Agilent Technologies

Advanced Design System 2002

Netlist Translator for SPICE and Spectre

February 2002

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Agilent Technologies
395 Page Mill Road
Palo Alto, CA 94304 U.S.A.

Contents

1 Introduction

2 Importing a Netlist File

3 Advanced Methods for Importing Files

Chapter 1: Introduction

Spectre is an EDA (Electronic Design Automation) tool produced by Cadence Design Systems, Inc. The Simulation Program with Integrated Circuit Emphasis (SPICE) was developed at University of California Berkeley and has been commercialized and modified by a large number of vendors. It has also been adopted and modified by electronics companies for their own in-house use. Spectre and SPICE are used by engineers throughout the world for simulating circuits of all types. Many designers and companies have large investments in existing subcircuits or device models described by Spectre and SPICE netlists that they want to use with the Advanced Design System (ADS) from Agilent Technologies.

Advanced Design System

Advanced Design System has been developed specifically to simulate the entire communications signal path. This unique solution integrates the widest variety of proven RF, DSP, and electromagnetic design tools into a single, flexible environment. Building on years of expertise developing new technologies for our EDA tools, such as Series IV and MDS, Advanced Design System provides a broad range of high-performance capability. This makes it easy to explore design ideas and model the electrical and physical design of the best candidates.

Netlist Translator

Providing the ability to simulate a Spectre or SPICE netlist in ADS is a fundamental purpose of the Netlist Translator. The Netlist Translator is an ADS tool that enables you to import netlists into schematics and/or netlists for use with the ADS simulator.

Since Spectre, SPICE, and ADS simulators use different simulator technology, there may be incompatibilities between your original file and the resultant ADS file that the translator is unable to resolve. The incompatibilities may show up as import errors, simulation errors or differences in simulation results. Every attempt has been made to record these inconsistencies in the translator log file, *spctoiff.log*. Detailed information in this manual will help you identify and understand these limitations.

Major Benefits

The Netlist Translator enables you to perform the following:

- Generate an ADS schematic from a Spectre or SPICE Netlist for use in circuit design and simulation.
- Generate an ADS netlist from a Spectre or SPICE Netlist for use in simulations.
- Translate model files to incorporate them into ADS Design Kits.

Major Features

Key features of the Netlist Translator include the following:

- Automatic ADS design generation that includes circuit schematic, required model definitions and circuit equations or a reference to an equivalent ADS netlist to be used directly in simulation.
- Translation of component and model parameters to equivalent ADS parameters. In some cases this requires additional equations if the parameters do not translate directly. Parameters that cannot be translated are listed in the translator log file, *spctoiff.log*.

Supported SPICE and Spectre Dialects

The Netlist Translator supports several SPICE dialects that are widespread throughout the electronics industry. Table 1-1 lists the four main dialects supported by the translator.

Table 1-1. Supported SPICE and Spectre Dialects

Name	Manufacturer	Version
Spice2	UC Berkeley	2G6 (1981)
Spice3	UC Berkeley	3F3 (1993)
PSpice	Cadence	9.9.2 (August 2001)
HSpice	Avanti/Meta Software	v2001.2 (June 2001)
Spectre	Cadence	4.4.6

Note For translation purposes, Spice2 and Spice3 are almost identical and are referenced in this manual as Spice2/3.

General Process

The main objective of using the Netlist Translator is to import your Spectre or SPICE netlist into ADS. This enables you to simulate your design using the powerful tools provided by ADS. Figure 1-1 is a simplified task flow for using the Netlist Translator.

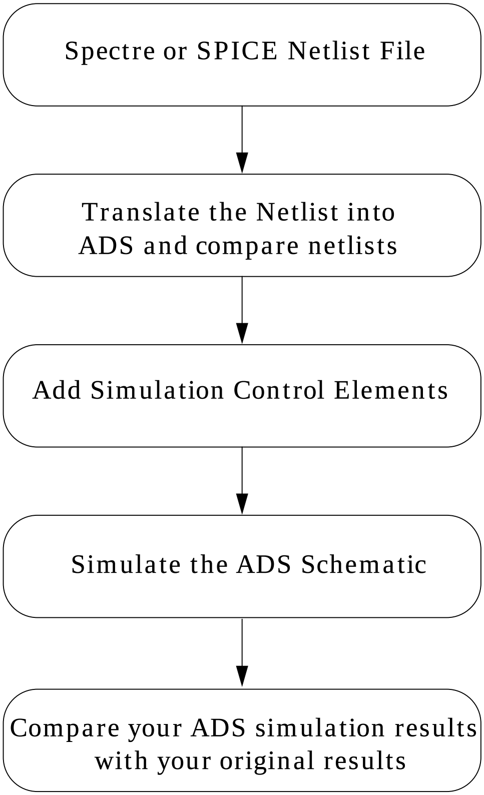

Spectre or SPICE Netlist File
Translate the Netlist into ADS and compare netlists
Add Simulation Control Elements
Simulate the ADS Schematic
Compare your ADS simulation results with your original results

Figure 1-1. Simplified Task Flow

What’s in this Manual

The goal of this manual is to help you get started, provide relevant examples that teach you how to use the software, assist you in analyzing your translation and show you where you can get more information as you need it.

- Chapter 2, Importing a Netlist File provides detailed instructions on using the Netlist Translator’s user interface to import a netlist file into an ADS schematic or to an ADS netlist. Several example Spectre and SPICE netlists are used to help you understand and practice using the translator.
- Chapter 3, Advanced Methods for Importing Files provides detailed instructions for advanced users who want to understand how to use the command line options that are available within the Netlist Translator. Additional information is provided to help with customizing components and other unique situations.
- Chapter 4, Simulating the Translated Netlist provides general information on setting up, performing and analyzing a simulation using Advanced Design System. References to more detailed information are also provided in this chapter.
- Chapter 5, Comparing Results provides information on comparing the output of your translation to your original netlist file. This chapter points out some specific details to look for when comparing your results.
- Chapter 6, Troubleshooting provides helpful information on debugging imported designs and includes information on error and warning messages. Some known problems are described and solutions are included where available.
- Chapter 7, Translating a Deviceprovides detailed device translation information for individual components.
- Chapter 8, Translating a Model provides detailed model translation information for individual models.
- Chapter 9, Adding User Defined Model Translations for Spectre provides information on adding your own unique models to ADS.
- Chapter 10, Translating Commands and Functions provides information on the various commands and functions supported by the Netlist Translator.

About Design Translation and Verification

Design translation can be a very complicated process. It is important to understand that it is rarely, if ever, a push-button utility. Before starting a translation, you should become familiar with all parts of your design and be prepared to spend adequate time to ensure that all parts are translated correctly. After the initial translation is complete, additional work may be required to understand model differences. Verification and debugging the translated design may involve breaking a complex design down into smaller pieces to test simulation of small circuits before attempting to simulate a large design.

The support staff at Agilent Technologies can assist you in this process, but should not be expected to perform a complete translation and simulation. Please plan your project accordingly, allowing sufficient time in your schedule to use this tool as it is intended-as a tool to assist in the transfer of design information from one system to another.

Chapter 2: Importing a Netlist File

This chapter describes the procedures for importing Spectre or SPICE netlist files into Advanced Design System using the Netlist Translator's User Interface. Performing a translation via the User Interface enables you to easily define the translation criteria and translate your netlist file. You have the option of generating either an *ADS Schematic* or an *ADS Netlist plus an ADS Schematic* with a NetlistInclude component depending on your selections.

If you have a relatively simple netlist file and only need a schematic, the *ADS Schematic* option is recommended. However, if you have a large or complex netlist source file or you are just more comfortable viewing a netlist version, the *ADS Netlist plus an ADS Schematic* with a NetlistInclude component option enables you to review the netlist. This makes it easy to compare your files line by line for debugging purposes. The netlist version also provides much faster processing.

Importing a Netlist File Using the User Interface

Importing a netlist file to an ADS Schematic or an ADS Netlist with a NetlistInclude component using the Netlist Translator can be broken down into several simple steps.

1. "Accessing the Import Dialog" on page 2-3
2. "Specifying the File Name" on page 2-4
3. "Setting the Import Options" on page 2-5
4. "Choosing the Translated Output Format" on page 2-8
5. "Defining Netlist Options" on page 2-13 *
6. "Examining the Output" on page 2-10
7. "Using the NetlistInclude Component" on page 2-25 *

Note Steps 5 and 7 above are only necessary if you want to import to an ADS Netlist with a NetlistInclude component.

Creating a Simple Spectre Example

The following simple RCL-Diode circuit will be will be referred to periodically in the documentation to help with your understanding of the import process. If you like, you can create the sample Spectre file shown in Table 2-1. You can create this file in your project directory using any ASCII text editor. Type the following text into a file.

Table 2-1. Sample Spectre File (ex1.scs)

```
*R, C, L & Diode Example
L1 (net1 net2) inductor l=1n
R1 (net2 net3) resistor r=1K
C1 (net3 net4) capacitor c=1p
D1 (net4 net5) pdiode l=3e-4 w=2.5e-4 area=1
model pdiode diode is=1.8e-5 rs=1.43 n=1.22
```

Save this file as *ex1.scs* in your current project directory then close the file. You can now use *ex1.scs* as an example the first time you attempt to perform an import operation.

Creating a Simple SPICE Example

Create the sample SPICE file shown in Table 2-1. You can create this file in your project directory using any ASCII text editor. This, and other simple SPICE files, will be referred to periodically in the documentation to help with your understanding of the import process. Type the following text into a file.

Table 2-2. Sample SPICE File (ex1.sp)

```
*SIMPLE RTL INVERTER
VCC 4 0 5
VIN 1 0 PULSE 0 50 2NS 2NS 2NS 30NS
RB 1 2 10K
Q1 3 2 0 MODQ1
RC 3 4 1K
.MODEL MODQ1 NPN BF=20 RB=100 TF=.1NS CJC=2PF
.DC VIN 0 5 0.1
.TRAN 1NS 100NS
.END
```

Save this file as *ex1.sp* in your current project directory then close the file. You can now use *ex1.sp* as an example the first time you attempt to perform an import operation.

Accessing the Import Dialog

You can import files through Advanced Design System's *Main* or *Schematic* window. The *Import* dialog box is accessed from the File pull-down menu.

Open a project in ADS before importing your design. Working in project directories enables the translator to organize design files in the standard ADS file structure.

Choose **File > New Project** to open a new project or **File > Open Project** to open an existing project.

Note The import option will not be active in the File menu unless you open a project.

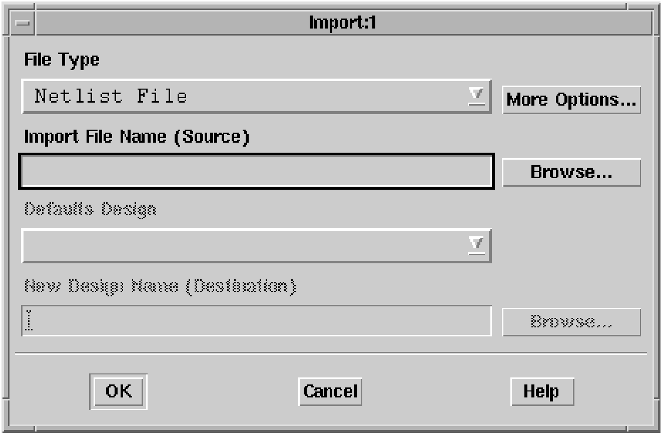
For more information on working in project directories, refer to "*Managing Projects and Designs*" in the Advanced Design System *User's Guide*.

Before invoking the import procedure, close any open designs. This will remove any active designs from memory.

Choose **File > Close All**

To access the import dialog and import your design from the ADS *Main* or *Schematic* window:

Choose **File > Import**. The *Import* dialog box appears.



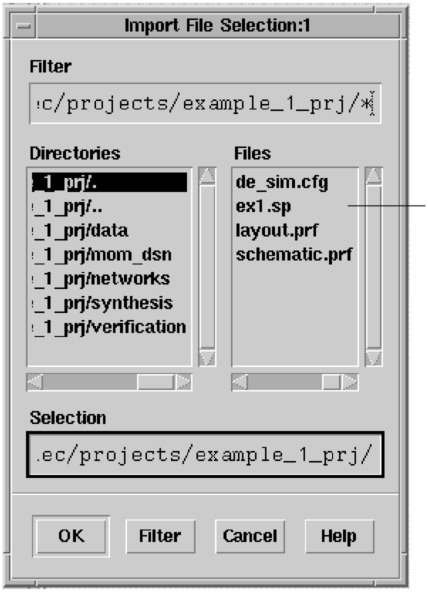
File Type: Netlist File

Specifying the File Name

In the *Import* dialog box, choose the type of file to import, specify the filename and supply other basic information needed by the translator.

1. In the *Import* dialog box, select *Netlist File* from the *File Type* drop-down list if it isn't already displayed.

2. To specify the path and filename of the file you want to import, click **Browse** in the *Import* dialog box. The *Import File Selection* dialog box appears.



Filter: c/projects/example_1_prj/*

Directories:
- _1_prj/.
- _1_prj/..
- _1_prj/data
- _1_prj/mom_dsn
- _1_prj/networks
- _1_prj/synthesis
- _1_prj/verification

Files:
- de_sim.cfg
- ex1.sp
- layout.prf
- schematic.prf

Selection: .ec/projects/example_1_prj/

Select your *ex1.scs* example file

3. Double-click as needed to locate the directory containing your source file in the *Directories* field, then click the file in the *Files* field. Alternatively, you can enter the full path and file name in the *Selection* field.

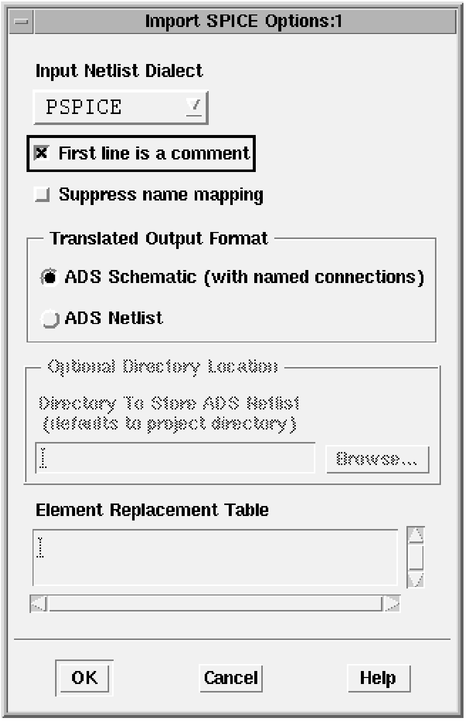
4. After selecting the design you want to import, *ex1.scs* for this example, click **OK**. You are returned to the Import dialog box and the selected filename appears in the field labeled *Import File Name (Source)*.

5. Click **More Options** to define the preferred import method. The *Import SPICE Options* dialog box appears.

Setting the Import Options

Selecting the Dialect

In the *Import SPICE Options* dialog box, select the appropriate netlist dialect from the *Input Netlist Dialect* drop-down list.



Import SPICE Options:1

Input Netlist Dialect: PSPICE

☒ First line is a comment

□ Suppress name mapping

Translated Output Format

● ADS Schematic (with named connections)

○ ADS Netlist

Optional Directory Location

Directory To Store ADS Netlist
(defaults to project directory)

Element Replacement Table

If you are unsure of the source of the netlist file, you may be able to tell from the syntax of in line comments shown in Table 2-3.

Table 2-3. Recognizing Netlist Variations

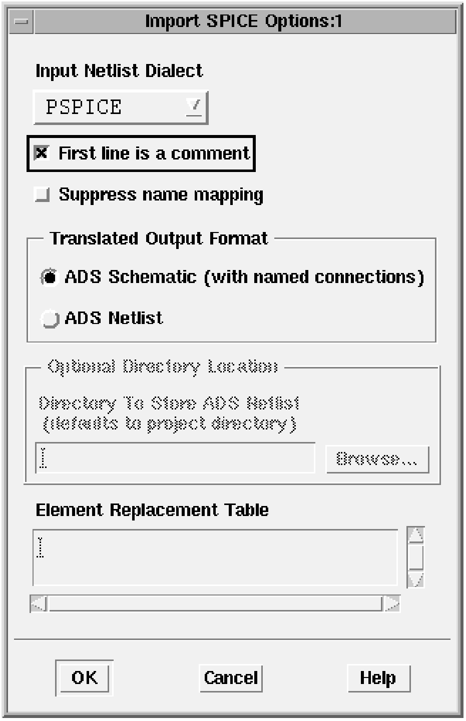
SPICE Dialect	In line comment syntax
Spice 2G	In line comments not allowed
Spice 3	In line comments not allowed
PSpice	;
HSpice	$
Spectre	//

Note For HSpice, the '$' must be preceded by a space or a comma if it is not the first non-blank character because the $ is allowed within node or element names.

You may also be able to match syntax with the device table or model levels in the model tables. For more information on device and model tables, refer to Chapter 7, Translating a Device and Chapter 8, Translating a Model.

First line is a comment:

To make sure the first line of a source file is recognized as a comment line and ignored, the *First line is a comment* checkbox must be selected. This is not necessary if the line already begins with a comment character.



Import SPICE Options:1

Input Netlist Dialect

PSPICE

☒ First line is a comment

□ Suppress name mapping

Translated Output Format

● ADS Schematic (with named connections)

○ ADS Netlist

Optional Directory Location

Directory To Store ADS Netlist
(defaults to project directory)

Element Replacement Table

The *First line is a comment* option is available because some simulators will always ignore the first line regardless of the first character.

Use the default value for the *ex1.scs* example.

Suppress name mapping:

When the translator changes the name of a node or a component to comply with ADS rules, this is referred to as *name mapping*. In ADS2001, name mapping is limited to all names being changed to lowercase, as well as replacement of an illegal character with an underscore. For more information on name mapping, refer to "Using Valid ADS Characters" on page 2-31.

Click the *Suppress name mapping* check box to activate or deactivate this option. Activating this option enables you to override any name mapping introduced by the translator.

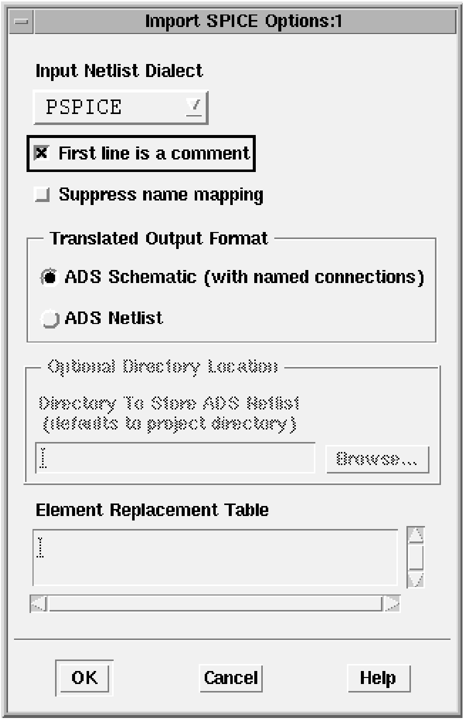
Note See "Understanding Capitalization" on page 2-31 for cautionary information on name mapping.

Use the default value for the *ex1.scs* example.

Choosing the Translated Output Format

In the *Import SPICE Options* dialog box, there are 2 import methods available in the *Translated Output Format* field: *ADS Schematic (with named connections)* or *ADS Netlist*. See "Translated Output Format Descriptions" on page 2-9 for format descriptions.

Note A third option, *Wires*, is available through the command line only. See "Translated Output Format Descriptions" on page 2-9 for more information.



Import SPICE Options:1

Input Netlist Dialect

PSPICE

☒ First line is a comment

□ Suppress name mapping

Translated Output Format

● ADS Schematic (with named connections)

○ ADS Netlist

Optional Directory Location

Directory To Store ADS Netlist
(defaults to project directory)

Element Replacement Table

Translated Output Format Descriptions

- **ADS Schematic (with named connections)** This translation method writes an intermediate file in the *Intermediate File Format* (see "Importing an IFF File" on page 3-10). After the IFF file is created, it is read by ADS and the schematic is created. For detailed information on how this works, refer to "Understanding the Import Operation" on page 3-1. Components are placed in a square array and Named Connections are used to place the correct node name on each pin. Pins with the same node name are connected. This translation is faster than the *Wires* option that tries to connect all the pins with wires.
- **ADS Netlist** An ADS netlist is written to the project directory and a simple schematic is created. The schematic contains only a *NetlistInclude* component that references the ADS netlist file. When an ADS simulation is run, the simulator automatically reads the new netlist. This saves a lot of time on very large circuits both in translation time and simulation time, since the schematic creation and traversal are both skipped. For information on *Netlist Options*,

refer to "Defining Netlist Options" on page 2-13. For information on editing the netlist, refer to "Viewing and Editing the Netlist" on page 2-16.

- **Wires (Available from command line only)** This translation method also writes an intermediate file in the IFF format. After the IFF file is created, it is read by ADS and the schematic is created. For detailed information on how this works, refer to "Understanding the Import Operation" on page 3-1. Components are placed and wired with a special algorithm. The larger the file is, the slower this will be. This method should only be used on very small circuits.

Table 2-4. Import Methods

Method	Advantage	Disadvantage	When to Use
ADS Schematic (with named connections)	Fast, easy to read	Can be difficult to tell which components are connected together	Default--use in most cases
ADS Netlist	Allows large circuits to be modular	Cannot edit the components of the netlist through UI	Large netlists that do not require editing through UI
Wires (available from command line only)	For small circuits, easy to tell which components are connected together	Slow, messy for large circuits	Small netlists

IMPORTANT It is highly recommended that the *ADS Netlist* option be used when translating large netlist file. In addition to taking many hours to create a schematic from a 20,000 line netlist, a schematic of that size is practically non-editable.

For the purpose of the *ex1.scs* example, select the *Named Connections* method and click **OK** in the *Import SPICE Options* dialog box. You are returned to the *Import* dialog box. Click **OK** to begin the translation.

Examining the Output

This section describes some of the information to look for during and after your translation to help you determine whether your translation is complete or not.

Viewing the Status Window

While the translator is running, a *Status* window appears and displays messages about the translation as it progresses (see Table 2-5). When the translation is complete, the Status window displays the message, *Translation completed*, along with date and time done.

Table 2-5. Example Status Window Contents

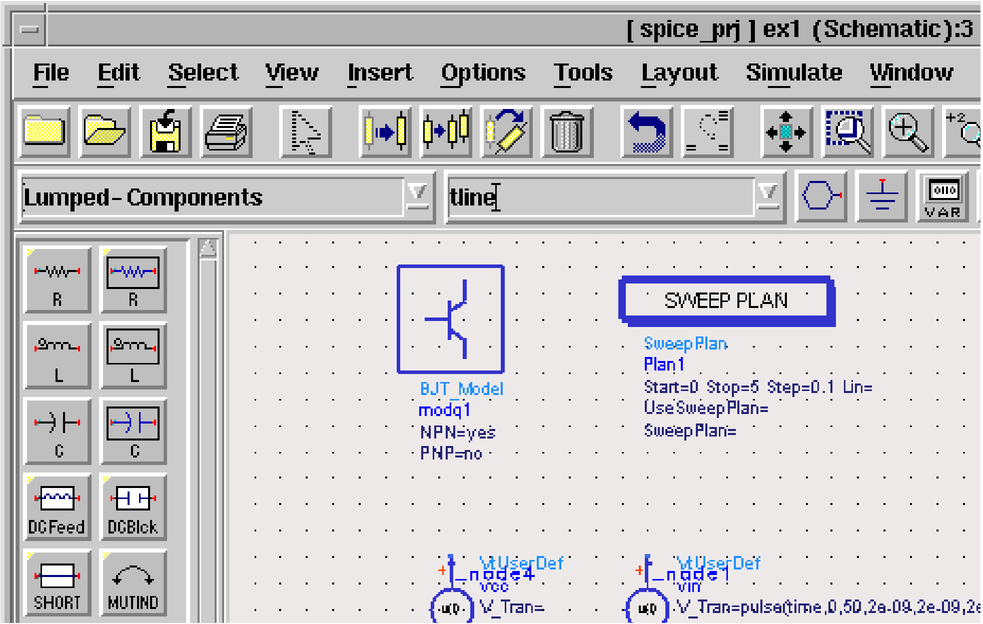
```
Netlist Translator (*) 170.day May 31 2001
IFF translation log

Input format: PSpice
Input filename:
/a/new/sr/ignite/d1/users/steelec/projects/spectreExample/ex1.scs

Output format: IFF file
Output filename: spice.iff

Special options:
   Processing first line as comment.

Begin translation at Thu May 31 12:35:03 2001

Creating schematic with named connections.
Reading item definition file
"/hped/builds/sr/dev170/rday/opt/prod/config/spctoiff.cfg"

Translation completed at Thu May 31 12:35:04 2001.

=============================================================
================
End of Netlist Translation Log file, beginning of IFF Translation Log File
=============================================================
================

Beginning IFF import
Processing IFF file: spice.iff
Creating design ex1
IFF import complete
```

The Status window reports information such as the input file format, the input file name and the output format. The output format for Advanced Design System is the *IFF file* in this case. The translation start time is also displayed.

