name.
`-nosquare`	Does not square the schematic; that is, does not manipulate rows and columns of devices to convert a rectangular schematic into a square schematic.
`-min_crossovers`	Minimizes crossover of nets.
`-fast_labels`	Enables faster placement of labels. When this option is on, the Connectivity-to-Schematic tool labels segments at the midpoint and does not check for minimum overlap.
`+noxtrsch`	Does not extract the schematic; that is, does not find errors and warnings that have been written to the OpenAccess databases.
`-cdslib <fileName>`	Specifies the `cds.lib` file to be used.

Option	Description
`-verbose`	Option to print detailed status messages during the schematic placement and routing process.
`-asg`	Specifies that an analog schematic is to be generated using analog place and route engine.

Creating the Parameter File

You can create a parameter file containing a list of all the options and parameter settings that are used by the Connectivity-to-Schematic tool to generate the schematics. This file can be specified with the *-param* command-line option while invoking the tool. The command to use a parameter file is:

conn2sch -param *paramFile* -lib *testLib* -cell *top* -view *netlist*

A sample parameter file, `parameter_file.txt`, is available in the `<install_directory>/doc/conn2schuser/example` directory.

Parameter File Format

Each line in the Connectivity-to-Schematic tool parameter file has a parameter specification statement. The syntax of the parameters follows the Backus Naur format:

parameter_file::= {parameter_specification_statement}

Each parameter specification statement begins with the parameter name followed by the field separator, and then by the value or set of values for the parameter. Following is an example of a parameter specification statement:

parameter_specification_statement::=
parameter_name ps_seperator parameter_value

Parameter name could be any of the following:

parameter_name::=
ref_lib_list |
dest_view_name |
dest_symbol_view_name |
generate_top_cell_symbol |
log_file_name |
import_if_exists |
sheet_symbol |
page_row_limit |
page_col_limit |
label_height |
line_line_spacing |
line_component_spacing |

```
density_level |
pin_placement |
power_symbol |
ground_symbol |
power_net |
ground_net |
noconn_symbol
```

The field separator is ':='

ps_seperator ::=   :=

You can specify either one value or a set of values separated by a comma for the parameters, as follows:

parameter_value ::=   value, {value}

The data type of the value can be a string, an integer, or a real number.

value ::= string_value | integer_value | real_value

In addition, the following rules are applicable for the value:

■   Any value that is not an integer or real is string.

■   String does not require quotes. Trailing and preceding white spaces are automatically removed.

■   White spaces within a string are preserved if the string is enclosed in quotes.

Following is a sample parameter file:

```
ref_lib_list := basic, sample, US_8ths
dest_view_name := schematic
dest_symbol_view_name := symbol
generate_top_cell_symbol := 1
log_file_name := ./conn2sch.log
import_if_exists := 0
sheet_symbol := none
page_row_limt := 1024
page_col_limit := 1024
label_height := 9
line_line_spacing := 0.200000
line_component_spacing := 0.500000
density_level := 100
power symbol := analogLib vdd symbol
ground_symol := analogLib gnd symbol
pin_placement := all_sides
power_net := vdd!
ground_net := gnd!
```

```
noconn_symol := basic noconn symbol
```

The default log file name is `conn2sch.log`. The log file name can be specified as an absolute path. By default, the log file is created in the current working directory. If a log file already exists with the same name, it overwrites the existing file.

Note: The command-line option `-log` takes precedence over the log file name specified in the parameter file.

Parameter File Options

The options used in the schematic generation parameter file are divided into three categories:

■ General Parameter Options

■ Schematic Generation Parameter Options

■ Analog Schematic Generation Parameter Options

General Parameter Options

Option	Description
`ref_lib_list`	Specifies the reference libraries. Use blank space or a comma (,) to separate library names.
	`ref_lib_list := basic, sample, US_8ths`
`dest_view_name`	Specifies the view name to be used for the generated schematic cellview.
	The default view name is `schematic`.
	`dest_view_name := schematic`
`dest_symbol_view_name`	Specifies the view name to be used for the generated symbol view.
	The default view name is `symbol`.
	`dest_symbol_view_name := symbol`
`generate_top_cell_symbol`	Specifies if the symbol of the top cell must be created.
	The default value is `0`. Set the value to `1` to generate the symbol of the top cell.
	`generate_top_cell_symbol := 1`