After the translation is complete, check to see if there were any error or warning messages reported. If any errors or warnings are displayed in the Status window, refer to Chapter 6, Troubleshooting for information on unexpected results from the Netlist Translator and how to fix the problem.

Viewing the Schematic

Open a new schematic window and then open the *ex1.dsn* design to view the schematic. For more information on schematic windows, refer to “*Working in Design Windows*” in the ADS *User's Guide*.



BJT_Model
modq1
NPN=yes
PNP=no

SweepPlan
Plan1
Start=0 Stop=5 Step=0.1 Lin=
UseSweepPlan=
SweepPlan=

V_Tran=

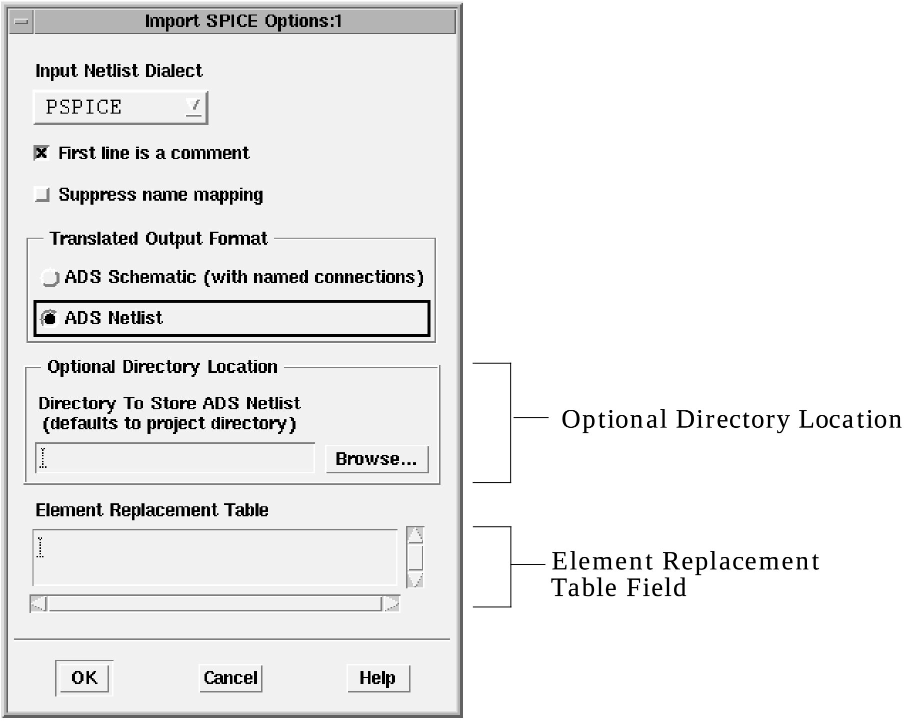
V_Tran=pulse(time,0,50,2e-09,2e-09,2e

Figure 2-1. ADS Schematic Window with Translated *ex1.dsn*.

Before simulating your design in ADS, you will need to place simulation control blocks from one of the simulation palettes in the schematic window. You may also need to place an instance of an imported design in a new schematic. For more information on setting up your simulation, refer to Chapter 4, Simulating the Translated Netlist.

Defining Netlist Options

If you selected *ADS Netlist* in "Choosing the Translated Output Format" on page 2-8, the *Optional Directory Location* area of the *Import SPICE Options* dialog is activated. Specify the component name (use all lower case), number of pins and an output directory in each of the respective fields, according to the guidelines below.



Import SPICE Options:1

Input Netlist Dialect: PSPICE

☒ First line is a comment

□ Suppress name mapping

Translated Output Format

○ ADS Schematic (with named connections)

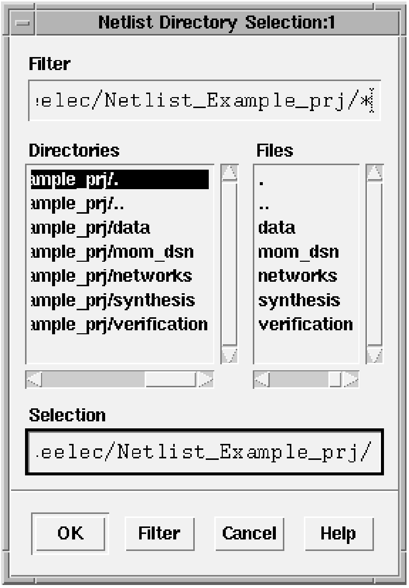
● ADS Netlist

Optional Directory Location

Directory To Store ADS Netlist (defaults to project directory)

Element Replacement Table

- **Directory to Store Netlist** To specify the path of the directory you want to store your netlist in, click **Browse**. The *Netlist Directory Selection* dialog appears. If you do not choose a directory to store your netlist, it will default to your current project directory.



Netlist Directory Selection:1

Filter

:elec/Netlist_Example_prj/*

Directories

ample_prj/.
ample_prj/..
ample_prj/data
ample_prj/mom_dsn
ample_prj/networks
ample_prj/synthesis
ample_prj/verification

Files

.
..
data
mom_dsn
networks
synthesis
verification

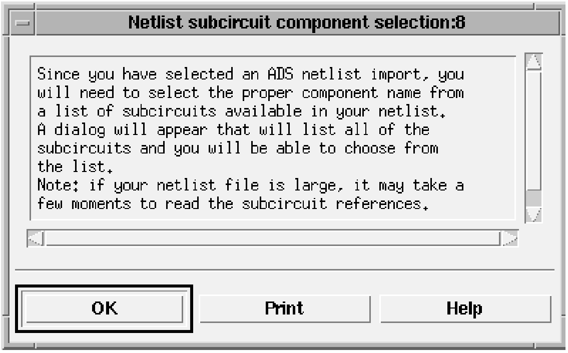
Selection

.eelec/Netlist_Example_prj/

- **Element Replacement Table** If you created a custom component in Advanced Design System that you want the system to use when it imports a netlist in place of the default ADS component, see "Using a Custom Component" on page 3-19 for instructions on completing this field.

Once all options are entered, click **OK**. You are returned to the *Import* dialog box.

Click **OK** on the *Import* dialog box. The following informational dialog appears:

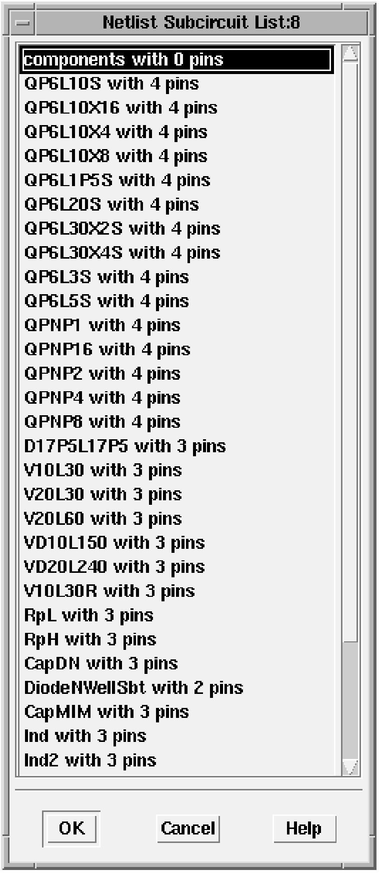


Netlist subcircuit component selection:8

Since you have selected an ADS netlist import, you
will need to select the proper component name from
a list of subcircuits available in your netlist.
A dialog will appear that will list all of the
subcircuits and you will be able to choose from
the list.
Note: if your netlist file is large, it may take a
few moments to read the subcircuit references.

The informational dialog informs you that you have selected an ADS netlist import and that you will need to select a subcircuit from your netlist to be the top level circuit that will used for the simulation.

Click **OK** on the informational dialog to select the subcircuit. The following dialog appears:



Netlist Subcircuit List:8

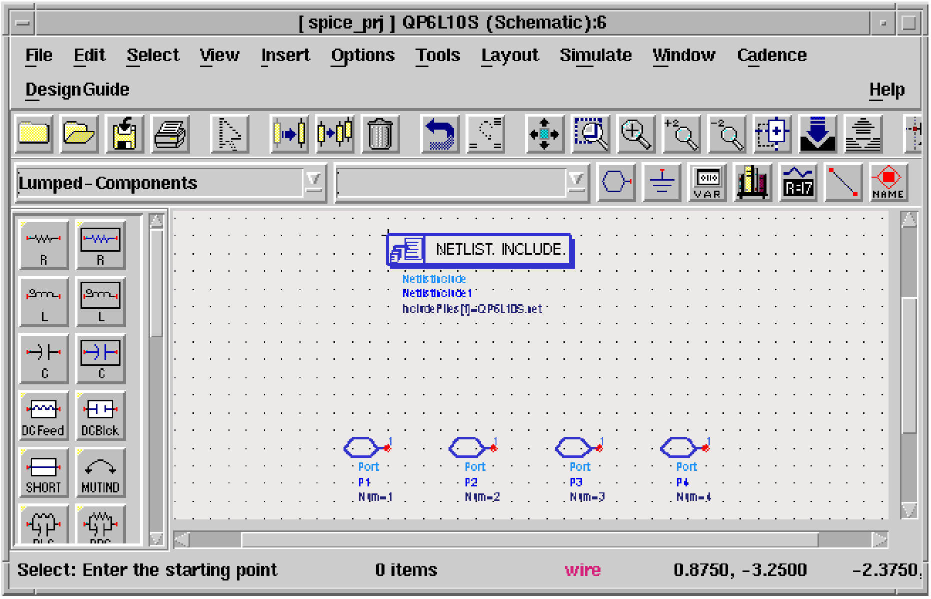
components with 0 pins
QP6L10S with 4 pins
QP6L10X16 with 4 pins
QP6L10X4 with 4 pins
QP6L10X8 with 4 pins
QP6L1P5S with 4 pins
QP6L20S with 4 pins
QP6L30X2S with 4 pins
QP6L30X4S with 4 pins
QP6L3S with 4 pins
QP6L5S with 4 pins
QPNP1 with 4 pins
QPNP16 with 4 pins
QPNP2 with 4 pins
QPNP4 with 4 pins
QPNP8 with 4 pins
D17P5L17P5 with 3 pins
V10L30 with 3 pins
V20L30 with 3 pins
V20L60 with 3 pins
VD10L150 with 3 pins
VD20L240 with 3 pins
V10L30R with 3 pins
RpL with 3 pins
RpH with 3 pins
CapDN with 3 pins
DiodeNWellSbt with 2 pins
CapMIM with 3 pins
Ind with 3 pins
Ind2 with 3 pins

All of the existing subcircuits with their pin counts are listed.

Note Only the subcircuits from the Spectre or SPICE netlist that you entered in the *Import* dialog are listed. If you want to use a subcircuit that is contained in an include file, you have to manually add that after the import process is complete.

Select the subcircuit to use as the top level subcircuit and click **OK** to start the translation.

The Netlist Translator performs the automatic translation of your source file into an ADS Netlist plus an ADS Schematic with a *NetlistInclude* component (see Figure 2-2).



NetlistInclude
NetlistInclude1
IncludeFiles[1]=QP6L10S.net

Port P1 Num=1
Port P2 Num=2
Port P3 Num=3
Port P4 Num=4

Select: Enter the starting point 0 items wire 0.8750, -3.2500 -2.3750

Figure 2-2. ADS Schematic Window with NetlistInclude Component.

Viewing and Editing the Netlist

If you selected *ADS Netlist* in “Choosing the Translated Output Format” on page 2-8, the Netlist Translator creates an ADS netlist and places it in the project directory. The ADS netlist is in the form *filename*.net, where *filename* is the component name you specified in the *Import SPICE Options* dialog box.

To view and edit the netlist, open the ADS text editor from the *Main* window, **Options** > **Text Editor.** You can also view the netlist from a command shell or with any ASCII text editor. Table 2-6 displays the translated ADS Netlist, *ex1.net*, of the example file, *ex1.scs*.

Table 2-6. Translated ADS Netlist (ex1.net)

```
; *SIMPLE RTL INVERTER
V_Source:vcc _node4 0 Vdc=5 Vac=0
V_Source:vin _node1 0 V_Tran=pulse(time,0,50,2e-09,2e-09,2e-09,3e-08) Vdc=0 Vac=0
R:rb _node1 _node2 R=10k
modq1:q1 _node3 _node2 0 Area=1 Region=1 Mode=1
R:rc _node3 _node4 R=1k
model modq1 BJT NPN=1 PNP=0 Bf=20 Tf=.1n Cjc=2p Rb=100 RbModel=1
SweepPlan:Plan1 Start=0 Stop=5 Step=0.1
DC:DC1 SweepPlan="Plan1" SweepVar="vin.Vdc" UseSweepPlan=yes
Tran:Tran1 StopTime=100n StartTime=0 MaxTimeStep=1n
```

Once you have your netlist displayed, verify that the translated netlist and the original netlist have the same parameters. For more information on parameter mapping in devices and models, refer to Chapter 7, Translating a Device and Chapter 8, Translating a Model.

Using Imported Designs in ADS

This section provides additional information that can help you prepare for simulating your imported designs in Advanced Design System. Depending on your specific needs and the way your translation was performed, you may be required to perform these steps before your design can be simulated. For information on simulating your design, refer to Chapter 4, Simulating the Translated Netlist.

Checking for Unconnected Nodes

If the log file reported any components that could not be translated, there will be open connectors which show up as red diamonds on the pins in the schematic. Pins that have a red diamond when the schematic is complete are not connected to any other pins. To check for unconnected nodes on a large schematic where the red diamonds are hard to see:

1. From the schematic window, choose **Options > Check Representation**. The *Check Representation* dialog box is displayed.

2. Ensure the *Unconnected pins* check box is active and then click **OK**. A *Check Representation Report* is displayed that will list all unconnected pins. Note that the components with unconnected pins are highlighted on the schematic.
3. In the *Check Representation Report*, click **Print** to save the report to a file. Click **OK** to clear the dialog.
4. To clear the highlighted components on the schematic, choose **Layout > Clear Highlighted > Components**.

If a pin is not connected, it may be because the component it was connected to was not included in the original file. More likely, the translation of that component failed because of a syntax error or there is no equivalent ADS component. In this case, you should manually place a component and connect them using the *Insert > Node Name* command.

A message will be written to the translation log file (spctoiff.log) in the case of any failures. For more information on the translation log, refer to "Viewing the Translation Log" on page 3-9.

Note There is no visible indication of unconnected nodes in the output ADS Netlist, so the log file must be checked for the names of un-translated components.

Including Models and Subcircuits

This section describes how to include models and subcircuits in your existing design using several example files. Before continuing, create the two example files shown in Table 2-7 and Table 2-8. Save each of the files in the project directory that was used for storing the example SPICE netlist (ex1.sp) in "Accessing the Import Dialog" on page 2-3.

The two new example SPICE files make up a simple transmission line inverter. The first SPICE netlist (tline.sp) is a subcircuit.

Table 2-7. Transmission Line Subcircuit (tline.sp) Netlist

```
*TRANSMISSION-LINE SUBCIRCUIT
.SUBCKT TLINE 1 3
T1 1 2 3 4 Z0=50 TD=1.5NS
T2 2 0 4 0 Z0=100 TD=1NS
.ENDS TLINE
.END
```

The next SPICE netlist (inverter.sp) includes additional components that complete the transmission line inverter design.

Table 2-8. Transmission Line Inverter (inverter.sp) Netlist

```
*TRANSMISSION-LINE INVERTER
V1 1 0 PULSE 0 1 0 0.1NS 0.1NS 20NS 40NS
R1 1 2 50
R2 4 0 50
```

Import both of your new SPICE files individually using the information in "Importing a Netlist File Using the User Interface" on page 2-1. These two files should be imported using the *PSpice* dialect and the *Named Connections* import method. Remember to close all designs before starting each import.

After the designs have been imported, leave the design *inverter.dsn* open. Notice the red diamonds on this schematic indicating that something is missing from the file. In some cases, this is caused by components that could not be translated (see "Checking for Unconnected Nodes" on page 2-17). In this case however, the original SPICE file did not supply this component so the *tline.dsn* subcircuit will be used to complete the circuit.

Including a Subcircuit

Your SPICE netlist may contain one or more subcircuit definitions. A subcircuit in the SPICE file becomes a reusable component in the ADS schematic. To use a subcircuit, you can place an instance of it in a new or existing design.

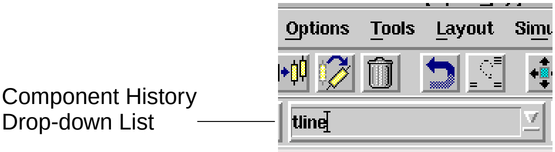
To browse for a component and include it in your design:

1. From the schematic window, choose **Insert > Component > Component Library** or click the *Display Component Library List* button on the toolbar.

   The *Component Library/Schematic* dialog box is displayed.

2. In the *Libraries* field, scroll down in the list until you find either the *Spice Netlists* library or the *Sub-networks* library.
3. Click either the *Spice Netlists* library or the *Sub-networks* library in the Libraries field to see the list of subcircuits that can be placed as components.
4. In the *Components* field, click the Component that you want to use (*tline* in the example case) and move the cursor back into the schematic window.
5. Click to place the component in your schematic.
6. Click the *End Command And Return to Select Mode* icon from the tool bar to stop placing instances of your component.

If you know the name of the subcircuit without looking it up, you may type it directly into the *Component History* drop-down list box as illustrated below.



When you enter the name of the subcircuit in the component history box, the case must match exactly with the design name in ADS. Your subcircuit name will be all lower case unless you checked the Suppress name mapping option in the Import Netlist Options dialog box. For more information on capitalization and name mapping, refer to "Understanding Capitalization" on page 2-31 and "Suppress name mapping:" on page 2-7.

Note If your SPICE netlist contains a set of models outside of any subcircuit block, the models will be placed in a separate top level design that has the same name as the SPICE file. To use these models, copy them into the design that references them. Optionally, a design with models only can be placed as a subcircuit within another design, but all of the models have to be edited to have global scope. To do this, select a model, open the edit parameter dialog, click the **Component Options** button and change *Scope* to *Global*.

Example Spectre Subcircuit with Referenced Parameter Value

The following example illustrates a translation of a Spectre subcircuit into an ADS netlist. The Spectre subcircuit contains two parameters with default values. The value of the second parameter, *r2*, refers to the first parameter, *r1*. (The value of *r1* could either be the default value or a specified value when the subcircuit is called from an instance component.)

ADS does not directly support parameter default values that reference values of other parameters passed into the subcircuit. The Netlist Translator handles this by placing extra variables (in this example, ADS_VAR_1, ADS_VAR_2, and UNDEF) and an equation(s) within the netlist to calculate the correct value of the contingent parameter (in this example, *r2*).

Note This is only available using the command line use the *-pp* (preserve parameter) option.

Spectre subcircuit

```
subckt myexample (t1 t2 s)
parameters r1=1 r2=r1
R0 (t1 net1) resistor r=r1
R1 (net1 t2) resistor r=r2
ends myexample
```

ADS subcircuit-translated

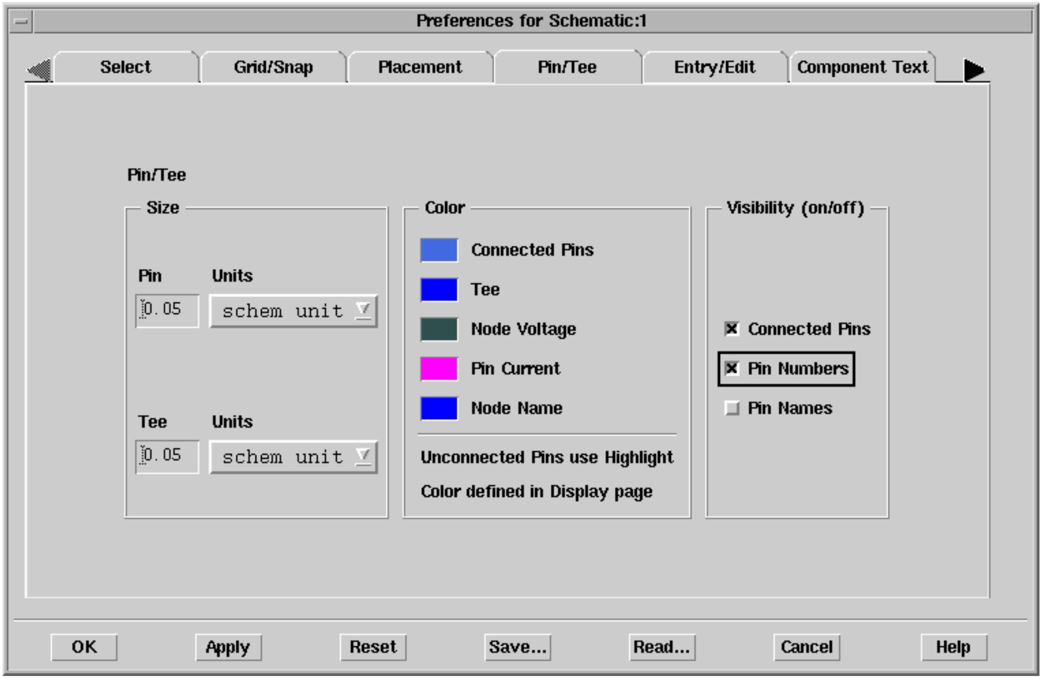
```
define myexample (t1 t2 s)
parameters r1=1 r2=”UNDEF”
ADS_VAR_0=r1
ADS_VAR_1=if (r2 == ”UNDEF”) then ADS_VAR_0 else r2 endif
R:R0 t1 net1 R=r1
R:R1 net1 t2 R=ADS_VAR_1
end myexample
```

Connecting a Component

This section describes how to set pin number preferences and assign node names in order to connect a subcircuit to a design. After placing the instance of the Transmission-Line Inverter subcircuit in the last example, you can use the ADS *Node Names* to connect the subcircuit pins to the rest of the circuit. You may need to turn the pin number visibility on to help identify the proper pins when naming nodes.

To set the *Pin Numbers* schematic preference in order to see the pin numbers:

1. From the schematic window, choose **Options > Preferences**. The *Preferences for Schematic* dialog box is displayed.



Pin/Tee

Size

Pin: 0.05 Units: schem unit

Tee: 0.05 Units: schem unit

Color

🟦 Connected Pins

🟦 Tee

⬛ Node Voltage

🟪 Pin Current

🟦 Node Name

Unconnected Pins use Highlight Color defined in Display page

Visibility (on/off)

☒ Connected Pins

☒ Pin Numbers

□ Pin Names

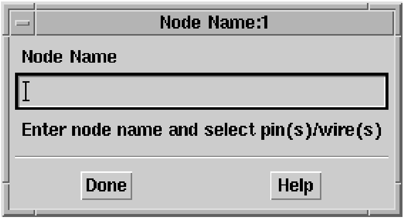
2. Select the **Pin/Tee** tab.
3. Click the *Pin Numbers* check box in the *Visibility (on/off)* section to activate the pin number visibility.
4. Click **OK** to save the settings and close the dialog box.

Defining the Node Names

To define the node names to connect the subcircuit:

1. Choose the **Insert > Node Name** menu selection or click the *Node Name* icon in the tool bar.

   The Node Name dialog box appears.



Enter node name and select pin(s)/wire(s)

2. Enter *_node2* in the Node Name field but *DO NOT* click the **Done** button. An instance of _node2 is attached to your cursor.

3. Drag your cursor onto the schematic and click pin 1 of the subcircuit symbol.

4. In the Node Name dialog box, change the node name to *_node4*. Again, *DO NOT* click the **Done** button. Drag your cursor onto the schematic and click pin 2 of the subcircuit symbol.

5. Save your new design as *tline_inv.dsn*.

This completes the circuit. Your combined *inverter.dsn* and *tline.dsn* should now appear similar to the *tline_inv.dsn* schematic design shown in Figure 2-3. This design is used in the simulation example in Chapter 4, Simulating the Translated Netlist.

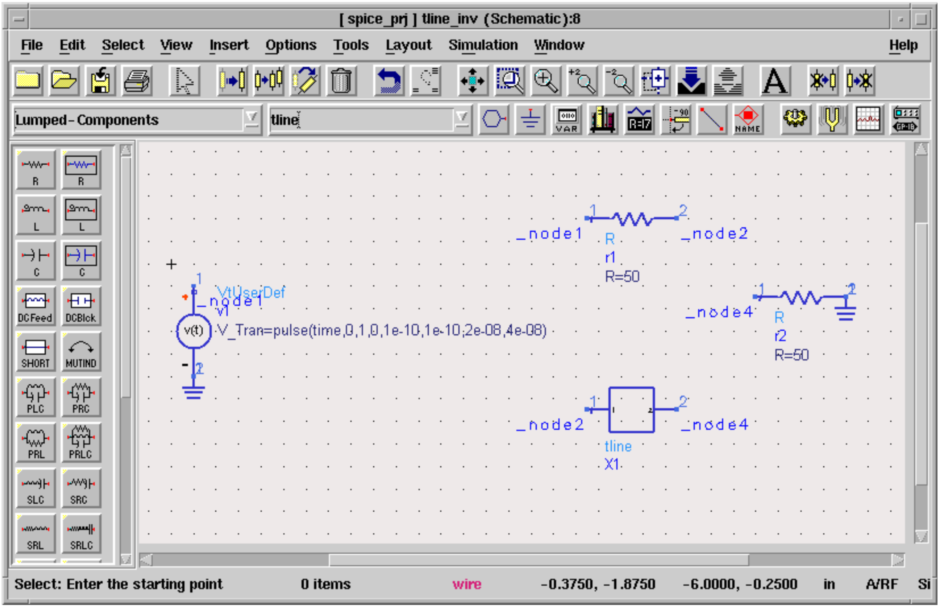


Figure 2-3. Translated Schematic with Subcircuit (*tline_inv.dsn*)

Creating a Hierarchical Project

You can create hierarchical projects using the *Include/Remove Projects* command. This command creates a reference, or link to another project.

If the import has produced a set of models or subcircuits that will be maintained in a project directory other than the one a designer will be working in, that project directory should be *included* into the working project directory so the simulator can see the imported items. The *Include & Remove* dialog can be accessed from the *Main* window. To include a project directory:

1. Open the working project directory from which you want to reference designs in other projects.
2. From the *Main* window, choose **File > Include/Remove Projects**. The *Include & Remove* dialog appears.
3. Use the *File Browser* to locate and select the project you want to include. This would be the project containing the set of models or subcircuits.

4. Click the **Include** button. The project is added to the *Project Hierarchy* listing.

5. Repeat as needed, then click **OK**.

Using the NetlistInclude Component

If you selected *ADS Netlist* in "Choosing the Translated Output Format" on page 2-8, your translated schematic will contain a *NetlistInclude* component. This section describes how you can view and modify the NetlistInclude component and then use the component within another design.

The NetlistInclude component in this section is created by modifying and translating a previous example SPICE Netlist into an ADS Netlist with a NetlistInclude component. The newly translated design is then used to show how you can place an instance of your NetlistInclude component into another design and then simulate the design.

To setup and translate your example SPICE Netlist into an ADS Netlist with a NetlistInclude component:

1. Close any open designs. Choose **File > Close All**.

2. If it's not already opened, open your project that you created in "Importing a Netlist File Using the User Interface" on page 2-1. Choose **File > Open Project**.

3. From the ADS Main window, choose **Options > Text Editor** to open the ADS text editor.

4. In the text editor, choose **File > Open** and open the file *tline.sp* that you created in Table 2-7 in "Including Models and Subcircuits" on page 2-18.

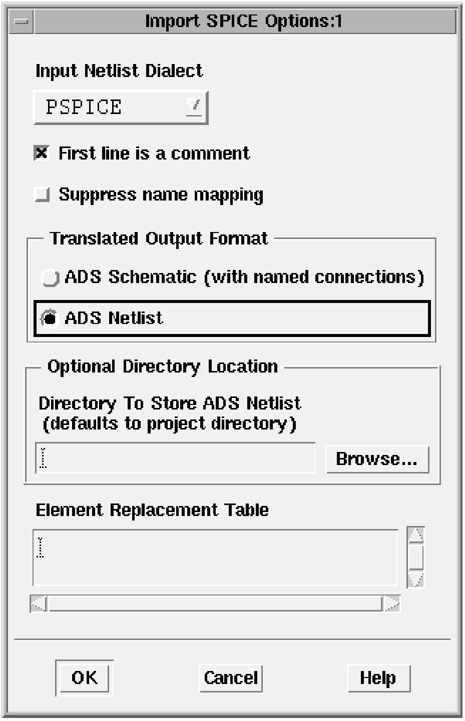
5. Choose **File > Save As** to save the file with a new name, *tline2.sp*.

6. Edit the new file to change the subckt name to TLINE2 as shown in Table 2-9, then re-save and close the new file.

Table 2-9. Transmission Line Subcircuit (tline2.sp) Netlist

```
*TRANSMISSION-LINE SUBCIRCUIT
.SUBCKT TLINE2 1 3
T1 1 2 3 4 Z0=50 TD=1.5NS
T2 2 0 4 0 Z0=100 TD=1NS
.ENDS TLINE
.END
```

7. From a blank schematic window, import the new file *tline2.sp*. In the *Import Netlist Options* dialog box, select *PSPICE* as the Spice Dialect and *ADS Netlist* as the Connection Method For Schematic. If you need more information on importing a SPICE file, refer to "Importing a Netlist File Using the User Interface" on page 2-1.



Import SPICE Options:1

Input Netlist Dialect: PSPICE

☒ First line is a comment

□ Suppress name mapping

Translated Output Format

○ ADS Schematic (with named connections)

● ADS Netlist

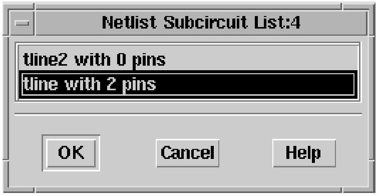
Optional Directory Location

Directory To Store ADS Netlist (defaults to project directory)

Element Replacement Table

8. Click **OK** to exit the Import Netlist Options dialog box.

9. In the Import dialog box, click **OK** to translate the netlist. An information dialog appears that informs you to select a subcircuit that exists within the netlist.



Netlist Subcircuit List:4

tline2 with 0 pins

tline with 2 pins

10. Select the *tline with 2 pins* from the Netlist Subcircuit List dialog box and then click **OK**.

    Once your translation is complete, ADS displays a schematic window with the *tline2.dsn* that contains two ports and a NetlistInclude component similar to the one in Figure 2-4.

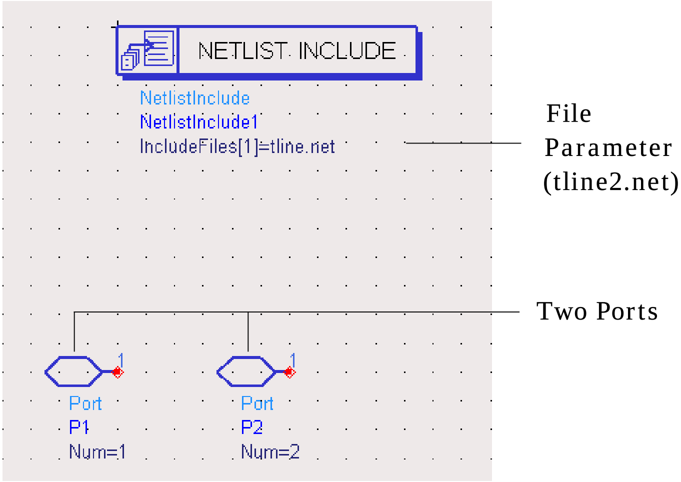


Figure 2-4. Design Containing Two Ports and a NetlistInclude Component

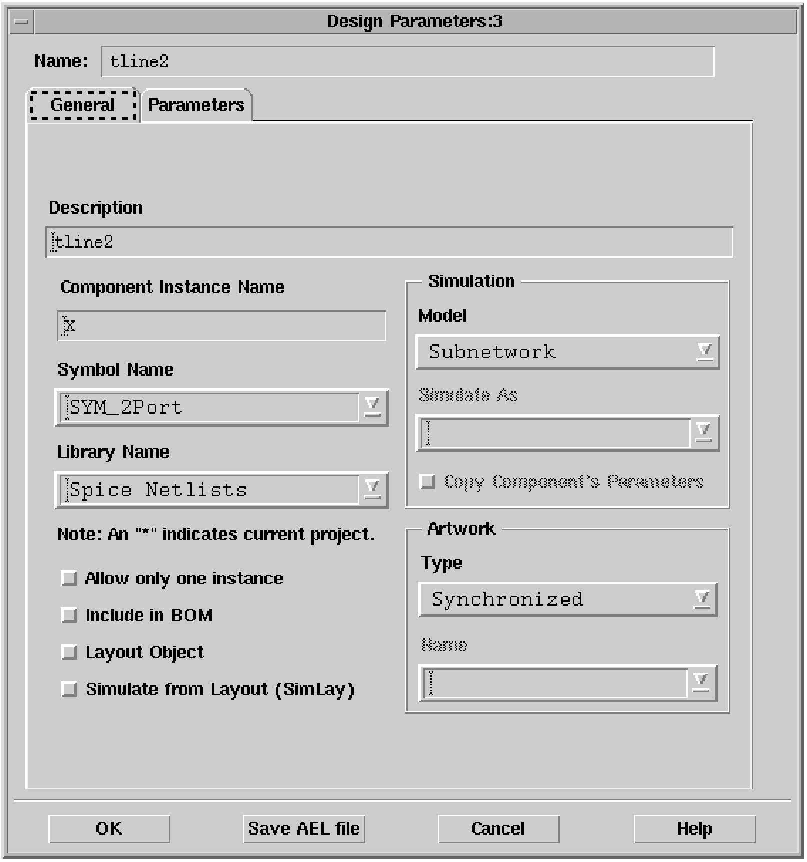
11. Note that the *File* parameter in the NetlistInclude component is called *tline2.net*. The translated file *tline2.net* is now in the project directory and looks like the netlist in Table 2-10. You can use the *ADS text editor* to open and view the ADS Netlist.

Table 2-10. Translated Transmission Line Subcircuit (*tline2.net*) Netlist

```
; *TRANSMISSION-LINE SUBCIRCUIT

define tline2 ( _node1 _node3)
TLIN4:t1 _node1 _node2 _node3 _node4 Z=50 E=(1.5e-09)*360 F=1
TLIN4:t2 _node2 0 _node4 0 Z=100 E=(1e-09)*360 F=1
end tline
```

To view the design details of your NetlistInclude component:

1. From the schematic window, choose **File > Design Parameters.** The *Design Parameters* dialog box appears. The *General* tab is displayed by default.



Name: tline2

Description: tline2

Component Instance Name: x

Symbol Name: SYM_2Port

Library Name: Spice Netlists

Note: An "*" indicates current project.

□ Allow only one instance

□ Include in BOM

□ Layout Object

□ Simulate from Layout (SimLay)

Simulation

Model: Subnetwork

Simulate As:

□ Copy Component's Parameters

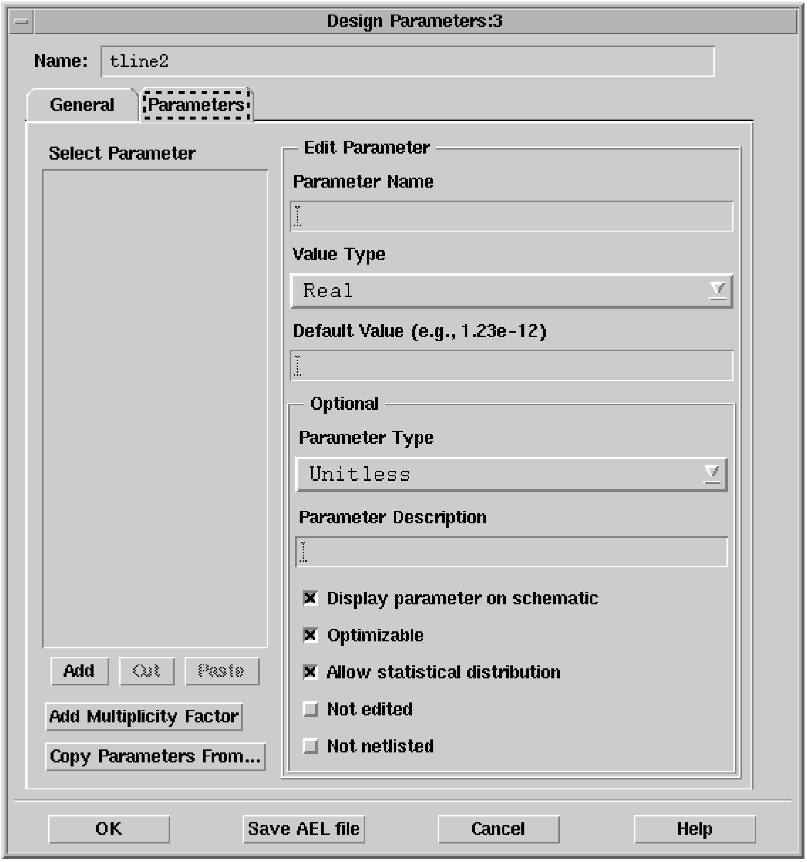
Artwork

Type: Synchronized

Name:

- The *Name* and *Description* fields display the current design name by default.
- The *Component Instance Name* field default is X. The text in this field is used as a prefix in building a unique name (ID) for every item. This prefix becomes part of the annotation displayed with the symbol representing the parametric subnetwork when you place it in a design.
- The *Symbol Name* field displays the standard symbol with the number of ports that you specified for the circuit in the *Import Netlist Options* dialog.
- The *Library Name* field displays the name of the library that the design is owned by. This might be something like *Spice Netlists*. For more information, refer to "Including Models and Subcircuits" on page 2-18.

2. Click the **Parameters** tab. The Parameters tab does not apply to this particular example; however, you may find it helpful to understand the function of this dialog box for future reference.



Design Parameters:3

Name: tline2

Select Parameter

Edit Parameter

Parameter Name

Value Type: Real

Default Value (e.g., 1.23e-12)

Optional

Parameter Type: Unitless

Parameter Description

☒ Display parameter on schematic

☒ Optimizable

☒ Allow statistical distribution

□ Not edited

□ Not netlisted

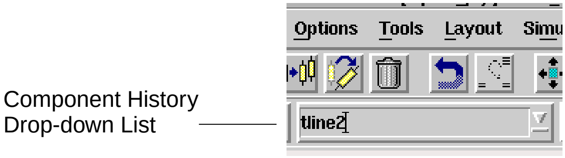
When you import a SPICE file containing a subcircuit with parameters to an ADS schematic, you can use this tab to view or edit the parameter details. Each parameter is listed in the *Select Parameter* field and has characteristics that determine how it is handled when the network is reused. These include the *Parameter Name*, the *Value Type* assigned to the parameter, the *Default Value*, and optional control attributes. Default values are listed only if they were specified in the SPICE file.

For more information on subcircuit parameters, refer to ".SUBCKT" on page 10-18. For more information on using the Design Parameters dialog, refer to "Defining Parameters" in the ADS User's Guide.

3. Click **OK** to continue.

To add an instance of your imported design:

1. Open the schematic *tline_inv.dsn* that was created as a result of "Including Models and Subcircuits" on page 2-18 and shown in Figure 2-3.
2. Save the tline_inv file as *tline_inv2.dsn* using the *File* > *Save As* command.
3. In the tline2_inv schematic design, delete the *tline* component.
4. In this example, *tline2* is the imported design with the *NetlistInclude* component. An instance of this component can be placed in your tline_inv2 design by entering the name *tline2* in the component history box. This attaches an instance of the component to your cursor that you can now click into place. Place the instance of tline2 in an appropriate location on the tline_inv2 schematic.



5. Connect the appropriate circuitry to the new subcircuit by naming pin 1 *_node2* and pin 2 *_node4*. For more information on naming nodes, refer to "Defining the Node Names" on page 2-22.
6. This completes the circuit. This circuit can now be simulated using the example shown in Chapter 4, Simulating the Translated Netlist.
7. ADVANCED OPTIONS

   You may create your own symbol by drawing it in the symbol view (**View > Create/Edit Schematic Symbol**) and changing the Symbol Name in the *Design Parameters* dialog to the name of the design, *tline2* in this example. Save the AEL file from the *Design Parameters* dialog and the design file from the File menu. A new symbol will not automatically appear in the design in which it is used. For the new symbol to appear in your design, replace the instance of tline2 in your test circuit.

For additional information about the NetlistInclude component, refer to "Adding the NetlistInclude Component" on page 3-14.

Understanding Import Details

This section describes some of the unique details of the translation process that should be understood before attempting to translate a design.

Using Valid ADS Characters

The legal character set for Advanced Design System names is alphanumeric _ + - = ^ ' @ # & $ %. Legal characters other than alphanumeric and underscore require special handling. Any string using special characters in ADS must be enclosed by quotes. This is handled by the translator.

Note Do not insert extra quotes in your file. Additionally, $ and % are not allowed in subcircuit names since these become invalid design and file names.

If the translator encounters an illegal character (any character not listed above), the illegal character is replaced with an underscore and a warning message is written to the log file.

Understanding Capitalization

Like Spectre, ADS is a case sensitive system, so *Abc* and *ABC* represent two different designs, components, etc. If the *Suppress name mapping* option is *not* selected in the *Import SPICE Options* dialog, the translator converts all node names, element and model names, variable names and subcircuit names to lower case. Conversion to all lower case can cause a problem if the design being imported references an existing ADS design that has a mixed case name or a name that is all capitals. In addition, this can case a problem if two or more items are referenced within the Spectre netlist use case to differentiate between them, for example, two components named *RS* and *rs*.

Refer to "Suppress name mapping:" on page 2-7 for information on toggling the *Suppress name mapping* option. For more information on the command line option, refer to "Importing a File from the Command Line" on page 3-3. If this option is used, the names in the file must use consistent capitalization; for example, every time *Abc* is used, it must be capitalized the same.

Using Unique Names

The Spectre and SPICE simulators are less restrictive than ADS with regards to unique names. In ADS2001, model names must be different than device names and the translator modifies the device name to help provide uniqueness.

The only other restriction is that a subcircuit cannot override the name of a built-in component. The translator does not check for this at this time, since the library of parts is always growing and the translator is run as a separate process from ADS.

- On UNIX, this should not be an issue because component names in ADS typically begin with a capital letter, and any subcircuit name that is output by the translator will be in lower case. The exception would be if the *Suppress name mapping* option were turned ON. For more information on the Suppress name mapping option, refer to "Suppress name mapping:" on page 2-7.
- On a PC, this may present a problem in ADS because design names follow a case-insensitive rule (for this one particular case). Because design names become file names and the file system is case-insensitive, a subcircuit could potentially attempt to override a built-in component.

If there is a situation where the name of the subcircuit is the same as the name of an existing ADS component, the name of the subcircuit will be changed in the schematic created by the translator. This will not be the case if the netlist option is used, which will lead to an error during simulation. The user is responsible for unique names in this situation.

Using Global Nodes

Global nodes are node names which when used at any level of the hierarchy in the circuit, will be connected to the same node. In Spectre, you declare a global node in the following manner in a definition statement:

global <*ground*> VCC

Berkeley SPICE does not support global nodes and HSpice and PSpice have different implementations. In HSpice, you declare a global node this way in a definition statement:

.global VCC

In PSpice, it is just defined when used by putting $G_ in front of the node name:

```
$G_VCC
```

The *global* statement is supported in ADS netlists but has scope limitations and might not be recognized in a hierarchical design if defined in the wrong subcircuit. Therefore, the translator uses the optional method of defining a global node (which is similar to the PSpice method). Instead of special characters at the beginning, an exclamation point is appended to the node name when it is used.

In the example above, the node name becomes *VCC!*. Additionally, this must be quoted since the exclamation point is a special character to ADS, so it becomes *"VCC!"*. No GLOBALNODE component will be placed for schematic and no globalnode statement will appear in the netlist.

Chapter 3: Advanced Methods for Importing Files

This chapter describes the procedures for Importing files into Advanced Design System using the Netlist Translator from the command line. The methods described in this chapter are for advanced users who want to have the ability to manipulate their netlist within a translation.

Performing your translation via the command line provides you with more flexibility in the process and more control over the output. Some of the advantages of translating your Spectre file from the command line are listed below:

- It enables you to check for errors midway through the translation.
- It enables you to modify the intermediate file before the schematic is created
- It enables you to do batch processing with multiple files.
- It gives you access to advanced command line options not supported by the user interface.

The output of a Netlist Translator command line translation can be either an IFF file that is used to create an *ADS Schematic* or an *ADS Netlist*.

Understanding the Import Operation

This section describes how the Netlist Translator software works when performing an import operation. It may be helpful to understand these steps if you plan on importing your designs using the Netlist Translator from the command line.

When the Netlist Translator performs an import, the software runs an executable program called *nettrans*. The *nettrans* program reads the file and then writes either an IFF file, *spice. iff*, or an ADS netlist, *spice.net*, depending on the connection method defined when setting the import options. If an IFF file is created, Advanced Design System reads the IFF file and creates a schematic. If an ADS netlist is created, it is stored in ADS netlist format and placed in a pre-specified directory. Any errors that occur during the translation are written to a file called *spctoiff.log*. This file records any components that could not be translated due to issues such as unrecognized syntax or incompatible models. Running the *nettrans* program from the command line enables you to modify the contents of the *spice.iff* file before importing the file into Advanced Design System. Once you are comfortable with the output of the *nettrans* program, you can simply perform an IFF import into ADS. The IFF importer loads the *spice.iff* file, generates the schematic and writes any error messages or warnings to a log file called *ifftolib.log*. Alternatively, you could convert the output of the *nettrans* program to an ADS Netlist.

Running the *nettrans* program from the command line enables you to modify the contents of the *spice.iff* file before importing the file into Advanced Design System. Once you are comfortable with the output of the nettrans program, you can simply perform an IFF import into ADS. The IFF importer loads the *spice.iff* file, generates the schematic and writes any error messages or warnings to a log file called *ifftolib.log*. Alternatively, you could convert the output of the nettrans program to an ADS Netlist. For information on how the import options are controlled from the command line, refer to "Executing the Nettrans Command" on page 3-5.

For information on how the import options are controlled from the user interface, refer to "Setting the Import Options" on page 2-5.

For information on how the import options are controlled from the command line, refer to "Executing the Nettrans Command" on page 3-5.

Figure 3-1 shows the different import options available when using the Netlist Translator. Depending on your needs, you can choose between the various options for optimum translation results.