Option	Description
log_file_name	Specifies a file that lists all error messages and log messages. `log_file_name := ./conn2sch.log`
import_if_exists	Specifies whether to import a cellview for which the schematic view already exists in the target library. When you set this value to 1, the cellview is imported. When you set this value to 0 (default), the cellview is skipped and is indicated in the log file. `import_if_exists := 0`

Schematic Generation Parameter Options

Option	Description
sheet_symbol	Specifies which sheet border size the Connectivity-to-Schematic tool applies when creating a multi-sheet schematic. The sheet borders reside in the US_8ths library. When this value is none, the Connectivity-to-Schematic tool generates an infinite sheet schematic. `sheet_symbol := none`
page_row_limit	Specifies the maximum number of rows on each sheet. Usually, the maximum number of rows is expressed as a number between 1 and 1024. If missing, the limit is automatically set to 1024. `page_row_limit := 1024`
page_col_limit	Specifies the maximum number of columns on each sheet. Usually the maximum number of columns is expressed as a number between 1 and 1024. If missing, the limit is automatically set to 1024. `page_col_limit := 1024`
label_height	Specifies the size of the font used for pin, wire, and instance labels. The value must be an integer greater than or equal to 1. `label_height := 9`

Option	Description
line_line_spacing	Specifies the spacing in inches between nets flowing in a channel. The value must be a real number in the range of 0.125 to 0.625. line_line_spacing := 0.2
line_component_spacing	Specifies the spacing in inches between a component and the nearest net flowing in a channel. The value must be a real number in the range of 0.125 to 0.625. line_component_spacing := 0.5
density_level	Specifies the density of the schematic. The input must be an integer from 0 to 100, where 100 is the most dense and 0 is the least dense. density_level := 100
pin_placement	Specifies the placement of pins on a symbol. Pin placement can be defined in the parameter file with any of the two values: left_and_right_sides or all_sides. For example: pin_placement := left_and_right_sides Alternatively, you can define placement of specific pins on specific sides of a module in a pin placement file, as described in the [Pin Placement](#) section. The pin placement file is referred in the parameter file as: pin_placement := file, pinConn.pl
-noconn_symbol	Connects the floating nets ports in the schematic with the noconn instance. For example: noconn_symbol:= basic noconn symbol

Analog Schematic Generation Parameter Options

Option	Description
`power_symbol`	Specifies a cellview, which you need to use as a master of a power cell instance. `power_symbol := analogLib vdd symbol`
`ground_symbol`	Specifies a cellview, which you want to use as a master of a ground cell instance. `ground_symbol := analogLib gnd symbol`
`power_net`	Specifies a power net, which is used to vertically align the instances. `power_net := vdd!`
`ground_net`	Specifies a ground net, which is used to vertically align the instances. `ground_net := gnd!`

A

Log File Messages

All messages, whether they are warnings or error messages, are directed to the log file. You can specify the log file name with the *-log* command line option or specify it in the parameter file. The default log file name is `conn2sch.log`.

Common Messages

The following are some common messages that may appear in the log file:

- If you have specified all the arguments correctly, but the target library does not exist before starting `conn2sch`, then the following messages appear in the log file:

  **conn2sch -lib src_lib -cell src_cell -view schematic -destLib dest_lib
  -param conn2sch_params**

  ```
  @(#)$CDS: conn2sch.exe version 6.1.2 08/02/2007 16:54 (cic612sun)
  $  Tue Aug  7 10:36:21 2007

  INFO (CONN2SCH-484): Target library dest_lib not found. Creating
  a new target library ...

  INFO (CONN2SCH-372): Checked-in schematic src_cell.

  INFO (CONN2SCH-206): End of Logfile.
  ```

- If the *import_if_exists* variable is set to 0 in the parameter file, then the following messages appear in the log file.

  **conn2sch -param conn2sch_params -lib src_lib -cell src_cell -view schematic
  -destLib dest_lib**

  ```
  @(#)$CDS: conn2sch.exe version 6.1.2 08/02/2007 16:54 (cic612sun)
  $  Tue Aug  7 10:55:59 2007

  ERROR (CONN2SCH-186): Cell src_cell and view schematic already
  exists in the target library. Not importing because the option,
  overwrite-if-exists, is not set to true.

  INFO (CONN2SCH-206): End of Logfile.
  ```

■   If the *import_if_exists* variable is set to 1 in the parameter file, then the following
   messages appear in the log file.

   **conn2sch -param conn2sch_params -lib srcl_lib -cell src_cell -view schematic
   -destLib dest_lib**

   ```
   @(#)$CDS: conn2sch.exe version 6.1.2 08/06/2007 15:12 (cic612sun)
   $  Tue Aug  7 11:18:12 2007

   WARNING (CONN2SCH-185): Cell src_cell view schematic already
   exists in the target library. Overwriting this cell view because
   the option, overwrite-if-exists, is set to true.

   INFO (CONN2SCH-372): Checked-in schematic src_cell.

   INFO (CONN2SCH-206): End of Logfile.
   ```

Error Messages

The following are some common errors and the corresponding messages that appear in the
log file.

■   If the specified source cellview does not contain the instances and connectivity
   information, the following messages appear in the log file and Connectivity-to-Schematic
   exits:

   conn2sch -lib srcLib -cell srcCell -view srcView -destLib destLib

   ```
   @(#)$CDS: conn2sch.exe version 6.1.2 08/06/2007 15:12 (cic612sun)
   $  Tue Aug  7 11:22:58 2007

   INFO (CONN2SCH-199): Source cell view NewCell has no instances.

   ERROR (CONN2SCH-200): The cell view srcCell does not have
   connectivity information. Therefore unable to create schematic
   for this cell view. Exiting ...

   INFO (CONN2SCH-206): End of Logfile.
   ```

■   If you specify invalid arguments for the -lib, -cell, or -view options then the
   following messages appear in the log file.

   **conn2sch -param conn2sch_param -lib srcLib -cell srcCell -view schematic
   -destLib destLib**

   ```
   @(#)$CDS: conn2sch.exe version 6.1.2 08/06/2007 15:12 (cic612sun)
   $  Tue Aug  7 11:26:58 2007

   ERROR (CONN2SCH-188): Could not open cell src_cell_new view
   schematic in library src_lib_new.
   ```

```
INFO (CONN2SCH-206): End of Logfile.
```

■ If you do not specify the parameter file name with the `-param` option, then the following message appears on the terminal window.

conn2sch -param -lib src_lib -cell src_cell -view schematic -destLib dest_lib

```
@(#)$CDS: conn2sch.exe version 6.1.2 08/06/2007 15:12 (cic612sun)
$: (c) Copyright 1994-2006, Cadence Design Systems, Inc.
```

```
ERROR (CONN2SCH-3): Parameter file name has not been specified.
You must specify a valid parameter file name after the -param
option.
```

B

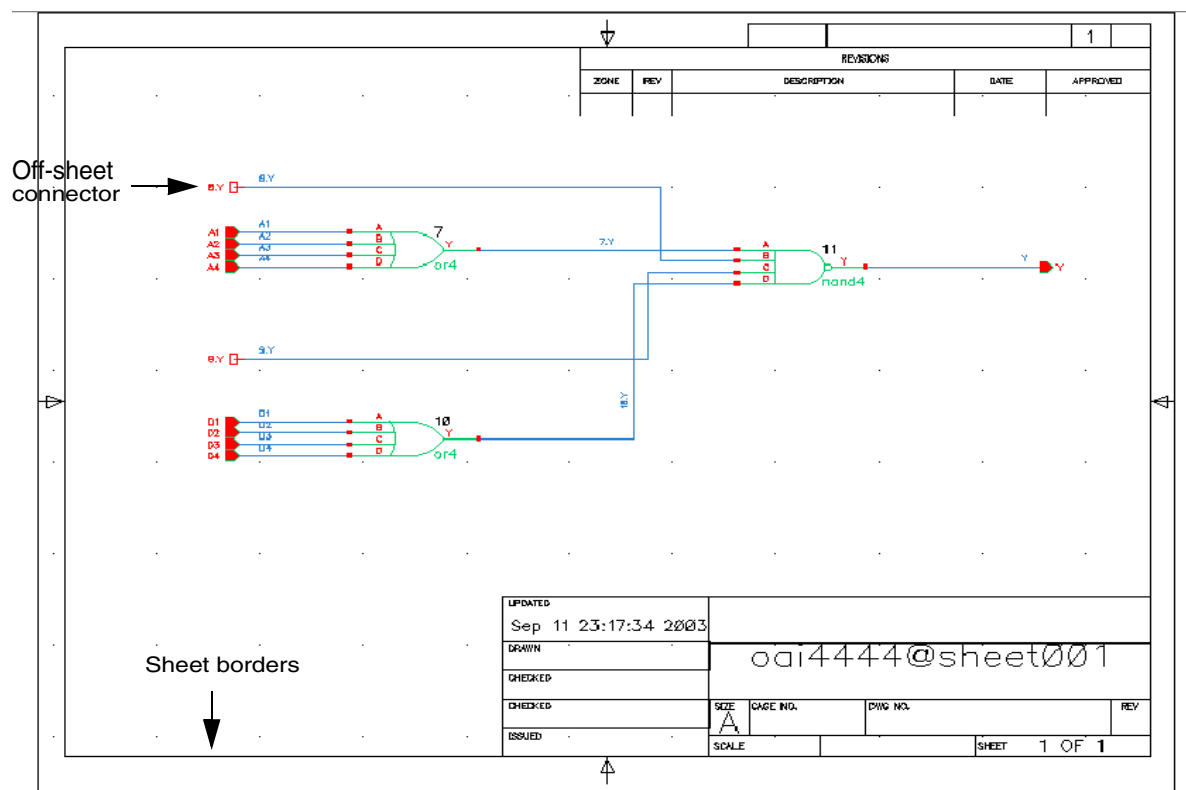# Managing Index and Multi-Sheet Schematics