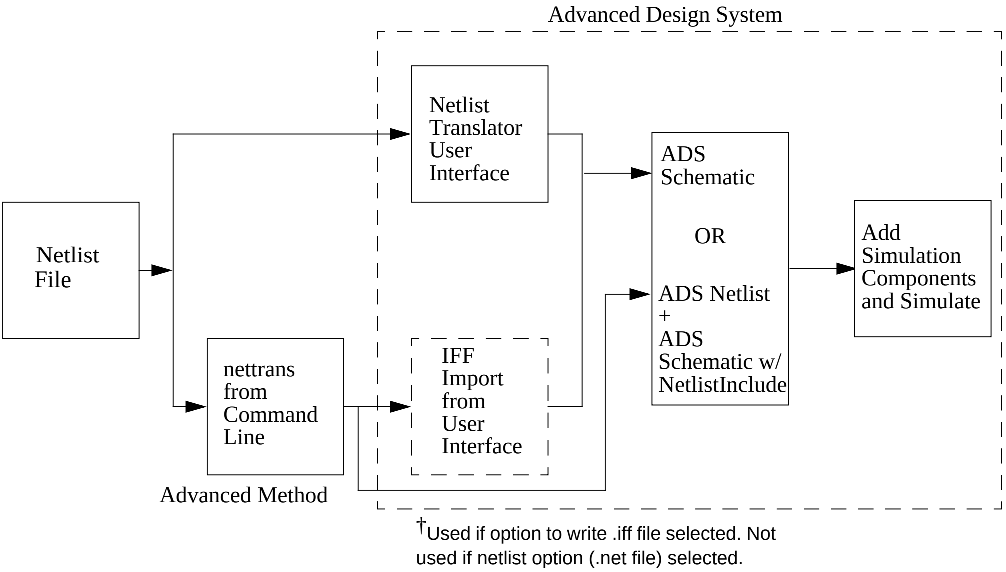

Advanced Design System
Netlist File
Netlist Translator User Interface
ADS Schematic
OR
ADS Netlist + ADS Schematic w/ NetlistInclude
Add Simulation Components and Simulate
nettrans from Command Line
IFF Import from User Interface
Advanced Method
†Used if option to write .iff file selected. Not used if netlist option (.net file) selected.

Figure 3-1. Import Options

With all methods of translation, you should compare your results against your original netlist if the size of the file permits. Once you are comfortable with your translation into ADS, place the appropriate stimulus and simulator control elements and simulate your design as described in Chapter 4, Simulating the Translated Netlist.

Importing a File from the Command Line

This section describes the details of setting up and operating the Netlist Translator from the command line.

Note It is recommended that only advanced users attempt to import files from the command line since it requires knowledge of setting environment variables and working in a UNIX shell or DOS window.

Setting up the Nettrans Command

To set up the *nettrans* program for use from the command line:

For Windows platforms:

1. Open an MSDOS shell.
2. Set your $HPEESOF_DIR environment variable to your ADS installation directory. For example,

   ```
   set HPEESOF_DIR=<ADS_install_dir>
   ```

3. Set your PATH environment variable to include the $HPEESOF_DIR\bin directory. For example,

   ```
   set path=C:\AdvDesSys2001\bin;%path%
   ```

4. Set the appropriate library path for your operating system. For example,

   ```
   set shlib_path=$shlib_path:$HPEESOF_DIR\lib\win
   ```

For UNIX platforms:

1. Set your $HPEESOF_DIR environment variable to your ADS installation directory. For example, if using the Korn shell enter,

   ```
   export HPEESOF_DIR=<ADS_install_dir>
   ```

2. Set your PATH environment variable to include the $HPEESOF_DIR/bin directory. For example, if using the Korn shell enter,

   ```
   export PATH=$HPEESOF_DIR/bin:$PATH
   ```

3. Set the appropriate library path for your operating system.

   For **HPUX** operating systems (i.e. hpux10 or hpux11), enter the following:

   ```
   SHLIB_PATH=$SHLIB_PATH:$HPEESOF_DIR/lib/hpux10
   ```

   OR

   ```
   SHLIB_PATH=$SHLIB_PATH:$HPEESOF_DIR/lib/hpux11
   ```

   For **SUN** operating systems (i.e. sun4 or sun55), enter the following:

   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HPEESOF_DIR/lib/sun4
   ```

   OR

   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HPEESOF_DIR/lib/sun55
   ```

For **AIX** operating systems (i.e.aix4), enter the following:

```
LIBPATH=$LIBPATH:$HPEESOF_DIR/lib/aix4
```

Executing the Nettrans Command

The *nettrans* command uses the following general syntax:

```
nettrans input_filename output_filename -{s2|s3|p|h|pl} -{g|u|w} [-l] [-n]
```

For Spectre use the *pl* parameter followed by *spectre*. The following is command syntax specific to Spectre:

```
nettrans input_filename output_filename -pl spectre -{g|u|w} [-l] [-n]
```

Simply entering *nettrans* at the command line with no parameters displays a detailed *nettrans* usage message. Entering the *nettrans -v* command displays the translator version, while entering *nettrans -adv* displays the syntax for the advanced options. Table 3-1 displays a listing of all parameters and definitions used by the *nettrans* command.

Table 3-1. Parameter Definitions for *nettrans*

Parameter	**Definition**
input_filename	The name of the SPICE or Spectre source file to read in.
output_filename	The name of the IFF or netlist file to write.
Input file type	**Definition**
-s2	Berkeley SPICE (2g6)
-s3	Berkeley SPICE (3c, 3e, 3f)
-p	PSpice
-h	HSpice
-pl <*dialect*>	Generic (use for translating Spectre netlists, *-pl spectre*)
Output options	**Definition**
-g	Create ADS netlist
-u	Create schematic w/o wires (uses named connections)
-w	Create schematic w/wires (not recommended on large circuits) (Available from command line only.)
Other options	**Definition**
-l	First line is *not* a comment (used for library files or model include files)

Table 3-1. Parameter Definitions for *nettrans*

-n	Do not perform any name mapping for conflict avoidance
-v	Return the version of the Netlist Translator
-adv	View advanced options
Advanced options	**Definition (available from command line only)**
-w	Create schematic w/wires (not recommended on large circuits)
-k	Create subcircuits for models at top level (used for library files or model include files)
-pp	Preserve Spectre subcircuit parameters.
-ps	Prefix string to be added to appropriate variables, model names, and subcircuit names in an ADS netlist. This is required to support a unique name space for design kit creation.
-sc	Suppress comments
-se	Suppress ADS subcircuit equations from being moved from the parameter list when translating Spectre netlists.
-si <*include_file*>	Suppress Spectre components contained in *include_file*. The full path must be specified.
-sw	Suppress warnings to log file.
-wrap <*int*>	Line length at which to perform line wrapping (must be greater than 10, default=80).
-m <*mapfile*>	Mapping file for user-defined components
-models	Read HSpice Model Library and output one file per .LIB/.END pair. Suppresses nested .LIB statements. output_filename should be a file extension only. ex: nettrans mix025.l net -h -g -models -pl dialect: Pass in the dialect to be parsed by Perl -u2: Create schematic w/o wires (uses named connections) Named Connections are read from log file so custom symbols can be used. If connections are not made, load ael file spcUtil.ael and run ael macro spctoiff_connect_pins("wireLabels.log");
-u2	Create schematic w/o wires (use named connections). Named Connections are read from log file so custom symbols can be used. If connections are not made, load ael file *spcutil.ael* and run ael macro *spctoiff_connect_pins("wireLabels.log")*. The *spcutil.ael* file can be found under *$HPEESOF_DIR/links/spice/spcutil.ael*.

Enter the appropriate *nettrans* command from the command line to translate your Spectre file. Once the *nettrans* command is complete you can view the output.

Additional Options Information

Using the -n Option

If the -n option is used, names are not mapped to the lower case. If case mixing was used in the file, the variable, instance and model names will not be converted to lower case. For more information on case sensitivity and name mapping, refer to "Understanding Capitalization" on page 2-31.

Using the -pp Option

ADS does not directly support parameter default values that reference values of other parameters passed into the subcircuit. This option will introduce logic in the translated netlist to manage reference values. See "Example Spectre Subcircuit with Referenced Parameter Value" on page 2-21 for an example using the -pp option.

Using the -se Option

Without this option, any Spectre parameters that reference a previous parameter in the subcircuit list will be pulled out of the subcircuit list and treated as an ordinary equation in the subcircuit. This option allows you to keep these parameters in place. Some manually editing is required to allow the netlist to simulate without errors.

Using the -si Option

The -*si* <*include_file*> option will translate a Spectre netlist with only a portion of the netlist outputted.

The information in the include file is used by the translator for model type resolution and other post processing needs. The contents of that file will not be output in the translated ADS netlist.

Using the -models Option

An HSpice library or model file can consist of hundreds of models, defined within a .LIB section. Each .LIB section defines one *model*, which may consist of a complete subcircuit. If this library file is referenced by a SPICE circuit file, only the libraries that are used will be translated. However, the user may want to translate the complete model library to a set of ADS netlist files. The -models option is available for this purpose.

To translate the complete model library in a file called mix025.l, invoke the translator from a shell command line as shown in the following example.

nettrans mix025.l net -h -g -models

The HSpice file is read and, for every .LIB section found in the library file, one ADS netlist file is output. The name of the library which is specified in the .LIB statement becomes the name of the ADS netlist file. An extension is given on the command line when the translator is invoked, and this is appended to the file name. In this case, the file extension is *net*. A library designated by *.LIB resistor* will be translated to an ADS netlist file called *resistor.net*.

Using the -u2 Option

When the -u option is used, connections in a schematic are made by naming nodes. This is done by placing a node name on each pin. The translator makes these connections by reading a configuration file that contains pin locations for all the element symbols that the translator uses. Subcircuit symbols are plain boxes with the appropriate number of leads and pins on them.

If you have defined a custom symbol for any element, and the pins are not in the location expected by the translator, some of the pins may not be named. Unconnected pins show up as red diamonds in the schematic. This will most likely happen when a custom library part exists in ADS and is referred to in a file. The translator does not create the subcircuit but is expected to connect to it.

When the correct pin information is not available to the translator in the spctoiff.cfg file perform the following steps:

1. Use the -u2 option from the command line as shown in the following example.

   nettrans *source_file*.sp *output_file*.iff -h -u2

   The translator will output the IFF file *output_ file*.iff and another file named wireLabels.log.

2. Import the IFF file from the user interface (refer to, “Importing an IFF File” on page 3-10).

   This will create a schematic with no connections. To complete the connections, make sure the component placement is satisfactory. There should be sufficient room between each component to display the parameters and the node names.

3. Save all designs in which you made changes.

4. Make sure the file wireLabels.log is in your project directory and enter the following commands on the AEL command line:

   ```
   load(strcat (HPEESOF_DIR, "links/spice/spcutil.ael"));
   ```

```
spctoiff_connect_pins("wireLabels.log")
```

When the macro has finished running, the pins will all be labeled with node names and all connections will be made as specified in the Spectre file.

Checking the Netlist Translator Version Number

To determine the version of the Netlist Translator that you are running perform the following steps:

1. Open a UNIX or DOS shell.
2. Ensure your environment variables are set correctly. For more information on setting environment variables, refer to "Setting up the Nettrans Command" on page 3-4.
3. Enter the following command:

   ```
   nettrans -v
   ```

   The Netlist Translator displays the version number.

   ```
   Netlist Translator, version <version_number>
   ```

   Where <*version_number*> is the actual version number of the translator.

   Note ADS Netlist Translator version 170.*xxx* or higher is required to translate Spectre files.

Viewing the Translation Log

The Spectre or SPICE to IFF translation log (see Table 3-2) contains important information about the netlist translation, including any error or warning messages recorded during the import operation. The translation log can be viewed using any ASCII text editor. To view the translation log file, open your text editor and load *spctoiff.log*.

If your log file displays any error or warning messages generated by the translator, refer to Chapter 6, Troubleshooting. Using a combination of the information in the log file and the information provided in the troubleshooting section, edit your translated netlist and save the new netlist with your corrections.

Since the translation log is an ASCII based text file, you can easily save the file with a new name using the **File > Save As** command for future reference.

Table 3-2. Example Translation Log File (spctoiff.log)

```
SPICE to IFF translation log

Input format: PSpice
Input filename: ex1.sp

Output format: ADS Netlist file
Output filename: ex1.net

Special options:
   Processing first line as comment.

Begin translation at Thu Aug 31 10:16:00 2000

Creating netlist.
Reading item definition file "spctoiff.cfg"

Translation completed at Thu Aug 31 10:16:03 2000.
```

Note The netlist to IFF translation log file (spctoiff.log) is over-written each time a new translation is performed. To avoid over-writing the log file, save it with a new file name.

Importing an IFF File

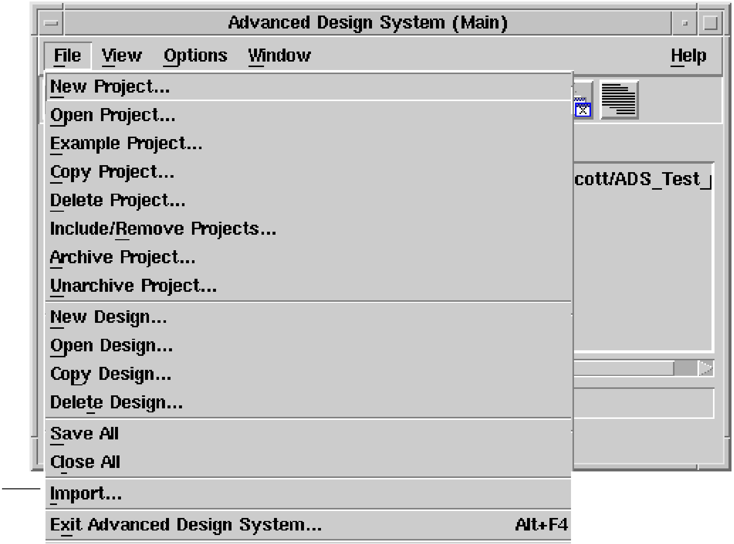
This section describes how to import Intermediate File Format (IFF) files, created by the *nettrans* command, into Advanced Design System. The IFF importer loads the *spice.iff* file, generates the new ADS Schematic and writes any error messages or warnings to a log file called *ifftolib.log*.

The Intermediate File Format (IFF) is an ASCII file that contains a simple, line-oriented command structure with a fairly rich set of constructs. The format is machine- and application-independent, thus simplifying design data transfer. For more information on Importing an IFF file, refer to “*IFF Files*” in the ADS *Importing and Exporting Designs* manual.

Accessing the Import Dialog

To import your IFF file,

Choose **File** > **Import** from the *Main* window.



File Import menu selection

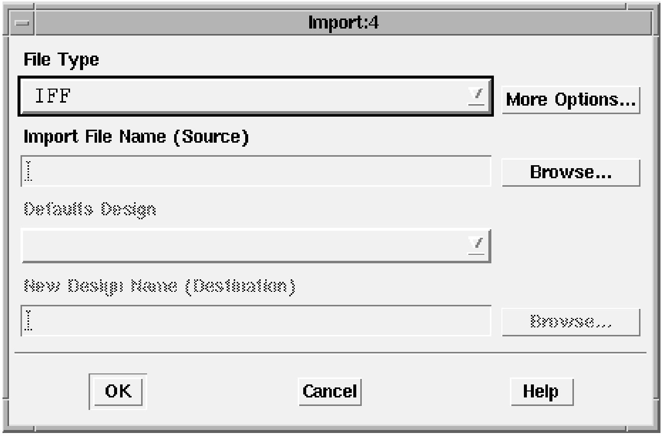
The *Import* dialog box appears.

Specifying the File Name

In the *Import* dialog box, choose the type of file to import, specify the filename, and supply other basic information needed by the translator.

1. In the *Import* dialog box that appears, select *IFF* from the *File Type* drop-down list if it isn't already displayed.
2. To specify the path and filename of the file you want to import click **Browse**. The *Import File Selection* dialog appears.
3. Double-click as needed to locate the directory containing your IFF file or enter the full path and file name in the *Selection* field. By default, all files are listed that have the suffix appropriate for the chosen file format.

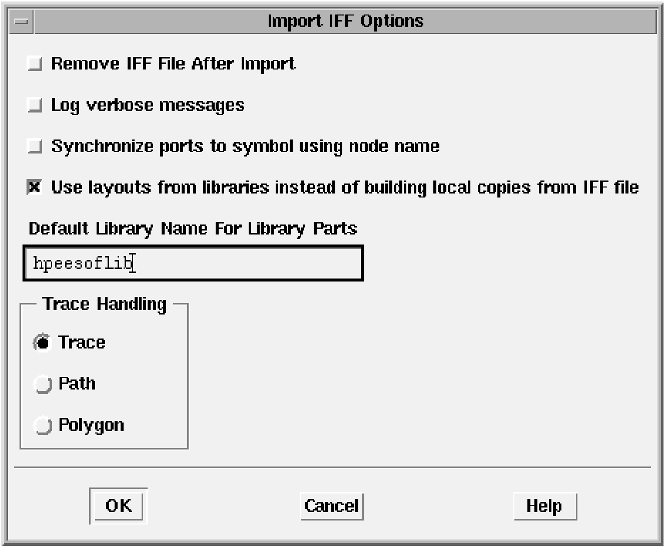
4. Select the IFF file you want to import and click **OK**. You are returned to the Import dialog box and the selected filename appears in the field labeled *Import File Name (Source)*.



File Type: IFF

Setting the Import Options

In the *Import* dialog box, click **More Options** to define the IFF import options. The *Import IFF Options* dialog box appears.



Import IFF Options

□ Remove IFF File After Import

□ Log verbose messages

□ Synchronize ports to symbol using node name

☒ Use layouts from libraries instead of building local copies from IFF file

Default Library Name For Library Parts: hpeesoflib

Trace Handling

● Trace

○ Path

○ Polygon

Note The Import IFF Options dialog box that appears is dependent upon where you execute the IFF import from (i.e ADS *Main*, *Layout* or *Schematic* window).

Set the Import Options:

1. Ensure the *Remove IFF File After Import* option is deselected. When selected, the IFF file is removed once it has been successfully imported. The default option is deselected.

2. Set the *Log verbose messages* as desired. When this option is selected, *all* translation information is recorded in the *ifftolib.log* file resulting in step-by-step description of what happened internally during your translation. This option is primarily intended to be used as a diagnostics tool. The default mode for this option is deselected. Note that error and warning messages will always appear in your status window regardless of this selection.

3. Ensure the *Synchronize ports to symbol using node name* option is deselected. This option resets the symbol pin numbers to match port numbers based on the node name of the schematic port. By default, symbol pin numbers are matched to schematic port numbers based on the port's instance name.

4. Check the *Default Library Name for Library Parts* field. This field is used for specifying the default library name. When the IFF file does not specify a library name for a component that needs to be created, the library name specified in this field is used. The IFF files that are created by the *nettrans* program specify the library SPICE INPUT, so the library name in this field will not change the outcome of the import.

Note The *Default Library Name For Library Parts* field is identical to the field of the same name in the *Export IFF Options* dialog box. Changes made to this field will modify the contents of the field in the *Export IFF Options* dialog box.

5. Accept the default *Trace* setting within the *Trace Handling* field. This field is normally used to enable you to select how you want layout traces interpreted during a translation. Because the Netlist Translator does not support layout import and export, this field is not used.
6. Click **OK** to save your settings or **Cancel** to retain the default settings. You are returned to the *Import* dialog box and the selected filename appears in the field labeled *Import File Name (Source).*
7. Click **OK** to import the IFF file into ADS. The IFF Importer performs the automatic translation of your IFF file into an ADS Netlist.

About Component Libraries

A component library in ADS consists of a collection of component definitions. Each primitive component has an associated component name, symbol and predefined component parameters that include relevant physical and electrical characteristics.

The IFF translator can be used as the initial step in creating an ADS component library; however, this topic is outside of the scope of this manual. Creating an ADS component library using IFF requires specialized tools and training. If you are interested in learning more about this topic, contact Agilent EEsof-EDA's Solution Services.

Adding the NetlistInclude Component

In Advanced Design System, you can insert a saved netlist file into an ADS simulation via an include statement by using the *NetlistInclude* component. The

NetlistInclude component is used to link the netlist that was translated into an ADS netlist, to the ADS schematic capture environment at the time of simulation.

The NetlistInclude component enables you to accomplish the following:

- Add parameter definitions, equations, and special sub-circuits from your design to the Advanced Design System schematic.
- Initialize technology dependent model parameters or variables in a file that can then be automatically inserted into the ADS netlist.
- Allow sectional includes to handle corner case analysis.

You can use a NetlistInclude in the following placements:

- In the top level of hierarchy, if the included file contains subcircuit definitions (it can contain an entire model library)
- In any circuit, if the included file contains only variables and/or model cards

A NetlistInclude component referencing a file containing subcircuit definitions *cannot* be placed on a schematic that is not at the top level of hierarchy, unless you do the manual steps defined below.

Manual Translation and Placement of a NetlistInclude Component

Note These steps are only necessary when doing manual translation and placement of the NetlistInclude component. When using the user interface for a translation to an ADS netlist, the ITEM_NOSUBNET_HEADER_EX field is set by the translator.

Suppose you have a top level design that includes a component *abc* and a DC simulation component. If the subcircuit named abc contains a NetlistInclude component that references a file with a subcircuit *xyz*, the resultant netlist would look like the following:

```
define abc (in out)
define xyz (input output)
parameters R1
R:R1 input output R=R1
end xyz
end abc

abc:x1 _net1 0

DC:DC1
```

In ADS, the nested define statements (*xyz* defined within the definition of *abc*) would cause an error such as the following when simulated:

```
Error detected by HPEESOFSIM during netlist parsing.
    In file ’netlist.log’ at, or just before, line 7.
        Syntax error in component call:
        define xyz (input output)
^
```

To avoid this situation, you must manually modify the AEL item definition for the subcircuit. Discussion of item definitions is beyond the scope of this manual. Please refer to the *Advanced Design System AEL* manual.

If you have a basic understanding of the item definition, you can proceed to modify the Extra item attribute, which is field 15 in the *create_item()* function. To find this function, go to the *networks* subdirectory of your project directory and edit the ael file that has the same name as the design (*abc* in this example). This field needs to be set to ITEM_NOSUBNET_HEADER_EX. This tells the ADS netlister not to put the define and end statements around the contents of the subcircuit. This will cause another problem in that abc will not be defined at all now. To avoid this situation, *abc* should be defined in your original Spectre netlist.

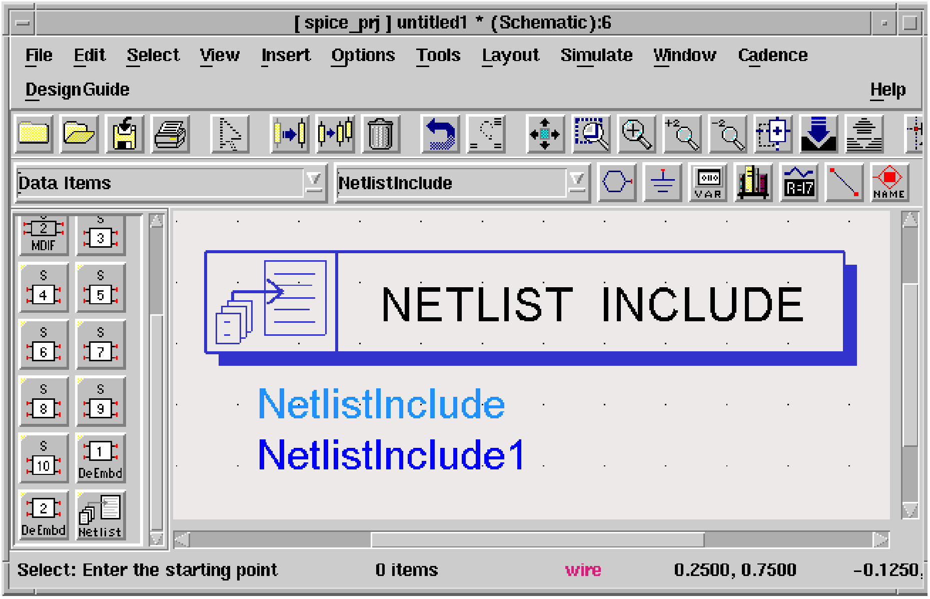
Note If it is at all possible, avoid placing a NetlistInclude component that references a file containing one or more subcircuits anywhere other than the top level of your design hierarchy.

Placing a NetlistInclude Component

If you imported your netlist using the command line, you will need to insert a NetlistInclude component in your new schematic.

To place a NetlistInclude component in an ADS schematic:

1. Select the *Data Items* component palette from the ADS schematic window.
2. Click the NetlistInclude component to place the component on the schematic. An instance is attached to your cursor. Place the component instance on the schematic as desired (click to place).



Select: Enter the starting point   0 items   wire   0.2500, 0.7500   -0.1250,

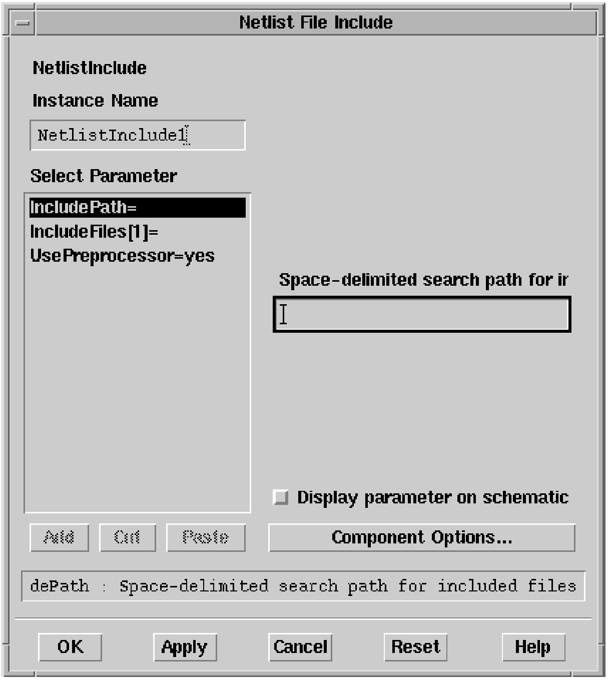
Figure 3-2. The NetlistInclude Component

3. You can continue to place component instances or choose the *Cancel Command And Return To Select Mode* icon to proceed with the next step.

4. To edit the netlist specifications choose **Edit > Component > Edit Component Parameters** or click the *NetlistInclude* component on the schematic.

   A dialog box appears, enabling you to specify the parameters of the file to include. See Table 3-3 for a description of each parameter.

Note You can also click the line to be changed directly in the schematic instance and edit as desired.



Netlist File Include

NetlistInclude

Instance Name

NetlistInclude1

Select Parameter

- IncludePath=
- IncludeFiles[1]=
- UsePreprocessor=yes

Space-delimited search path for ir

□ Display parameter on schematic

dePath : Space-delimited search path for included files

Table 3-3. Parameter Definitions for the NetlistInclude Component

NetlistInclude Parameter	Description
IncludePath=	Specifies the space-delimited search path for included files.
IncludeFiles=	Specifies the list of files to include.
UsePreprocessor=	Specify "yes" to use a #include directive, or "no" to copy the full text of the file (default is "yes").

Using a Custom Component

If you have created a custom component in Advanced Design System that you want the system to use in place of the default ADS component when it imports a Spectre netlist, you must create a translation table in the *Import Netlist Options* dialog box that specifies which custom component to use.

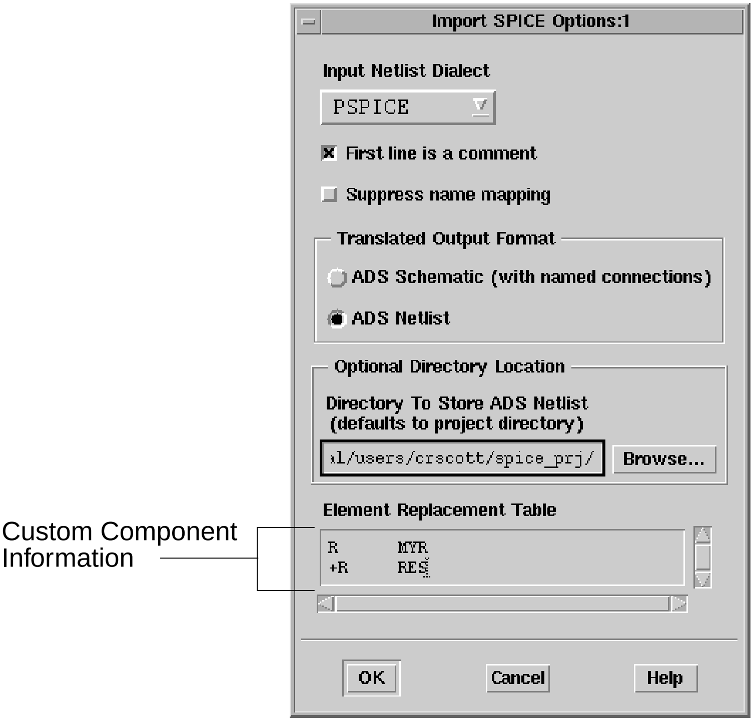
Creating a Translation Table

A translation table is simply a two column table that does the following:

- Specifies a standard ADS component (left-hand column) and the custom ADS component (right-hand column) used to replace it
- Specifies the standard ADS component parameters (left-hand column) and the custom ADS component parameters (right-hand column) used to replace them

To create the translation table perform the following steps:

1. Begin your normal import operation. See "Importing a Netlist File Using the User Interface" on page 2-1.
2. When you are "Defining Netlist Options" on page 2-13 using the *Import Netlist Options* dialog box, enter the translation table in the *Element Replacement Table* field.



Import SPICE Options:1

Input Netlist Dialect

PSPICE

☒ First line is a comment

□ Suppress name mapping

Translated Output Format

○ ADS Schematic (with named connections)

● ADS Netlist

Optional Directory Location

Directory To Store ADS Netlist
(defaults to project directory)

al/users/crscott/spice_prj/ Browse...

Element Replacement Table

```
R       MYR
+R      RES
```

OK Cancel Help

Refer to Table 3-4 for an example of how to format the information in the *Element Replacement Table* field.

Table 3-4. Example Translation Table

`# Standard ADS Component`	`Custom ADS Component`
`# +Standard ADS Component Parameter`	`Custom ADS Component Parameter`
`R`	`MYR`
`+R`	`Res`
`+Tnom`	`T`

Comment Character - The system recognizes the pound sign (#) as a comment character. Note that the first two lines in Table 3-4 are comments used to describe the elements of the table.

Component - Both the standard ADS component (**R**) and the custom ADS component (**MYR**) are given in line 3 of Table 3-4. On import, any time the translator would normally place the **R** component in ADS, the system will place a **MYR** component instead.

Note The component name (**R**) in the left-hand column is not the Spectre name. It is the ADS component that would be placed if the table were not used. Performing the import first, without the mapping, will allow you to see the name of the component to be replaced.

Component Parameters - Component parameter mappings are listed one per line following the component name. Each parameter line starts with the **+** character. List only the parameters that are unique to the custom component. It is not necessary to list the parameter names on the custom component that are the same as the ADS component; the translation automatically matches the correct parameters.

The only limitation on import is that new items have a similar symbol to the replaced component so that the placement and wiring of the schematic work correctly. This means that the custom symbol should be approximately the same size as the default component, and the pins must be in the exact location on the two symbols. If the pins are not in the same location, the *-u2* option must be used. For more information on the -u2 option, refer to "Executing the Nettrans Command" on page 3-5.

To access the component mapping functionality while running the translator from the command line, use the *-m* option. You create the file from scratch or modify the *spice.map* file that is created in the project directory when an import is performed from the user interface. For more information on using the -m option, refer to"Executing the Nettrans Command" on page 3-5.

Example Netlist Translation

The following shows an example comparison of how the system might translate a SPICE netlist to an ADS netlist with or without the custom component defined in Table 3-4.

Using the *Example Spice Netlist* in Table 3-5, if no custom component was defined, the translator would generate the *ADS Netlist (without mapping)*. Using the translation table in Table 3-4, the resultant ADS netlist would appear as in the *ADS Netlist (using a translation table)* in Table 3-5.

Table 3-5. Netlist Comparison

SPICE Netlist	ADS Netlist (without a mapping)	ADS Netlist (using a translation table)
R1 1 2 50 Tnom=27	R:r1 _node1 _node2 R=50 Tnom=27	MYR:r1 _node1 _node2 Res=50 T=27

Note If your file contains functions that are not Advanced Design System functions, refer to "Using Non-ADS Functions" on page 3-24.

Modifying the Translator Configuration File

The translator configuration file, *spctoiff.cfg*, is a file that is used when the translator creates a schematic. It is stored in $HPEESOF_DIR/config. This file contains pin locations and sizes for the symbols of all the components that the translator knows how to translate.

Since the Netlist Translator runs as a separate process from ADS (see "Understanding the Import Operation" on page 3-1), the translator cannot determine this information on the fly. Periodically, symbols will be updated but the translator configuration file will not be updated at the same time. In this case, you can manually modify the file if needed to avoid overlapping components in the schematic.

The file consists of a block of information for each component. The first component in the file is the Capacitor (C).

```
C  2  0  -0.625  0 .625  1.5  -3.25  0  0
```

There are four lines that start with C. The C is followed by an integer number which indicates the number of pins on the component. This is followed by an integer number that indicates the rotation in degrees of the component.

The next four fields define the **bounding box*** of the symbol, calculated with the default settings for parameter visibility. If you choose to make all parameters visible while editing the schematic or if you make the font of the parameters larger than default, you may need to increase the size of the bounding box, or spread your components out on the schematic.

The final two coordinates on the first line are ignored.

The following two lines are the pin information.

0 0 180 Pin 1 is located at 0,0 on the symbol (the origin) and rotated to 180 degrees.

1 0 0 Pin 2 is located at 1,0 on the symbol and placed at 0 rotation.

In the rare case that this file needs to be modified, usually only the bounding box information would have to be modified. If there seems to be a problem with the file, you can request an updated file from Agilent EEsof-EDA Customer Support, or you can make a few simple changes yourself if it is needed urgently.

The file can be modified for your installation by a system administrator. For individual use, a copy of the file can be placed in the user's $HOME/hpeesof/config directory. The file located in the user's directory will take precedence over the system file. For command line execution or batch processing, the file can also be placed in the directory where the translator is running.

* The bounding box of a symbol is the technical term for the rectangular space occupied by a symbol on the schematic. The four coordinates given are the upper left corner and the lower right corner, when the origin of the component is placed at 0,0 on the schematic. The origin of the component is the position that is tied to the mouse when the drag image is visible. Where you click the mouse to place a component becomes the location where the origin of the component is placed. This *hot-spot* is usually pin 1.

Importing an HSpice File from Cadence after Parasitic Extraction

HSpice files output from Cadence have been observed to have syntax errors that will not be corrected by the translator. Some of these syntax errors need to be corrected manually before importing or they will cause the ADS simulation to fail. Others may be corrected after the translation and before simulation.

Make the following changes to your ADS netlist or schematic before starting the import:

- Expressions are usually enclosed in single quotes. Occasionally, expressions have been observed to be enclosed in double quotes, which are not valid. This must be corrected before the file can be translated or the translator will drop part of the expression.

Make the following changes to your ADS netlist or schematic after the import has completed:

- Values have been observed with non-matching pairs of parentheses. These will be passed through the translator but cause the simulator to fail. This can be fixed prior to translation or prior to simulation.
- Files have been observed with multiple declarations of the same variable. All will pass through the translator but cause an error in the simulator. Parameters should be declared only one time. Remove the extra parameters and re-simulate.
- Subcircuit calls have been seen with parameters that were not declared on the subcircuit when it was defined. Again, the translator will not check for this error but the simulator will halt with an error. Remove the extra parameters and resimulate.

Using Non-ADS Functions

If your file contains functions such as **gauss()** or **pwr()** or any other function that is not a built-in ADS function, the simulator will give an *Unresolved reference* error.

Look for the function in the supplied function file:

$HPEESOF_DIR/links/spice/spicefunction.net (for SPICE functions)

$HPEESOF_DIR/links/spice/perl/spectre/spectrefunc.rul (for Spectre functions)

The function can be copied into the ADS netlist, placed in the schematic, or the whole file can be included by entering the following line in the ADS netlist:

#include "<*path*>/spicefunctions.net" (for SPICE functions)

#include "<*path*>/spectrefunctions.net" (for Spectre functions)

Replace <*path*> with the correct location for your system.

Chapter 4: Simulating the Translated Netlist

Once your netlist is successfully translated into Advanced Design System, you can simulate your new circuit in ADS. This chapter discusses some of the details involved in circuit simulation and provides references to other sources. The example used in this chapter refers to the Transmission-Line Inverter design created in "Including Models and Subcircuits" on page 2-18.

Example SPICE Design

The following two example SPICE files make up a simple transmission line inverter. The first SPICE netlist (tline.sp) is a subcircuit.

Table 4-1. Transmission Line Subcircuit (tline.sp) Netlist

```
*TRANSMISSION-LINE SUBCIRCUIT
.SUBCKT TLINE 1 3
T1 1 2 3 4 Z0=50 TD=1.5NS
T2 2 0 4 0 Z0=100 TD=1NS
.ENDS TLINE
.END
```

The next SPICE netlist (inverter.sp) includes additional components that complete the transmission line inverter design.

Table 4-2. Transmission Line Inverter (inverter.sp) Netlist

```
*TRANSMISSION-LINE INVERTER
V1 1 0 PULSE 0 1 0 0.1NS 0.1NS 20NS 40NS
R1 1 2 50
R2 4 0 50
```

The resultant file from the import process is called *tline_inv.dsn*. See the *Netlist Translator for SPICE* manual, chapter 2 *Importing a SPICE File* for more information.

Note Although *tline_inv.dsn* is a design translated from SPICE netlists, the simulation steps and concepts described in this chapter are directly applicable to Spectre imported designs.

Setting Up and Performing a Simulation

Setting up and performing a simulation in Advanced Design System requires several steps:

1. "Opening the Schematic" on page 4-2
2. "Adding Simulation Components" on page 4-3
3. "Running the Simulation" on page 4-3
4. "Displaying the Results" on page 4-6

Opening the Schematic

To open an existing schematic design:

1. Choose **File > Open** in the ADS Schematic window. The *Open Design* dialog box is displayed. You can use this dialog box to select the design you wish to simulate.
2. Choose the appropriate project from the *Project* drop-down list in the *Open Design* dialog box.
3. Click the appropriate design from the *Designs* list in the *Open Design* dialog box. (For this example the *tline_inv.dsn* file created in the *Netlist Translator for SPICE* manual will be used.)
4. Click **OK** to include the schematic in the ADS Schematic window.
5. To setup the schematic to output selected node data to the dataset, rename the nodes using the *Node Name* command. For the *tline_inv.dsn* example, rename the nodes as follows:

   - **_node1** to **vsource**
   - **_node2** to **netin**
   - **_node4** to **netout**

   For more information on Naming Nodes, refer to "*Defining the Node Names*" in the *Netlist Translator for SPICE* manual.

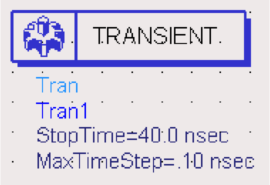
For more information on opening designs, refer to "*Opening an Existing Design*" in the ADS *User's Guide*.

Adding Simulation Components

After translating your netlist into Advanced Design System, you may need to add stimulus or simulation control components to your design. In the schematic window containing the design you want to simulate, select the appropriate simulation control elements from the component palette.

To add a simulation component to the *tline_inv.dsn* example:

1. Select the **Simulation - Transient** from the component palette. An instance of the TRANSIENT component is attached to your cursor (see Figure 4-1).
2. Using your mouse, locate the component to an appropriate area on the schematic and click to place the component.
3. Click and edit the values of the *StopTime* and *MaxTimeStep* Tran parameters. For this example as follows:
   - StopTime=40 nsec
   - MaxTimeStep=.10 nsec



StopTime=40.0 nsec
MaxTimeStep=.10 nsec

Figure 4-1. The ADS Simulation-Transient Component

For more information on using the Transient/Convolution Simulation component, refer to "*Transient and Convolution Simulation*" in the ADS *Circuit Simulation* manual.

Running the Simulation

Once you have added all of the necessary simulation components to your design, execute the simulation.

To run the simulation, choose **Simulate > Simulate** from the menu selection or click the *Simulate* button in the toolbar.

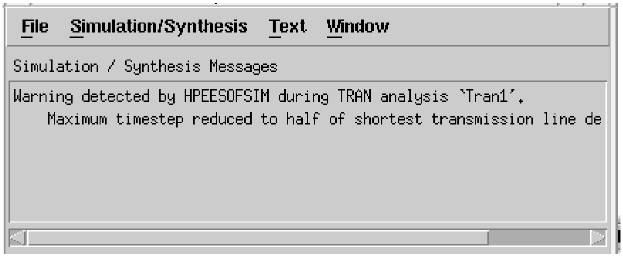
Simulation/Synthesis Message Window

The Simulation/Synthesis Message window appears whenever a simulator is launched and displays messages about the status of the current process, as well as warning messages. The window contains two information panels:

- *Simulation/Synthesis Messages*
- *Status/Summary*

Simulation/Synthesis Messages

The *Simulation/Synthesis Messages* portion of the window displays detailed messages about problems encountered during a simulation or synthesis, and where possible, what you can do to solve the problem.

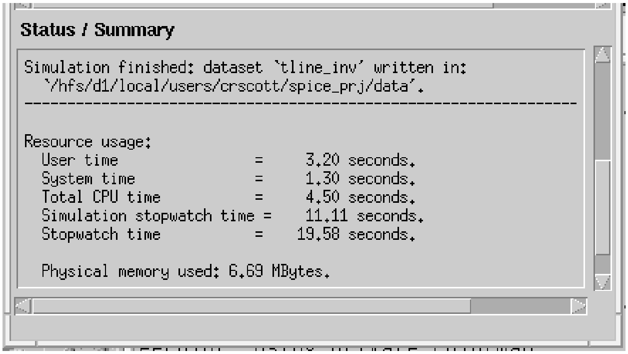


```
Warning detected by HPEESOFSIM during TRAN analysis `Tran1'.
    Maximum timestep reduced to half of shortest transmission line de
```

Hint Watch for a message that prompts you to click to view the source of the problem. Clicking this message highlights the component(s) (in the Schematic window) causing the problem.

Status/Summary

The *Status/Summary* portion of the window displays a *Simulation finished* message, statistics such as how long the simulation or synthesis took, and the system resources used.



```
Simulation finished: dataset `tline_inv' written in:
  `/hfs/d1/local/users/crscott/spice_prj/data'.
-----------------------------------------------------------------

Resource usage:
  User time               =     3.20 seconds.
  System time             =     1.30 seconds.
  Total CPU time          =     4.50 seconds.
  Simulation stopwatch time =   11.11 seconds.
  Stopwatch time          =    19.58 seconds.

  Physical memory used: 6.69 MBytes.
```

Viewing Simulation Status and Error Messages

When the simulation/synthesis is finished, you can save the displayed information to file or you can send it directly to the printer.

To save the currently displayed information to file with a default filename:

> Choose **File > Save** and click **OK**. The default filename consists of the simulation process number (from the title bar of the window), with a prefix of the string *sessloghpeesofsim* and a file extension of *.txt*. The file is saved to the current project directory.

To save the currently displayed information to a file with a different filename:

> Choose **File > Save As**. Supply a filename and click **OK**. The file is saved to the current project directory.

Note If you have changed projects during the current session, the file may be written to the initial project opened in this session.

To send the information directly to the printer:

1. If needed, choose **File > Print Setup** to establish the desired setup and click **OK**.
2. Choose **File > Print**. The displayed information is sent to the printer. For details on print setup, refer to *"Printing and Plotting"* in the ADS *User's Guide*.

For more information on performing a simulation, refer to "*Simulation Basics*" in the ADS *Circuit Simulation* manual.

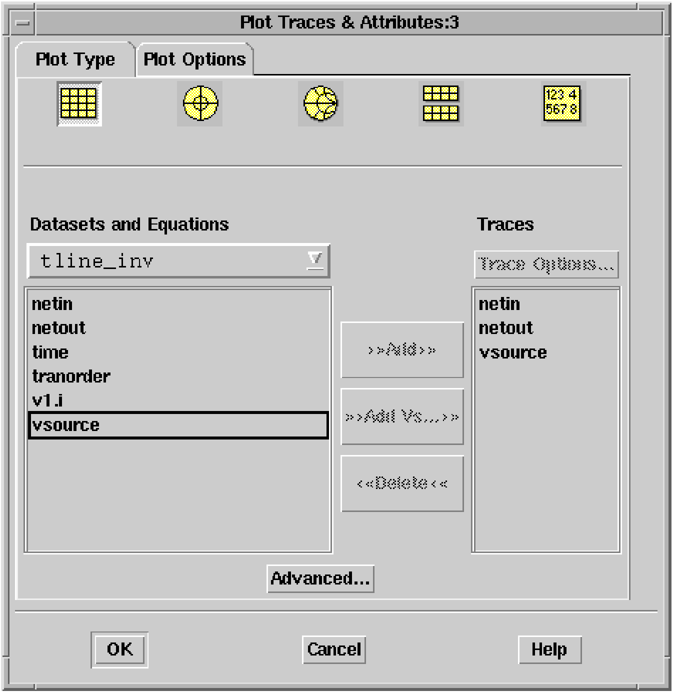
Displaying the Results

After your simulation is complete, ADS automatically opens a Data Display window so that you can view your simulation results. You can also open a Data Display window to see the results of a simulation analysis by choosing *Window > New Data Display* from the Main or Schematic window. After a Data Display window is open, you can select an independent swept variable, select dependent measurements, scale the data, and add captions to your graph. Then you can print or plot the graph.

To view the results of the simulation in a plotted Data Display:

1. Choose the *Rectangular Plot* icon to drag and drop the plot frame in the Data Display window.

   The *Plot Traces & Attributes* dialog box appears.



Datasets and Equations

tline_inv

netin
netout
time
tranorder
v1.i
vsource

Traces

netin
netout
vsource

2. Select the dataset to display from the *Datasets and Equations* drop-down list (*tline_inv* for this example).

3. To define the traces for display on your plot, individually double-click the data (*netin*, *netout* and *vsource* for this example) to add each item to the *Traces* field. Alternatively, you can select each item and then click the **Add** button.

4. Click **OK** in the *Plot Traces & Attributes* dialog box. The graph is displayed in the Data Display window. Choose **View** > **View All** in the Data Display window to enlarge the plot to fit the display window.

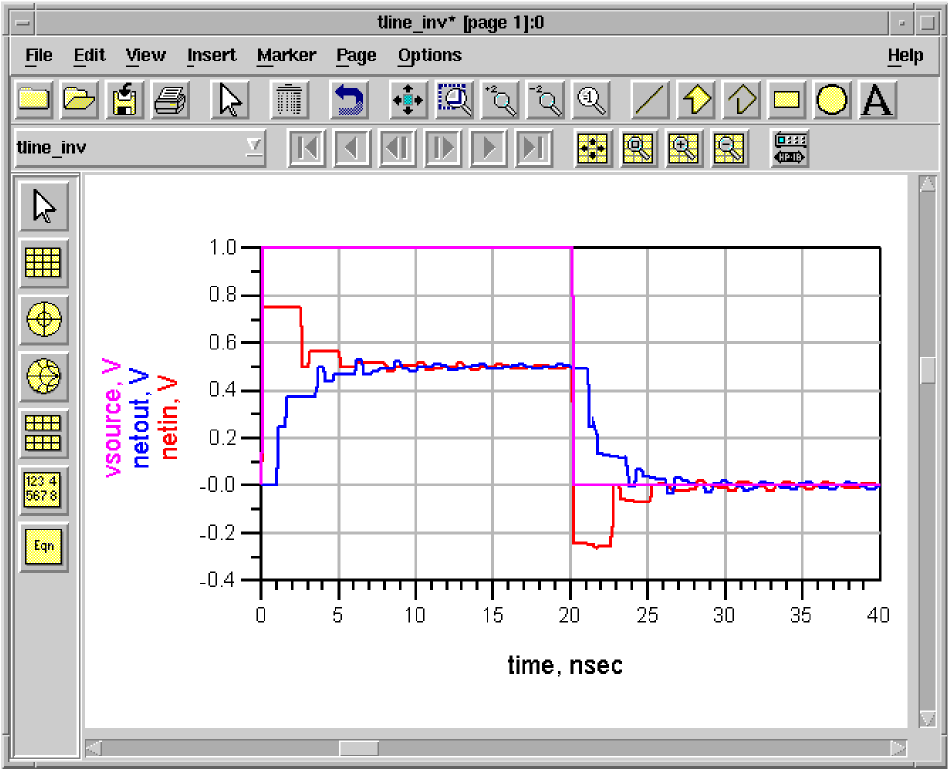


Figure 4-2. Transmission-Line Inverter Simulation Results

Figure 4-2 displays the simulation results for the Transmission-Line Inverter (*tline_inv*).