---

Managing Multi-Sheet Schematics

Connectivity-to-Schematic does not create multiple sheets when the source cellview is a multi-sheet schematic cellview. Therefore, all off-sheet connectors in the input are converted to cellview terminals. For example, if you provide a sheet of a multi-sheet schematic as input, then although you get a valid single-sheet schematic, the off-sheet connectors are converted to cellview terminals. Also, the generated schematic does not have any sheet border.

The following figure shows a sheet from a multi-sheet schematic that is provided as input to Connectivity-to-Schematic:

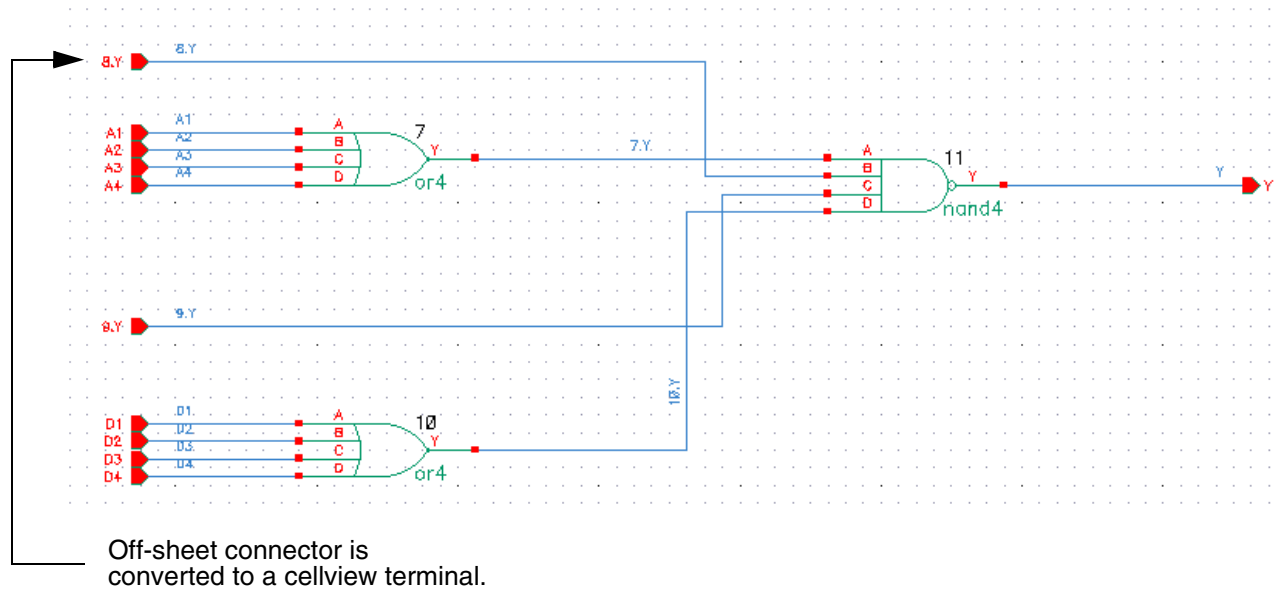Input sheet with an off-sheet connector

The following figure shows the output that Connectivity-to-Schematic generates for the multi-sheet schematic:

Off-sheet connector is
converted to a cellview terminal.