Analyzing and Comparing the Results

It is important to understand the results of your simulation to ensure that what you are looking at is not a result of your netlist translation, but rather the results of your circuit design. You can analyze your data by using the various tools available in the ADS Data Display, including a variety of plots, formats, markers and equations. For more information on working with data displays, refer to "*Data Display Basics*" in the ADS *Data Display* manual.

After you understand your results, you can compare your ADS simulation results to your original Spectre simulation results. Ideally, you would like to compare your Spectre simulation data against your ADS simulation data using the same data file format; however, this is not always possible. There are two suggested methods that can be used to compare your simulation results:

1. Visually compare your ADS Data Display output to the visual display output you have available in your simulator tool. If you have the ability to output your data to an *MDIF*, *Citifile* or *Touchstone* file format, you can use the ADS Instrument Server to convert your data to an ADS Dataset and view it in the ADS Data Display.
2. If your Spectre simulation results are in ASCII text format, you can convert your ADS dataset to *MDIF*, *Citifile* or *Touchstone* file format (ASCII text) and compare the data.

For more information on the different file formats, refer to "Converting to an ADS Dataset" on page 4-8.

Note Remember that Spectre simulators use different simulator technology from the ADS simulator; therefore, there may be incompatibilities between your file and the resultant ADS file that the translator was unable to accommodate. Refer to Chapter 6, Troubleshooting for more information.

Converting to an ADS Dataset

Ideally, you would like to compare your simulation data against your ADS simulation data in the same data display format. One method of viewing your simulation data in an ADS Data Display is to convert your Spectre simulation data into a common file format that ADS has the ability to read. Once your data is in an accessible format, you can convert it to an ADS Dataset using the Instrument Server. ADS Datasets can be viewed in the ADS Data Display enabling you to view your original simulation using the same display technique as your translated data. The file formats supported by the ADS Instrument Server and associated file extension are listed in Table 4-3.

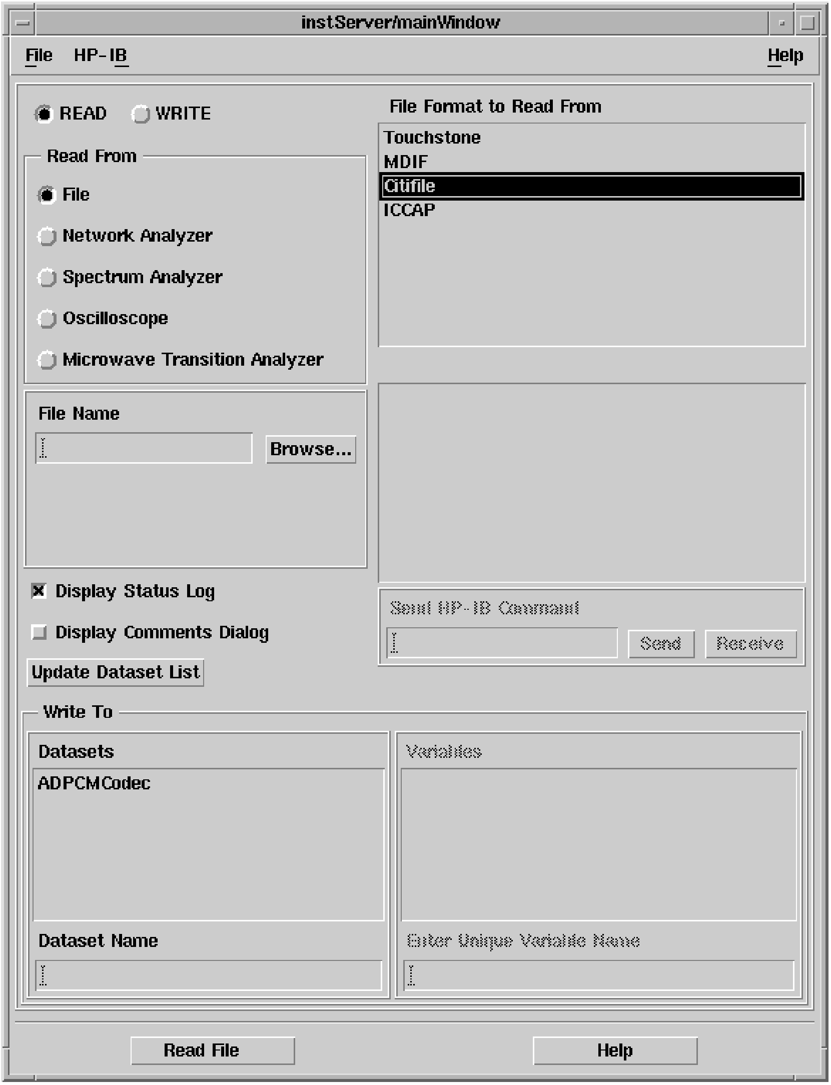
Table 4-3. Instrument Server Supported File Types

File Type	File Extension	Description
Dataset	.ds	ADS uses datasets to collect and store data either from internal sources, such as a simulation, or from external sources, such as a network analyzer or Touchstone file. The dataset can be viewed and analyzed using the ADS Data Display tool. For more information on Datasets and Data Display, refer to the ADS Data Display manual.
Touchstone File	.s*p where * is the number of ports	Touchstone data files are analog/RF circuit component data files that contain small-signal G-, H-, S-, Y-, or Z-network parameters described by frequency-dependent linear network parameters for 1-, 2-, 3-, or 4-port components. The 2-port component files can also contain frequency-dependent noise parameters. For more detailed information on the Touchstone file format, refer to the ADS Circuit Simulation manual.
Citifile	.citi	The Common Instrumentation Transfer and Interchange file (CITIfile) format is a standardized data format that is used for exchanging data between different computers and instruments. For more detailed information on the CITIfile format, refer to the ADS Circuit Simulation manual.
MDIF	.p2d, .s2d, .t2d	Measurement Data Interchange Format (MDIF) files are component data files for the .p2d, .s2d, and .t2d files.

For more information on the file formats listed above, refer to "*Working with Data Files*" in the ADS *Circuit Simulation* manual.

To use the Instrument Server to create an ADS dataset:

1. From the ADS Schematic window, start the instrument server by choosing **Window > File/Instrument Server**. The *instServer/mainWindow* dialog box appears.



● READ ○ WRITE

● File

☒ Display Status Log

□ Display Comments Dialog

2. In the instrument server dialog box, click **READ**.

3. Under *Read From*, click **File**.

4. Under *File Format to Read From*, click the appropriate file format (i.e. **Touchstone**, **MDIF** or **Citifile**).

5. Under *File Name*, click **Browse** to navigate file paths and select your Spectre simulation result file.
6. Click **Display Status Log** to view any messages or errors that may occur during the transfer.
7. In the *Write To* field, click **Datasets**.
8. In the *Dataset Name* field, enter the name of the Spectre dataset.
9. When all of the above settings are entered, click the **Read File** button.

For more information on reading files using the Instrument Server, refer to “*Reading from and Writing to Files*” in the *Using Instruments with ADS* manual.

Chapter 5: Comparing Results

This chapter provides information on how to compare your translated design output results with your original netlist. Once you have translated, verified and simulated your netlist, compare your ADS results with your original version.

Comparing the Schematic

For wired or unwired import options, compare your translated ADS Schematic with your original netlist. For a small design, you can visually verify connections by matching node numbers. For information on nodes flagged with red diamonds, refer to "Checking for Unconnected Nodes" on page 2-17.

To compare a larger schematic with the original netlist, first generate an ADS netlist. To generate an ADS netlist from a schematic without starting a simulation:

1. Open the ADS command line from the *Main* window.

   Choose **Options > Command Line**.

2. Enter the following command in the *Command* field:

   ```
   de_netlist();
   ```

   After applying this command, the netlist is stored in a file called *netlist.log* in the current project directory.

3. Proceed to the next section for instructions on how to compare the translated ADS netlist to the original netlist.

Comparing the Netlist

You can compare your translated ADS netlist to your original netlist to verify that nodes were connected correctly and parameters were translated correctly. If you performed your translation from the command line and selected the -g option to create an ADS netlist, you can easily view and compare your translated netlist with your original netlist.

Since Spectre, SPICE, and ADS netlists are ASCII based text files, they can be viewed using any ASCII text editor. To view the original netlist, open a text editor and load the file defined as *input_filename* in the nettrans command. To view the translated ADS netlist, open a text editor and load the file *netlist.log* or the file defined as *output_filename* in the nettrans command.

You can now compare your original netlist to the translated ADS netlist noting the differences between Spectre and ADS.

Comparing Original and ADS Netlists

You can now compare your original netlist to the translated ADS netlist noting the differences between the original and ADS. Table 5-1 shows an example comparison of a simple SPICE netlist translation. The comparison is the same for Spectre.

Table 5-1. SPICE Netlist vs. Translated ADS Netlist

#	Original SPICE Netlist	Translated ADS Netlist
1	`BJT Curve Tracer Example`	`; BJT Curve Tracer Example`

2	`.param vce=0 ibb=0`	`vce=0`
		`ibb=0`

3	`V1 3 0 VCE`	`V_Source:v1 _node3 0 Vdc=vce Vac=0`
4	`I1 0 10 IBB`	`I_Source:i1 0 _node10 Idc=ibb Iac=0`
5	`Q1 3 10 0 BJTM1`	`bjtm1:q1 _node3 _node10 0 Area=1 Region=1`
6	`.model bjtm1 NPN`	`Mode=1`
7	`.end`	`model bjtm1 BJT NPN=1 PNP=0 RbModel=1 Tnom=27`

Note The numbers in the left hand column of Table 5-1 coincide with the numbers in the descriptions below. Each number references a line or group of lines in the adjacent netlist.

Here are some details you might notice in the translated netlist in Table 5-1:

1. There is a semi-colon preceding the first line in the translated netlist. The translator inserted the comment character even though there was no comment character in the SPICE file. This is because the SPICE simulator always considers the first line of a SPICE file to be a comment. To override this behavior in the translator, deactivate the *First line is a comment* check box in the Import dialog or use the *-l* command line option.
2. The .param keyword in the SPICE netlist indicates the beginning of a parameter list. Each of the parameters defined are written separately in the ADS netlist.

3. Components are indicated by a component designator (i.e. V_Source) followed by a colon and then a unique item name (i.e. v1).
4. Component node numbers are replaced with _node*n*. The translator adds the _node prefix to each node number to reduce the amount of data output by the simulator. To suppress this renaming, refer to the *Suppress name mapping* option in "Setting the Import Options" on page 2-5
5. If a component references a model (bjtm1), the unique model name will be listed first, instead of a component designator (V_Source).
6. The .model keyword in SPICE indicates the beginning of a model. The translator converts the SPICE model to an equivalent ADS model.
7. The .end (end of circuit) command has no equivalent in the ADS netlist file. The translator stops reading after encountering the .end command.

Now that you have compared your translated netlist to the original SPICE netlist and made some observations, you can view the translation log (spctoiff.log) to help understand some other reasons the translated netlist appears as it does. For information on reviewing the spctoiff.log file, refer to "Viewing the Translation Log" on page 3-9

Comparing Models and Devices

Compare each of your translated models and devices with those in your original netlist.

Model parameters are not always defined consistently for every simulator. There are differences between ADS, Spectre and SPICE. Additionally, each simulator may use different equations in their model calculations. These inconsistencies may lead to differences in simulation results for translated models.

If you have verified that the model parameters have translated as you expected by using the information in Chapter 7, Translating a Device and Chapter 8, Translating a Model, and your simulation results are not what you expect, please contact Agilent EEsof-EDA Customer Support for the latest information on model compatibilities. You may be asked to submit your archived project to help resolve the issue.

For information on troubleshooting your translation, refer to Chapter 6, Troubleshooting.

Comparing ADS Results to Hspice

Hspice uses different values for the physical constants k and q. This can subtly affect some of the temperature scaling code. It can make small differences in kT/q, which is part of the equation for a forward biased diode.

Format	**Boltzmann's Constant k**	**Electron Charge q**
ADS Codata-86	1.380 658 0 x10–^{23}	1.602 177 33 x10^{-19}
Spice2, Spice3, Spectre	1.380 622 6 x 10^{-23}	1.602 191 8 x10^{-19}
Hspice	1.380 620 0 x 0^{-23}	1.602 120 0 x 10^{-19}
Codata-98	1.380 650 3 x 0^{-23}	1.602 176 462 x 10^{-19}

Care must be exercised when comparing the currents flowing through reverse-biased PN junctions. A diode with Is=1x10^{-14} should have a current of –1x 10^{-14} at –5V; instead it will show –5 x10^{-12}. This is because ADS/Hspice adds a small conductance of size gmin (nominally 1x 10^{-12} S) in parallel with every reverse biased PN junction to aid convergence. Thus if these currents are compared, all that is being compared is gmin and not the reverse-biased diode results.

Chapter 6: Troubleshooting

This chapter provides information on possible translation failures from the Netlist Translator and how to resolve issues that arise.

When an error occurs during a translation, an error message is written to the *spctoiff.log* file. Sometimes, when potential problems are found or the translator makes a change that the user needs to be informed of, warning messages are also written to the *spctoiff.log* file.

Debugging Imported Designs

Since no two design environments or simulators are alike, there may be problems that arise in the translation. Here are a few tips for debugging the translation.

1. Read the log files. The log files are ASCII text files that contain valuable translation information.
    - The *spctoiff.log* file will always be present unless the translation completely failed. Running in a read-only directory might cause this problem. This file will list any components that could not be translated. Unrecognized syntax or incompatible models are the most likely causes of this problem.
    - The *ifftolib.log* file will be present if a schematic output was requested. This file lists problems encountered during creation of the schematic. For a summary of the import process, refer to “Understanding the Import Operation” on page 3-1.
2. Choose the netlist output option. Since the netlist to netlist translation gives you an output file that is human-readable, it is very easy to compare line by line with the source netlist. Even if you desire a schematic eventually, this method can be used to debug the translation. For information on netlist comparison, refer to “Comparing the Netlist” on page 5-1.
3. Output the netlist from ADS. First open the command line window from the ADS Main window.

    Choose **Options > Command Line**.

    Enter the command **de_netlist();**

    This will create a file called *netlist.log* in the project directory. This is the information that goes to the simulator.

If you are using a *NetlistInclude* component, the *#include* statement will be visible. To flatten that and show the actual contents of the file, find the *NetlistInclude* component in the schematic and set its last parameter NetlistDebugMode=0. Regenerate the netlist from the ADS command line.

4. The translator can be run manually from the command line using the *nettrans* command. For detailed instructions on using the nettrans command, refer to "Importing a File from the Command Line" on page 3-3.

5. When debugging a very large file, the best thing to do is to break the file down into smaller pieces. Subcircuits and include files can be imported one at a time; however, you should import the lowest level of hierarchy first so the referenced design is created before it is used.

If the imported design appears to be syntactically correct and contains the correct data, but simulation results are not as expected, please contact Agilent-EEsof-EDA customer support for the latest information on model compatibility.

Error Messages

Invalid subckt line <*number*>, <*line text*>.

The translator has encountered a `subckt` line with unrecognized syntax. Correct the problem and rerun the translator.

Failed opening netlist file <*name*>.

The file specified by <*name*> was not found, or could not be opened. Verify that the file exists, and then check file and directory permissions.

Failed opening output file <*name*>.

The file specified by <*name*> could not be opened for write. Check file and directory permissions. Make sure you are not running ADS from a read-only directory.

Failed opening logfile spctoiff.log.

The file `spctoiff.log` could not be opened. Verify that the file exists, and then check file and directory permissions.

Fatal error in position ports ( ). OR
Node <*name*> missing from memory.

These are errors internal to the translator. Have your system administrator contact Agilent Technologies technical support at *eesof_support@agilent.com*. In Canada and the United States, you can also call 1-800-473-3763. Elsewhere, contact your local Agilent Technologies sales office.

Recursive subckt reference found in network <*name*>.

A reference to subcircuit <*name*> was found in the subcircuit definition of the same name. A subcircuit cannot be placed within itself.

Failed memory allocation.

Not enough memory to complete the translation; the file is very large, or has many subcircuits. Separate the file into several smaller files and translate them individually. If needed, run the translation on a different machine

Found unmatched end of subckt at line <linenum>

An ends command was found but no subckt line corresponds to it. Correct the file and re-run the translation.

Invalid equation <eqn>

The translator failed parsing the equation while checking for reserved words or invalid characters. A complex equation may need to be broken down into simpler equations that reference each other if no obvious errors are visible.

Warning Messages

Item mapping file <*name*> not found.

The item mapping file specified by <*name*> was not found, or could not be opened. Verify that the file exists, and then check the file and directory permissions.

Failed attempting to convert node voltage syntax for <*varname*>.

Variable reference not found.

For information on this warning message, refer to the comments under "Capacitor Device" on page 7-7 or "Resistor Device" on page 7-5.

Failed opening include or library file <*name*>, line <*number*>.

The file specified by <*name*> was not found, or could not be opened. Verify that the file exists, and then check the file and directory permissions.

Failed processing line <*number*>, <*text*>.

Encountered syntax that cannot be translated. For more information on device, model and Command Line syntax, refer to Chapter 7, Translating a Device, Chapter 8, Translating a Model and Chapter 10, Translating Commands and Functions.

Skipping unsupported capacitor syntax PWL, line <*number*>.

Optional PWL syntax # capacitor was encountered and cannot be translated. See capacitor description "Capacitor Device" on page 7-7.

Skipping unsupported inductor syntax PWL, line <*number*>.

Optional PWL syntax for inductor was encountered and cannot be translated. See inductor description "Inductor Device" on page 7-9.

Skipping unsupported source syntax <*name*>, line <*number*>.

Encountered source syntax that cannot be translated. See source descriptions "Non-Linear Voltage-Controlled Current Source" on page 7-28, and "Linear Voltage-Controlled Voltage Source" on page 7-25.

Skipping unsupported <*name*> syntax, line <*number*>.

Encountered syntax for item <*name*> that cannot be translated. For more information on device, model and Command Line syntax, refer to Chapter 7, Translating a Device, Chapter 8, Translating a Model and Chapter 10, Translating Commands and Functions.

Skipping unsupported 4-port JFET device <*name*>, line <*number*>.

Specified JFET is not available in ADS. See JFET description "JFET Device" on page 7-22.

Skipping unsupported model type <*name*>, line <*number*>.

Specified model is not available in ADS. See model descriptions Chapter 8, Translating a Model.

Skipping unsupported statement <*name*>, line <*number*>.

Spectre <*name*> statement is unrecognized or unsupported. See control line descriptions Chapter 10, Translating Commands and Functions.

Skipping unrecognized element type: line <*number*>, <*name*>.

Element type is not one of supported elements for the selected Spectre type. For information on supported devices, refer to Chapter 7, Translating a Device.

Skipping unsupported element <*name*>.

Encountered element <*name*> that cannot be translated. For information on supported devices, refer to Chapter 7, Translating a Device.

Invalid level <*name*> - skipping model <*name*>, line <*number*>.

Encountered unexpected level value for model <*name*>. See model descriptions in the Chapter 8, Translating a Model.

Invalid library syntax - missing library name, line <*number*>.

See library syntax in the control lines section of "library" on page 10-3.

Model <*mname*> for device <*name*> in circuit <*circuitname*> not found.

Model referenced by device <*name*> is not found in the netlist. Check the names. Make sure the model is present globally or in the specified circuit. The model may exist or be added later in ADS but the translator will still generate a warning. If a model is not present in the file, the translator may not have enough information to place the proper device (ex: NPN vs. PNP). You may need to correct the schematic or netlist before simulating.

Referenced circuit <*name*> not found.

A circuit was used that has not been defined. Make sure the circuit is defined in the file or ADS.

Schematic not created for subcircuit <*name*> **with no translated components.**

A design will not be created if there is nothing to put in it. Look for a message regarding untranslated components.

Appended <*_?*> **to** <*item*> **due to a conflict with a reserved word.**

A partial list of Advanced Design System reserved words is listed in Table 6-1. The full list is much longer. If the translator finds a node, element or variable name that is the same as a reserved word, the name is appended with an _n, _e or _v respectively.

Table 6-1. Partial Listing of ADS Reserved Words

abs	db	e	ln	mag	nf	value
c0	deg	exp	ln10	max	step	y
cos	delay	i	log	min	v	z

For more information on ADS reserved words, refer to the section on "*Reserved Words*" in Appendix E of the *RFIC Dynamic Link Library Guide*.

The subcircuit "*ckt_name*" contained parameters that reference each other. In order to maintain compatibility with ADS, these have been moved off the subcircuit parameters list and are treated as circuit equations. Component instances that reference this subcircuit will not be able to directly override these values.

The subcircuit *ckt_name* contains parameters that reference each other. The ADS netlist format does not support this syntax. Any subcircuit equations that reference each other are moved from the subcircuit parameter list and they are listed as subcircuit equations. Component instances that reference this subcircuit will no be able to directly override these values.

Known Problems

Problem ADS simulation results do not match my original simulation.

Model differences exist between all simulators due to different default parameters and different equations. ADS models are usually implemented exactly as specified in the Berkeley model. Differences that occur between ADS and other simulators are beyond our control since the equations for these simulators are not published.

Problem I have a very large netlist with a number of included files.

If your set of files seems too large to be handled in one translation, the included files can be imported first. Design files will be created for all subcircuits. Close all designs before doing the next import. If a referenced design has already been imported, you can ignore the warnings from the translator and ADS that say the referenced design cannot be found.

Problem Symbol and pin topology on subcircuits is hard-coded for schematic imports.

If a file contains references to a custom subcircuit, the pins will not be connected properly if the symbol is not the standard nport symbol supplied in the design environment. This is only a problem for importing to a schematic (wired or unwired). A work-around is available for the method that uses named connections (unwired). For more information on working around the unwired method, refer to "Using the -u2 Option" on page 3-8. For more information on using *Named Connections*, see "Defining Netlist Options" on page 2-13.

Problem Symbols in schematic overlap each other.

If symbol pins overlap, there can be problems with pins being left unconnected. This shows up as red diamonds on the pins in the schematic. For information on correcting this situation, refer to the following sections:

- "Checking for Unconnected Nodes" on page 2-17
- "Using the -u2 Option" on page 3-8.
- "Modifying the Translator Configuration File" on page 3-22.

Problem User defined functions of the form below, are not supported by the Netlist Translator:.

```
real myfunc ( real a, real b ) {
  return a+b*2+sqrt)a*sin(b));
  }
```

Problem Statistical blocks of the form below, are not supported by the Netlist Translator:.

```
statistics {
  mismatch {
  }
}
```

Problem Expressions of the form below are translated straight across and are not supported by the Netlist Translator:

- `r=(p1 ? p2+1 : p3+1)`
- `c=(p1==p2)`

Note Expressions of the form `c=(p1==p2)` are supported with the "-pp" option for Spectre only.

Problem Statistical parameters of the form `%x` are not translated to the ADS "stat" format.

Chapter 7: Translating a Device

This chapter provides device translation tables for each of the devices supported by the Netlist Translator.

Recognizing Device Identifiers

Each SPICE device is distinguished by the first letter in it's device name. The following table describes the device type associated with each letter of the alphabet.

Table 7-1. SPICE Device Table

Device	Spice2	Spice3	PSpice	HSpice	ADS Translation
Axxxxx					Not translated
Bxxxxx		dep source	gaasfet		N type GaAsFET
Cxxxxx	capacitor	capacitor	capacitor	capacitor	Capacitor
Dxxxxx	diode	diode	diode	diode	Diode
Exxxxx	vcvs	vcvs	vcvs	vcvs	Voltage-Controlled Voltage Source
Fxxxxx	cccs	cccs	cccs	cccs	Current-Controlled Current Source
Gxxxxx	vccs	vccs	vccs	vccs	Voltage-Controlled Current Source
Hxxxxx	ccvs	ccvs	ccvs	ccvs	Current-Controlled Voltage Source
Ixxxxx	current	current	current	current	I_Source (ADS Netlist) ItUserDef (ADS Schematic)
Jxxxxx	jfet	jfet	jfet	jfet,mesfet	N or P type JFET
Kxxxxx	mutual	mutual	mutual	mutual	Mutual Inductor
Lxxxxx	inductor	inductor	inductor	inductor	Inductor
Mxxxxx	mosfet	mosfet	mosfet	mosfet	N or P type MOSFET, also with substrate
Nxxxxx			digital		Not translated
Oxxxxx		lossy tline	digital		Ideal Physical Transmission Line
Pxxxxx					Not translated
Qxxxxx	bjt	bjt	bjt	bjt	NPN/PNP BJT, VBIC
Rxxxxx	resistor	resistor	resistor	resistor	Resistor
Sxxxxx		vc switch	vc switch		Not translated
Txxxxx	tran line	tran line	tran line	tran line	Ideal 4-Terminal Transmission Line

Table 7-1. SPICE Device Table

Device	Spice2	Spice3	PSpice	HSpice	ADS Translation
Uxxxxx		rc line	digital src	lossy tline	2 Ideal Transformers, 1 Ideal DRC
Vxxxxx	voltage	voltage	voltage	voltage	V_Source (ADS Netlist) VtUserDef (ADS Schematic)
Wxxxxx		ic switch	ic switch		Not translated
Xxxxxx	subcircuit	subcircuit	subcircuit	subcircuit	subcircuit
Yxxxxx					Not translated
Zxxxxx		mesfet			N or P type MESFET

Documentation Conventions

To help you interpret the ADS and Spectre syntax given for a device, the following table describes the conventions used in this manual.

Table 7-2. Device Table Conventions

Description	Example
Keywords, parameter names and other literals are in bold.	**poly**
Parameters and names that are replaced with values by the user and are translated are in italics.	*n1*
Optional items are enclosed in square brackets.	[ic=x]
A choice between two or more items is displayed in square brackets separated by vertical bars.	[off | on]
Repeated optional items are enclosed in square brackets followed by an Asterisk.	[cn]*
Many parameters consist of name=value pairs; the left side is a keyword (literal text), the right side is typically a number, expression or variable (exceptions are noted).	ic=x
Simple keywords and values must appear in the given order, but name=value pairs may occur in any order.	

Example:

ctank2 *n1 n2* **capacitor** *c0* [*cn*]* [ic=*x*] [tc1=*y*] [tc2=*z*] [**off** | **on**] [param=*value*]*

Notes

- The example above is a capacitor with the name: *ctank2*
- The capacitor has two nodes: *n1* and *n2*

- The capacitor has a required keyword: **capacitor**
- The keyword is followed by a required value: *c0*
- The required value may be optionally followed by any number of values: [*cn*]*
- There are three stated optional name=value pairs which may appear in any order: [ic=*x*] [tc1=*y*] [tc2=*z*]
- There is one set of optional keywords of which one may appear but not both: [**off** | **on**]
- There other name=value pairs which may appear in any order: [param=*value*]*

Note When translating to the ADS Simulator, the node names, values and right hand sides of name=value pairs remain the same; however, the keywords may change to their ADS Simulator equivalents.

Using Parameter Mapping Tables to Understand a Translation

Device parameter information is organized in tables with the ADS parameter, unit, and default value in the three left-hand columns of the table. The parameter translation information is described in the three right- hand columns.

The parameters that are not translated are included in the tables. The parameters under *Parameters not in Spectre Model* are parameters that are supported in ADS and are not supported in Spectre. The parameters under *Parameters not in ADS Model* are parameters that are supported in Spectre and are not translated to ADS.

Refer to the Advanced Design System *Circuit Components* manual for an alphabetical listing of ADS supported devices and a detailed description of their parameters and default values.

Device Tables for Spectre

This section provides individual device translation information for each specific device supported by the Netlist Translator. The following is a list of supported devices:

Resistor Device

The Spectre resistor device is translated as an ADS resistor device. For information on the resistor model, refer to "R_Model:Resistor Model" on page 8-4.

Example Spectre Command Line:

Without model referenced:

```
r1 (1 2) resistor r=1.2k m=2
```

With model referenced:

```
r1 (1 2) resmod l=8u w=1u
```

Spectre Netlist Syntax:

Without model referenced:

name n1 n2 **resistor** [param=*value*]*

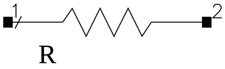
With model referenced:

name n1 n2 mname [param=*value*]*

ADS Netlist Syntax:

R:*rid n1 n2* **R=***value* [param=*value*]*

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-3. Resistor Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
R	ohms	50	r	ihms	
TC1	1/°C	0.0	tc1	1/°C	0.0
TC2	1/°C^2	0.0	tc2	1/°C^2	0.0
† Temp = trise + temp ("temp" is an ADS global variable)					

Table 7-3. Resistor Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Noise		yes	isnoisy		yes
Length	m		l	m	
Width	m		w	m	
_M		1	m		1
Parameters not in Spectre Model					
Temp[†]	°C	25			
Tnom	°C	25			
wPmax	W	infinity			
wImax	A	infinity			
Model					
Secured					
			Parameters not in ADS Model		
			trise	°C	0.0
			resform		yes
			scale		1

[†] Temp = trise + temp ("temp" is an ADS global variable)

Comments:

- The value of *R* may be positive or negative but not zero in the netlist.
- The value of *R* cannot be an expression that changes during the simulation.
- If *SCALE* is specified, *Value=Value*scale*.
- If *trise* is provided, *Temp=trise+temp* (*temp* is an ADS global variable).

Capacitor Device

The Spectre capacitor device is translated as an ADS capacitor device. For information on the capacitor model, refer to "C_Model:Capacitor Model" on page 8-6.

Example Spectre Command Line:

Without model referenced:

```
c1 (1 0) capacitor c=2.5u w=2u l=2.5u tc1=1e-8
```

With model referenced:

```
c1 (1 0) proc_cap c=2.5u w=2u l=2.5u tc1=1e-8
```

Spectre Netlist Syntax:

Without model referenced:

name n1 n2 mname [param=*value*]*

With model referenced:

name n1 n2 **capacitor** [param=*value*]*

ADS Netlist Syntax:

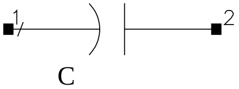
Without model referenced:

C:*cid n1 n2* **C=***value* [param=*value*]*

With model referenced:

*mname***:***cid n1 n2* [param=*value*]*

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-4. Capacitor Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
C	F		c	F	
TC1	1/°C		tc1	1/°C	0.0
TC2	1/°C^2		tc2	1/°C^2	0.0
InitCond	V	0.0	ic	V	0.0
Length	m		l	m	
Width	m		w	m	
_M		1	m		1
Parameters not in Spectre Model					
Temp†	°C	25			
Tnom	°C				
wBv	W	infinity			
Model					
			Parameters not in ADS Model		
			trise	°C	0.0
			scale		1
† Temp = trise + temp ("temp" is an ADS global variable)					

Comments:

- The value of *C* cannot be an expression that changes during the simulation.
- If *trise* is provided, *Temp=trise+temp* (*temp* is an ADS global variable).

Inductor Device

The Spectre mutual_inductor device is translated as an ADS mutual inductor.

Example Spectre Command Line:

Without model referenced:

```
l33 (0 net29) inductor l=10e-9 r=1 m=1
```

With model referenced:

```
l33 (0 net29) ind l=6e-9 r=1 tc1=1e-12 tc2=1e-12 tnom=25
```

Spectre Netlist Syntax:

Without model referenced:

name n1 n2 **inductor** [param=*value*]*

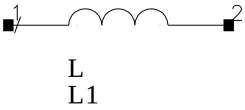
With model referenced:

name n1 n2 mname [param=*value*]*

ADS Netlist Syntax:

L:_lid n1 n2_ **L=**_value_

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to “Using Parameter Mapping Tables to Understand a Translation” on page 7-3

Table 7-5. Inductor Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
L	H	0.0	l	H	0.0
R	ohms	0.0	r	ohms	0.0
InitCond	A	0.0	ic	A	0.0

† Temp = trise + temp ("temp" is an ADS global variable)

Table 7-5. Inductor Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Noise		yes	isnoisy		yes
_M		1	m		1
Parameters not in Spectre Model					
Temp †	°C	25			
Tnom	°C				
TC1	1/°C				
TC2	1/°C^2				
			Parameters not in ADS Model		
			trise	°C	0.0
† Temp = trise + temp ("temp" is an ADS global variable)					

Comments:

- If *SCALE* is specified, *Value*=*Value***scale*.
- If *trise* is provided, *Temp*=*trise*+*temp* (*temp* is an ADS global variable).

Mutual Inductors

The Spectre mutual_inductor device is translated as an ADS mutual inductor.

Example Spectre Command Line:

```
l1 (1 0) inductor
l2 (2 0) inductor
ml1 mutual_inductor coupling=1 ind1=l1 ind2=l2
```

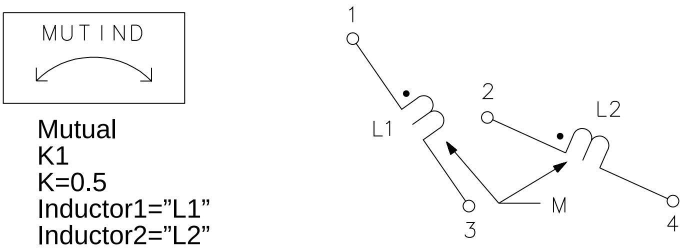
Spectre Netlist Syntax:

name **mutual_inductor** [param=*value*]*

ADS Netlist Syntax:

Mutual:*kid* **K=***value* **Inductor1="***lid1***" Inductor2="***lid2***"**

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-6. Mutual Inductors Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
K			coupling		0.0
Inductor1			ind1		
Inductor2			ind2		
Parameters not in Spectre Model					
M	H				

Comments:

- The ADS Schematic symbol for an individual inductor uses a slash on the wire near pin 1 as opposed to the standard dot used in Spectre.
- ADS Schematic Symbol: Note that the *MUTIND* on the left is the symbol placed when the *K* is read. The example on the right is an illustration of the referenced inductors.
- ADS allows a coefficient of coupling between -1 and 1.

Diode Device

The Spectre diode device is translated as an ADS diode device. For information on the diode model, refer to "Diode_Model:PN-Junction Diode Model" on page 8-10.

Example Spectre Command Line:

```
d0 (dp dn) pdiode l=3e-4 w=2.5e-4 area=1
```

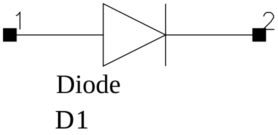
Spectre Netlist Syntax:

name n1 n2 mname [param=*value*]*

ADS Netlist Syntax:

mname:*did node1 node2* [param=*value*]*

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-7. Diode Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Area		1	area		
Periph		0	perim		
Length	m		l	m	1e-6
Width	m		w	m	1e-6
Region		on	region		on
_M		1	m		1
Parameters not in Spectre Model					
Temp †	°C	25			
Mode		nonlinear			

† Temp = trise + temp ("temp" is an ADS global variable)

Table 7-7. Diode Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Noise		yes			
Gd	Siemens				
Cd	F				
			Parameters not in ADS Model		
			trise	°C	
			scale		1

† Temp = trise + temp ("temp" is an ADS global variable)

Comments:

If *trise* is provided, *Temp=trise+temp* (*temp* is an ADS global variable).

BJT Device

The Spectre BJT device is translated as one of the following:

- a nonlinear NPN or PNP BJT device
- a nonlinear NPN or PNP BJT device with substrate
- an NPN device

For information on the BJT model, refer to "BJT_Model:Bipolar Transistor Model" on page 8-15.

Example Spectre Command Line:

```
q1 (vcc net3 minus) npn_mod region=fwd area=1 m=1
```

Spectre Netlist Syntax:

name nc nb ne [*ns*] *mname* [param=*value*]*

ADS Netlist Syntax:

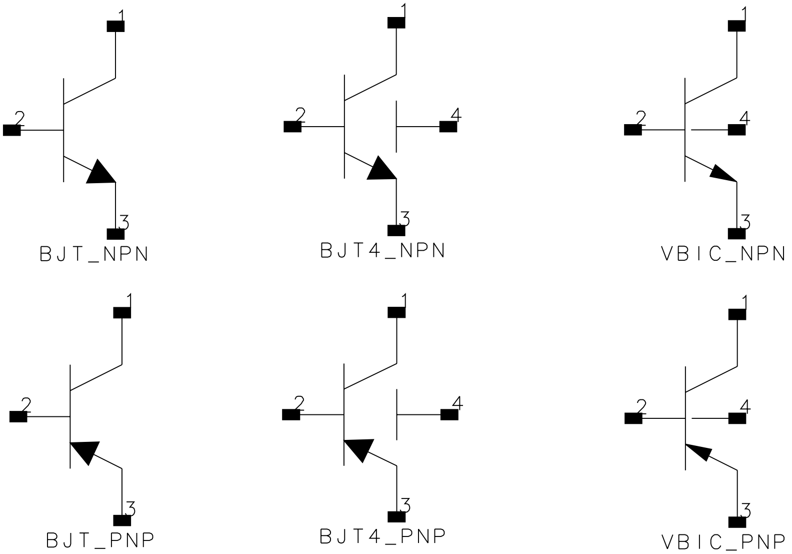
BJT[4]:

mname: *qid nc nb ne* [*ns*] [param=*value*]*

VBIC:

mname: *qid nc nb ne ns* [param=*value*]*

ADS Schematic Symbols:



Instance Parameters:

For information on parameter format, refer to “Using Parameter Mapping Tables to Understand a Translation” on page 7-3

Table 7-8. BJT Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Area		1.0	area		1.0
Region		0 (off)	region		fwd
NPN			(see *type* model parameter)		
PNP			(see *type* model parameter)		
_M		1	m		1
Parameters not in Spectre Model					
Temp [†]	°C	25			
Mode		on			

[†] Temp = trise + temp ("temp" is an ADS global variable)

Table 7-8. BJT Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Noise		on			
			Parameters not in ADS Model		
			trise	°C	0.0

† Temp = trise + temp ("temp" is an ADS global variable)

Table 7-9. VBIC Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Scale		1	area		1
Region		on	region		fwd
NPN		yes	(see *type* model parameter)		
PNP		no	(see *type* model parameter)		
_M		1	m		1
Parameters not in Spectre Model					
Temp †	°C	25			
Mode		nonlinear			
Noise		on			
			Parameters not in ADS Model		
			trise	°C	0.0

† Temp = trise + temp ("temp" is an ADS global variable)

Comments:

- The referenced “BJT_Model:Bipolar Transistor Model” on page 8-15 is read to determine which device to use:

  NPN keyword indicates BJT_NPN or BJT4_NPN
  PNP keyword indicates BJT_PNP or BJT4_PNP

- If *SCALE* is specified, *Value*=*Value***scale*.
- If *trise* is provided, *Temp*=*trise*+*temp* (*temp* is an ADS global variable).

MOSFET Device

The Spectre MOSFET device is translated as either a nonlinear N-type or a nonlinear P-type MOSFET device. For information on MOSFET models, refer to "MOSFET and JFET Models" on page 8-36.

Example Spectre Command Line:

```
nch1 (1 2 0 0) nchmod1 l=2u w=15u ad=60p as=37.5p pd=23u ps=6u
```

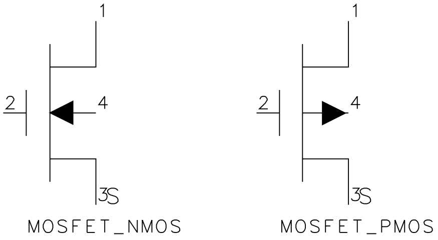
Spectre Netlist Syntax:

name nd ng ns nb mname [param=*value*]*

ADS Netlist Syntax:

mname:**m**_id nd ng ns nb_ [param=*value*]*

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-10. Level 1 MOSFET Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Length	m		l	m	
Width	m		w	m	
Ad	m^2		ad	m^2	
As	m^2		as	m^2	
Pd	m		pd	m	

† Temp = trise + temp ("temp" is an ADS global variable)

Table 7-10. Level 1 MOSFET Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Ps	m		ps	m	
Nrd			nrd	m/m	
Nrs			nrs	m/m	
Region		on	region		triode
_M		1	m		1
Parameters not in Spectre Model					
Temp †	°C	25			
NMOS		yes			
PMOS		no			
Mode		1			
Noise		yes			
			Parameters not in ADS Model		
			trise	°C	0.0
			ld	m	
			ls	m	
			degradation		no
† Temp = trise + temp ("temp" is an ADS global variable)					

Table 7-11. BSIM3v3 Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Length	m		l	m	
Width	m		w	m	
Ad	m^2		ad	m^2	
As	m^2		as	m^2	
Pd	m		pd	m	
Ps	m		ps	m	
Nrd			nrd	m/m	
† Temp = trise + temp ("temp" is an ADS global variable)					

Table 7-11. BSIM3v3 Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Nrs			nrs	m/m	
Region		fwd	region		triode
Nqsmod			nqsmod		
_M		1	m		1
Parameters not in Spectre Model					
Geo		1			
NMOS		yes			
PMOS		no			
Mode		1			
Noise		yes			
Temp [†]	°C	25			
			Parameters not in ADS Model		
			trise	°C	

[†] Temp = trise + temp ("temp" is an ADS global variable)

Table 7-12. Model9 MOSFET Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Length	m	1e–4	l	m	1.0 scale
Width	m	1e–4	w	m	1.0 scale
Region		on	region		triode
_M		1	m		1
Mult		1	mult		1
Mult		1	area		1
Parameters not in Spectre Model					
Model					
Temp [†]	°C	25			
Ab	m^2	1e–12			

[†] Temp = trise + temp ("temp" is an ADS global variable)

Table 7-12. Model9 MOSFET Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Ls	m	1e–4			
Lg	m	1e–4			
Mode		nonlinear			
			Parameters not in ADS Model		
			mult		1

† Temp = trise + temp ("temp" is an ADS global variable)

Comments:

- The Netlist Translator looks up the model statement referenced by the model name on the instance line to determine which ADS MOSFET device to place. For more information on MOSFET Models, refer to "MOSFET and JFET Models" on page 8-36.
- If *trise* is provided, *Temp*=*trise*+*temp* (*temp* is an ADS global variable).

JFET Device

The Spectre JFET device is translated as either a nonlinear N-type or a nonlinear P-type JFET device. For information on the JFET model, refer to "JFET_Model:Junction Field Effect Transistor Model" on page 8-51.

Example Spectre Command Line:

```
jk1 (net1 net2 0) jmod area=1
```

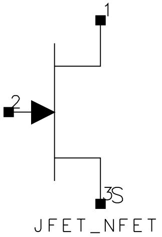
Spectre Netlist Syntax:

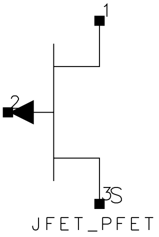
name nd ng ns [*nb*] *mname* [param=*value*]*

ADS Netlist Syntax:

vcvs:name p n ps ns [param=*value*]*

ADS Schematic Symbol:





Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-13. JFET Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Length	m		l	m	
Width	m		w	m	
Ad	m^2		ad	m^2	
As	m^2		as	m^2	
Pd	m		pd	m	
† Temp = trise + temp ("temp" is an ADS global variable)					

Table 7-13. JFET Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Ps	m		ps	m	
Nrd			nrd	m/m	
Nrs			nrs	m/m	
Region		on	region		triode
_M		1	m		1
Parameters not in Spectre Model					
Temp †	°C	25			
NMOS		yes			
PMOS		no			
Mode		1			
Noise		yes			
			Parameters not in ADS Model		
			trise	°C	0.0
			ld	m	
			ls	m	
			degradation		no

† Temp = trise + temp ("temp" is an ADS global variable)

Comments:

- The referenced "JFET_Model:Junction Field Effect Transistor Model" on page 8-51 model is read to determine which JFET device to use:

  PJF keyword indicates JFET_PFET

  NJF keyword indicates JFET_NFET

- The 4-port JFET is not available in ADS. The device will be skipped.
- If *SCALE* is specified, *Value=Value*scale*.
- If *trise* is provided, *Temp=trise+temp* (*temp* is an ADS global variable).
- The following syntax bay be ambiguous to the translator if the referenced model is not in the translated files:

```
Jid nd ng ns var1 var2
```

The translator is unable to determine which of the following should be used:

```
Jid nd ng ns na modelname
```

OR

```
Jid nd ng ns modelname area
```

Since the 4 port device is not available anyway, it will assume that modelname and area are specified. A warning message will be reported in the log file.

Linear Voltage-Controlled Voltage Source

The Spectre vcvs device is translated as an ADS VCVS device.

Example Spectre Command Line:

```
e1 (out1 0 pos neg) vcvs gain=10
```

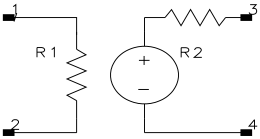
Spectre Netlist Syntax:

name p n ps ns **vcvs** [param=*value*]*

ADS Netlist Syntax:

VCVS:*name node1 node2 node3 node4* [param=*value*]*

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-14. Linear VCVS Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
G	V/V	0.0	gain	V/V	0.0
Parameters not in Spectre Model					
R1	Ohms	0.0 (Infinity)			
R2	Ohms	0.0			
F	Hz	0.0 (Infinity)			
			Parameters not in ADS Model		
			m		1

Linear Voltage-Controlled Current Source

The Spectre vccs device is translated as an ADS VCCS device.

Example Spectre Command Line:

```
v1 (1 0 2 3) vccs gm=-1 m=2
```

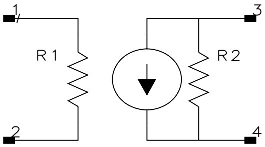
Spectre Netlist Syntax:

name sink arc ps ns **vccs** [param=*value*]*

ADS Netlist Syntax:

VCCS:*name node1 node2 node3 node4* [param=*value*]*

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-15. Linear VCCS Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
G	Siemens	0.0	gm	Siemens	0.0
Parameters not in Spectre Model					
R1	Ohms	0.0 (Infinity)			
R2	Ohms	0.0 (Infinity)			
F	Hz	0.0 (Infinity)			
			Parameters not in ADS Model		
			m		1

Non-Linear Voltage-Controlled Voltage Source

The Spectre pvcvs device is translated as an ADS NonlinVCVS device.

Example Spectre Command Line:

```
v1 (p 0 c1 0) pvcvs coeffs=[0 0 0 0.1 1 1] gain=1
```

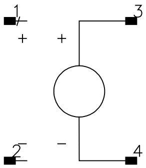
Spectre Netlist Syntax:

name p n ps1 ns1 ... **pvcvs coeffs=[**...**]** [param=*value*]*

ADS Netlist Syntax:

VCVS_Z:*name node1 node2 node3 node4* [param=*value*]*

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-16. Non-Linear VCVS Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Coeff=[...]			coeffs=[...]		
			Parameters not in ADS Model		
			gain		1
			m		1

Non-Linear Voltage-Controlled Current Source

The Spectre pvccs device is translated as an ADS NonlinVCCS device.

Example Spectre Command Line:

```
v2 (net1 0 net2 0) pvccs coeffs=[0 -2e-3 -10e-3] gain=2 m=1
```

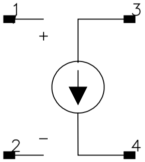
Spectre Netlist Syntax:

name sink arc ps1 ns1 ... **pvccs coeffs=[...]** [param=*value*]*

ADS Netlist Syntax:

VCCS_Z:*name node1 node2 node3 node4* [param=*value*]*

ADS Schematic Symbol:



Instance Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 7-3

Table 7-17. Non-Linear VCCS Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Coeff=[...]			coeffs=[...]		
			Parameters not in ADS Model		
			gain		1
			m		1

Subcircuit Reference

Subcircuit Reference: This device is translated as an instance of a subnetwork. For information on the parametric subnetwork command, refer to "subckt, ends" on page 10-5.