Managing Index Schematics

If you provide an index schematic as input, then all the m-symbols are converted into a schematic representation with wires and pins. The following shows the index schematic that is provided as input to Connectivity-to-Schematic:
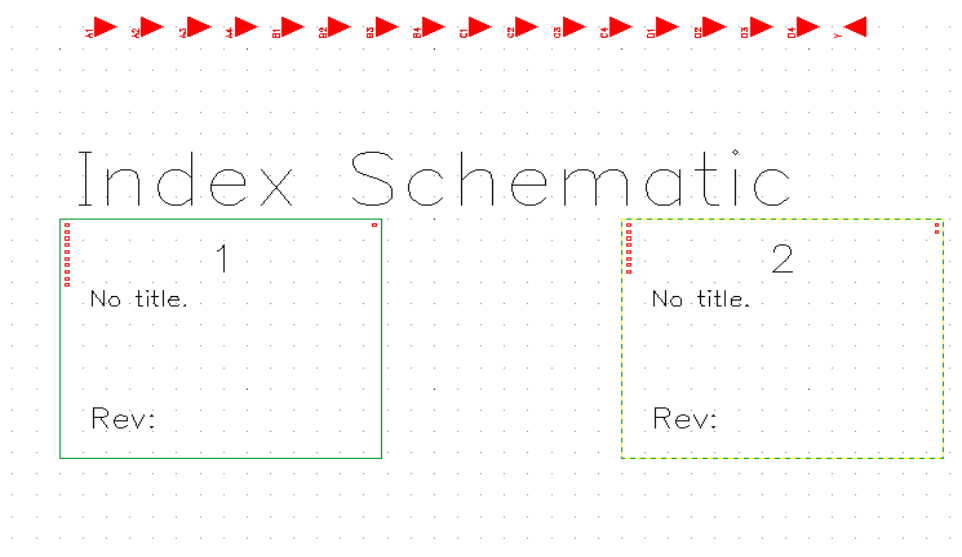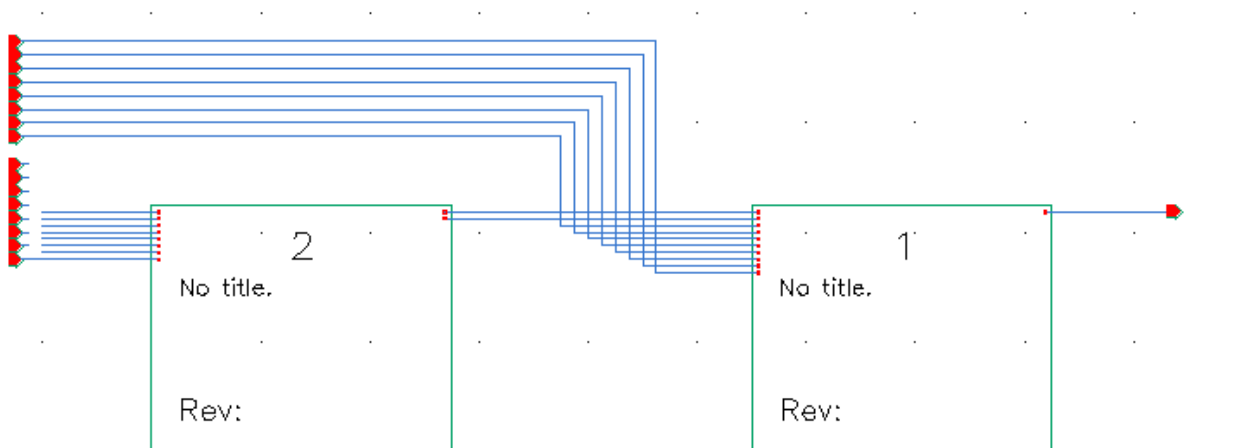

Index sheet as input to the Connectivity-to-Schematic tool

The following shows the output that Connectivity-to-Schematic generates for an index schematic:

m-symbols of an Index sheet are wired in the generated schematic.