Example Spectre Netlist:

```
inline subckt indline ( in out gnd )
parameters
+ hoverw=pmsxt/pw
+ fhwl=1/((1/hoverw)+2.42-(0.44*hoverw)+1-hoverw)**6)
+ fhwg = d1==(2*pi())*log((8*hoverw)+(0.25/hoverw))
+ fhw = (hoverw==1)*fhwl+(hoverw>=1)*fhwg
rsxout subout gnd resistor r=prsx*2 tc1=trsub trise=dtemp
ends indline
```

ADS Netlist Syntax:

SUBNAME:x*id* [*n1**] [*name*=*value**] [_**M=***mult*]

ADS Instance Parameters:

xid = Subnetwork element name

n1* = Node names

subname = Specifies the name of the subnetwork.

name = Parameter name or subcircuit definition.

value = A parameter value passed to the subnetwork.

mult = Multiplier used to simulate multiple parallel devices.

Comments:

The ADS netlist format does not support subcircuit that contain parameters that reference each other. Any Spectre subcircuit equations that reference each other are moved from the subcircuit parameter list and are listed as subcircuit equations. In addition, a warning message will be generated as follows:

```
The subcircuit “ckt_name” contained parameters that reference each
other. In order to maintain compatibility with ADS, these have been
moved off the subcircuit parameters list and are treated as circuit
equations. Component instances that reference this subcircuit will
not be able to directly override these values.
```

Device Tables for SPICE

This section provides individual device translation information for each specific device supported by the Netlist Translator. Supported devices are listed here.

For information on unsupported devices, refer to “Unsupported Devices” on page 7-75.

Bxxxxxxx

Semiconductor GaAsFET Device: This device is translated as a nonlinear N-type GaAsFET device. For information on the GaAsFET model, refer to “GaAsFET and JFET Models for SPICE” on page 8-94.

Example SPICE Command Line:

```
b1 1 2 3 Bmodel 2
```

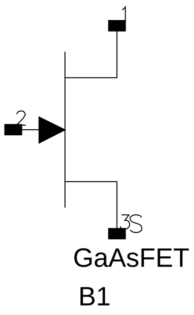
SPICE dialect and netlist syntax:

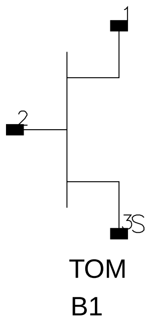
Spice2/3:	Non-linear dependent source - not translated
PSpice:	**b***id nd ng ns mname* [**AREA**]
HSpice:	Does not exist

ADS Netlist Syntax:

mname:**b***id nd ng ns* **Area=**area

ADS Schematic Symbols:





Instance Parameters:

bid	=	GaAsFET element name
mname	=	Model name
nd	=	Drain node
ng	=	Gate node

ns = Source node

area = Area factor

Cxxxxxxx

Capacitor: This device is translated as a capacitor. For information on the capacitor model, refer to "C_Model:Capacitor Model" on page 8-6.

Example SPICE Command Line:

```
c1 1 2 1pF
```

SPICE dialect and netlist syntax:

Spice2/3: **c***id n1 n2* [*value* | *mname*] [**l=***l*] [**w=***w*] [**ic=***ic*]

PSpice: **c***id n1 n2* [*mname*] *value* [**ic=***ic*]

HSpice: **c***id n1 n2* [*mname*] [**c=**]*value* [[**tc1=**]*tc1* [[**tc2=**]*tc2*]]
+ [**scale=***scale*] [**ic=***ic*] [**m=***mult*] [**w=***w*] [**l=***l*] [**dtemp=***dtemp*]

ADS Netlist Syntax:

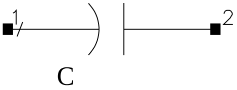
Without model referenced:

C:c*id n1 n2* **C=***value* [**_M=***mult*]

With model referenced:

*mname***:c***id n1 n2* [**C=***value*] [**Length=***l*] [**Width=***w*] [**InitCond=***ic*]
+ [**TC1=***tc1*] [**TC2=***tc2*] [**Temp=temp+(***dtemp)*] [**_M=***mult*]

ADS Schematic Symbol:



Instance Parameters:

cid = Capacitor element name

mname = Model name

n1 = Positive node

n2 = Negative node

value = Capacitance in farads

l = Length in meters

w = Width in meters

ic = The optional initial condition (time-zero) value of capacitor voltage (in volts). The initial condition (if any) applies only if the UIC option is specified on the .TRAN control line.

tc1 = Temperature coefficient per degrees celsius.

tc2 = Temperature coefficient per degrees celsius squared.

scale = Element scale factor.

mult = Multiplier used to simulate multiple parallel devices.

dtemp = Element temperature difference with respect to circuit temperature.

Comments:

HSpice: If the value of C is an expression which is a function of node voltages or independent variables, then the component is a dependent capacitor. In ADS, the component is represented by a Symbolically Defined Device (SDD).

If one of the SDD parameters uses a variable containing node voltages, the translator needs to look up the node voltages on the resultant SDD to convert them to the correct ADS syntax and update the variable expression. For instance, the node voltage described as v(nodeX) in HSpice must be converted to _v1, if the first pair of nodes on the SDD is (node1, 0).

For translation purposes of this special case, the SDD and the variable used by it are expected to be defined in the same subcircuit, with the variable being used by only one SDD. If the variable were defined globally or used by different SDDs, the node voltage variables (_v1, etc.) might not match up correctly. For example, in the case described above one SDD might have (nodeX,0) as the first pair of nodes (_v1), while another SDD might have (nodeX,0) as the 2nd pair of nodes (_v2). The translator would not be able to replace v(nodeX) in the variable expression to define both SDDs correctly at the same time.

If this SDD is not found in the same subcircuit as the variable expression, a warning message will be written to the log file *spctoiff.log*. Also, the expression will not be converted automatically, and it will have to be fixed manually before simulation is attempted.

If SCALE is specified, Value=Value*scale

If dtemp is provided, Temp=temp+dtemp

Translator Limitations: Functionality is available in ADS but not yet supported by the Netlist Translator. Workarounds are provided below.

HSpice:

1) Nonlinear capacitor cid n1 n2 poly c0 c1 c2…

This is supported by ADS as a component called NonlinC. The ADS Netlist Syntax should be:

```
NonlinC:cid n1 n2 coef=list(c0 c1 c2…) [ic=ic]
```

2) cid n1 n2 c='equation' ctype=[0|1]

This syntax requires a Symbolically Defined Device (SDD) in ADS if ctype=0. This is not currently handled by the translator. Please contact customer support for assistance in writing the SDD to represent the voltage across the capacitor.

The Netlist Translator fails to recognize the capacitor model in the following syntax:

```
c1 1 2 cmodel
```

The translator assumes that it is a variable name for the capacitance value. To correct your imported schematic, manually move the model name to the parameter Model. To correct the ADS Netlist, use the following netlist syntax:

```
cmodel:c1 1 2 [param=value]*
```

Dxxxxxxx

Semiconductor Diode: This device is translated as a nonlinear diode device. For information on the diode model, refer to "BJT_Model:Bipolar Transistor Model" on page 8-15.

Example SPICE Command Line:

```
d1 1 2 Dmodel 2 OFF
```

SPICE dialect and netlist syntax:

Spice2/3: **d***id n1 n2 mname* [*area*] [**off**] [**ic=***vd*] [**temp=***temp*]

PSpice: **d***id n1 n2 mname* [**AREA**]

HSpice: **d***id n1 n2 mname* [[**AREA=***area*] [**PJ=***periph*]] | [**W=***wval* **L=***lval*]]
+ [**off**] [**dtemp=***dtemp*] [**m=***mult*] [**ic=***vd*] [**pj=***x*] [**wp=***x*] [**lp=***x*]
+ [**wm=***x*] [**lm=***x*]

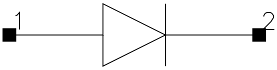
OR

d*id n1 n2 mname* [*area* [*periph*]] [**OFF**] [**IC=***vd*] [**M=***val*]

ADS Netlist Syntax:

mname:**d***id node1 node2* [[**Area=***area*] [**Periph=***periph*] |
+ [Width=wval] [Length=lval]] [**Temp=***temp* | [**temp**+*dtemp*]] [_**M=***mult*]

ADS Schematic Symbol:



Diode
D1

Instance Parameters:

did = Diode element name

mname = Model name

n1 = Positive node

n2 = Negative node

area = Area factor. If area is not specified and W and L are, area is calculated as lval*wval.

periph = Scaling factor that affects the sidewall. If PJ or periph is not specified, and W and L are, Periph is calculated as 2*(wval + lval).

Width = Geometric width of diode junction (meters).

Length = Geometric length of diode junction (meters).

temp = The temperature at which this device is to operate and overrides the temperature specification on the .OPTIONS control line.

OFF = Initial condition on the device for dc analysis. Translates to Region=0.

mult = Multiplier used to simulate multiple parallel devices.

Exxxxxxx

Linear Voltage-Controlled Voltage Source: This device is translated as a Symbolically Defined Device. For information on SDD's, refer to "Using a Symbolically Defined Device" on page 7-76.

Example SPICE Command Line:

```
e1 1 2 3 4 2.0
```

SPICE dialect and netlist syntax:

Spice2/3: **e***id n1 n2 cn1 cn2 value*

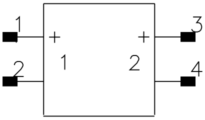
PSpice: **e***id n1 n2 cn1 cn2 value*
e*id n1 n2* **poly(***1***)** *cn1 cn2 value*
nonlinear: **e***id n1 n2* **poly(***n***)...**

HSpice: **e***id n1 n2 cn1 cn2 value*
e*id n1 n2* **poly(***1***)** *cn1 cn2 value*
nonlinear: **e***id n1 n2* **poly(***n***)...**

ADS Netlist Syntax:

SDD:e*id cn1 cn2 n1 n2* I[1,0]=0 F[2,0]=_v2-(*value**_v1)

ADS Schematic Symbol:



SDD2P

Instance Parameters:

eid	=	Source element name
n1	=	Positive Node
n2	=	Negative Node
cn1	=	Positive Controlling Node
cn2	=	Negative Controlling Node
value	=	Voltage gain
poly()	=	Polynomial function

Comments:

The following syntax extensions are not supported by the translator. If these keywords or parameters are found, component translation will fail and a warning message will be written to the translation log file.

PSpice:

TABLE, LAPLACE, FREQ, CHEBYSHEV

HSpice Keywords:

PWL, AND, NAND, OR, NOR, DELAY, OPAMP, TRANSFORMER

HSpice Parameters:

MAX, MIN, SCALE, TC1, TC2, ABS, DELTA, IC, NPDELAY

Fxxxxxxx

Linear Current-Controlled Current Source: This device is translated as a Symbolically Defined Device. For information on SDD's, refer to "Using a Symbolically Defined Device" on page 7-76.

Example SPICE Command Line:

```
f1 vout 0 vname 0.05
```

SPICE dialect and netlist syntax:

Spice2/3: **f***id n1 n2 sourceName value*

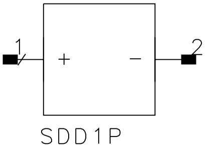
PSpice: **f***id n1 n2 sourceName value*

HSpice: **f***id n1 n2 sourceName value*

ADS Netlist Syntax:

SDD:f*id n1 n2* I[1,0]=*value**_c1 C[1]="*sourceName*"

ADS Schematic Symbol:



Instance Parameters:

fid	=	Source element name
n1	=	Positive Node
n2	=	Negative Node
value	=	Transconductance (in mhos).
sourceName	=	Name of controlling source

Comments:

The following syntax extensions are not supported by the translator. If these keywords or parameters are found, component translation will fail and a warning message will be written to the translation log file.

HSpice Keywords:

PWL, AND, NAND, OR, NOR, DELAY

HSpice Parameters:

MAX, MIN, SCALE, M, TC1, TC2, ABS, DELTA, IC, NPDELAY

Gxxxxxxx

Linear Voltage-Controlled Current Source: This device is translated as a Symbolically Defined Device. For information on SDD's, refer to "Using a Symbolically Defined Device" on page 7-76.

Example SPICE Command Line:

```
g1 1 2 3 4 .1mMho
```

SPICE dialect and netlist syntax:

Spice2/3: **g***id n1 n2 cn1 cn2 value*

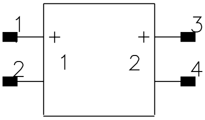
PSpice: **g***id n1 n2 cn1 cn2 value*
g*id n1 n2* **poly(***1***)** *cn1 cn2 value*
nonlinear: **g***id n1 n2* **poly(***n***)...**

HSpice: **g***id n1 n2 cn1 cn2 value*
g*id n1 n2* **poly(***1***)** *cn1 cn2 value*
nonlinear: **g***id n1 n2* **poly(***n***)...**

ADS Netlist Syntax:

SDD:g*id cn1 cn2 n1 n2* I[1,0]=0 F[2,0]=_v2-(*value**_v1)

ADS Schematic Symbol:



SDD2P

Instance Parameters:

gid	=	Source element name
n1	=	Positive Node
n2	=	Negative Node
cn1	=	Positive Controlling Node
cn2	=	Negative Controlling Node
value	=	Transconductance (in mhos).
poly()	=	Polynomial function

Comments:

The following syntax extensions are not supported by the translator. If these keywords or parameters are found, component translation will fail and a warning message will be written to the translation log file.

PSpice

TABLE, LAPLACE, FREQ, CHEBYSHEV

HSpice Keywords:

PWL, AND, NAND, OR, NOR, DELAY, VCR, VCCAP

HSpice Parameters:

MAX, MIN, SCALE, M, TC1, TC2, ABS, DELTA, SMOOTH, IC, NPDELAY

Hxxxxxxx

Linear Current-Controlled Voltage Source: This device is translated as a Symbolically Defined Device. For information on SDD's, refer to "Using a Symbolically Defined Device" on page 7-76.

Example SPICE Command Line:

```
h1 vout 0 vname 0.05
```

SPICE dialect and netlist syntax:

Spice2/3: **h***id n1 n2 sourceName value*

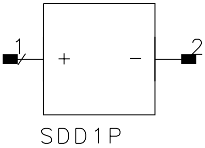
PSpice: **h***id n1 n2 sourceName value*
h*id n1 n2* **poly(***1***)** *sourceName value*
nonlinear: **h***id n1 n2* **poly(***n***)...**

HSpice: **h***id n1 n2 sourceName value*
h*id n1 n2* **poly(***1***)** *sourceName value*
nonlinear: **h***id n1 n2* **poly(***n***)...**

ADS Netlist Syntax:

SDD:h*id n1 n2* F[1,0]=-(*value**_c1)+_v1 C[1]="*sourceName*"

ADS Schematic Symbol:



Instance Parameters:

hid	=	Source element name
n1	=	Positive Node
n2	=	Negative Node
value	=	Transconductance (in mhos).
sourceName	=	Name of controlling source
poly()	=	Polynomial function

Comments:

The following syntax extensions are not supported by the translator. If these keywords or parameters are found, component translation will fail and a warning message will be written to the translation log file.

HSpice Keywords:

PWL, NPWL, PPWL, AND, NAND, OR, NOR, DELAY

HSpice Parameters:

MAX, MIN, SCALE, TC1, TC2, ABS, DELTA, IC, NPDELAY

Ixxxxxxx

Independent Current Source: This device is translated as either an I_Source (ADS Netlist) or an ItUserDef (ADS Schematic).

Example SPICE Command Line:

```
i1 1 2 DC 6
i2 2 3 AC 1 90
i3 3 0 PULSE(-1 1 2NS 2NS 2NS 50NS 100NS)
i4 3 0 SIN(0 1 100MEG 1NS 1E10)
i5 3 0 EXP(-4 -1 2NS 30NS 60NS 40NS)
i6 7 5 PWL(0 -7 10NS -7 11NS -3 17NS -3 18NS -7 50NS -7)
i7 12 0 SFFM(0 1M 20K 5 1K)
```

SPICE dialect and netlist syntax:

Spice2/3: **i***id n+ n-* [[**dc**] *dcval*] [**ac** [*mag*[*phase*]]] [*tranfunc*]

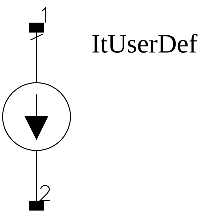
PSpice: **i***id n+ n-* [[**dc**] *dcval*] [**ac** [*mag*[*phase*]]] [*tranfunc*]

HSpice: **i***id n+ n-* [[**dc**] *dcval*] [**ac** [*mag*[*phase*]]] [*tranfunc*]

ADS Netlist Syntax:

I_Source:i*id n+ n-* **Idc=***dcval* **Iac=polar(***mag,phase***)** **I_Tran=***tranfunc*

ADS Schematic Symbol:



Instance Parameters:

iid = GaAsFET element name

n+ = Positive node

n- = Negative node

dcval = The dc current

mag, phase = The magnitude and phase of the ac current.

tranfunc = Transient Source Functions. There are several built in ADS functions that mimic the SPICE transient source functions listed below.

SPICE	**Advanced Design System**
EXP()	exp_pulse(time, low, high, tdelay1, tau1, tdelay2, tau2)
PULSE()	pulse(time, low, high, delay, rise, fall, width, period)
PWL()	pwl(time, t1, x1,..., tn, xn)
SFFM()	sffm(time, offset, amplitude, carrier_freq, mod_index, signal_freq)
SIN()	damped_sin(time, offset, amplitude, freq, delay, damping)
HSpice	
PU()	pulse(time, low, high, delay, rise, fall, width, period)
AM()	Not translated.
PL()	Not translated.

Comments:

HSpice:

HSpice multiplier on Current sources - This is implemented by making an expression out of the given current value. Example:

```
HSpice: i1 n1 n2 500ua m=10
ADS: eqn0=(500ua)*(10)
I_Source: i1 n1 n2 Idc=500ua*10
```

Unsupported HSpice transient functions - AM, PL

Unsupported HSpice syntax extensions -

Data driven PWL source

Repeat and Time Delay specifications in PWL transient function (R=repeat, TD=delay)

PSpice:

Unsupported PSpice syntax extensions

STIMULUS

corner points, TIME_SCALE_FACTOR, VALUE_SCALE_FACTOR

Jxxxxxxx

Semiconductor JFET Device: This device is translated as either a nonlinear N-type or a nonlinear P-type JFET device. For information on the JFET model, refer to "JFET_Model:Junction Field Effect Transistor Model" on page 8-51.

Example SPICE Command Line:

```
j1 1 2 3 Jmodel 2 OFF IC=0.6, 5.0
```

SPICE dialect and netlist syntax:

Spice2/3: **j***id nd ng ns mname* [*area*] [**off**] [**ic=***vds*, *vgs*] [**temp=***x*]

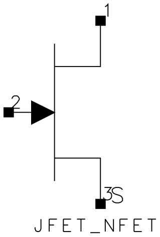
PSpice: **j***id nd ng ns mname* [*area*]

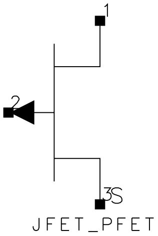
HSpice: **j***id nd ng ns* [*nb*] *mname* [*area* | **w=***w* **l=***l*] [**off**] [**m=***mult*]
+ [**ic=***vds*, *vgs*] [**vds=***vds*] [**vgs=***vgs*] [**dtemp=***dtemp*]

ADS Netlist Syntax:

mname:**j***id nd ng ns* [**Area=***area*] [**Temp=temp**+*dtemp*] [**Region=**0]
+ [**_M=***mult*]

ADS Schematic Symbol:





Instance Parameters:

jid	=	JFET element name
mname	=	Model name
nd	=	Drain node
ng	=	Gate node
ns	=	Source node
nb	=	optional bulk node - not supported.

area = Area factor

off = Initial condition on the device for dc analysis. Translates to Region=0.

mult = Multiplier used to simulate multiple parallel devices.

temp = Temperature at which the device is to operate.

Comments:

The referenced “JFET_Model:Junction Field Effect Transistor Model” on page 8-51 model is read to determine which JFET device to use:

PJF keyword indicates JFET_PFET
NJF keyword indicates JFET_NFET

The 4-port JFET is not available in ADS. The device will be skipped.

If SCALE is specified, Value=Value*scale

If dtemp is provided, Temp=temp+dtemp

The following syntax bay be ambiguous to the translator if the referenced model is not in the translated files:

```
Jid nd ng ns var1 var2
```

The translator is unable to determine which of the following should be used:

```
Jid nd ng ns na modelname
```

or:

```
Jid nd ng ns modelname area
```

Since the 4 port device is not available anyway, it will assume that modelname and area are specified. A warning message will be reported in the log file.

Kxxxxxxx

Coupled (Mutual) Inductors: This device is translated as a mutual inductor.

Example SPICE Command Line:

```
k1 L1 L2 .5
L1 1 3 10
L2 2 4 20
```

SPICE dialect and netlist syntax:

Spice2/3: **k**_id lid1 lid2 value_

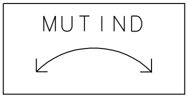
PSpice: **k**_id lid1 lid2_***** _value_

HSpice: **k**_id lid1 lid2_ [**K=**]_value_

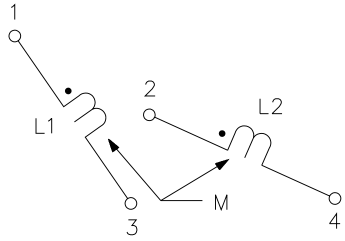
ADS Netlist Syntax:

Mutual:k_id_ **K=**_value_ **Inductor1="**_lid1_**"** **Inductor2="**_lid2_**"**

ADS Schematic Symbol:



Mutual
K1
K=0.5
Inductor1="L1"
Inductor2="L2"



Instance Parameters:

kid = Mutual inductors element name

lid1 = Name of the first coupled inductor.

lid2 = Name of the second coupled inductor.

value = The coefficient of coupling, K, which must be greater than 0 and less than or equal to 1.

Comments:

The ADS Schematic symbol for an individual inductor uses a slash on the wire near pin 1 as opposed to the standard dot used in SPICE. Note that the MUTIND on the left is the symbol placed when the K is read. The example on the right is an illustration of the referenced inductors.

For value, HSpice and ADS allows a coefficient of coupling between -1 and 1.

* PSpice supports a mutual inductor syntax that allows any number of inductors to be coupled together with the same coupling coefficient. This needs to be converted to a set of mutual inductors representing every possible combination of inductors from the list. The translator does not support this at this time.

Unsupported Syntax and Parameters

PSpice

kid lid1 coupling value model [size]
kid T1name T2name Cm=cval Lm=1val

HSpice

HSpice model syntax
HSpice MAG parameter

Lxxxxxxx

Inductor: This device is translated as an inductor.

Example SPICE Command Line:

```
l1 1 2 1pH IC=15mA
```

SPICE dialect and netlist syntax:

Spice2/3: **l***id n1 n2 value* [**ic=***icamp*]

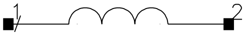
PSpice: **l***id n1 n2 value* [**ic=***icamp*]

HSpice: **l***id n1 n2* [**l=***value*] [[**tc1=**]*tc1* [[**tc2=**]*tc2*]] [**scale=***scale*]
+ [**ic=***ic*] [**m=***mult*] [**dtemp=***dtemp*] [**r=***r*]

ADS Netlist Syntax:

L:l*id n1 n2* **L=***value*

ADS Schematic Symbol:



L
L1

Instance Parameters:

lid	=	Inductor element name
n1	=	Positive node
n2	=	Negative node
value	=	The inductance in Henries
ic	=	The initial (time-zero) value of inductor current (in Amps)
tc1	=	First order temperature coefficient
tc2	=	Second order temperature coefficient
scale	=	Element scale factor. Default=1.0.
mult	=	Multiplier used to simulate multiple parallel devices. Default=1.0
dtemp	=	Element and circuit temperature difference. Default=0.0
R	=	Resistance in ohms of the inductor element.

equation	=	The inductor value can be described as a function of any node voltages, branch currents, and any independent variables such as TIME, frequency (HERTZ), or temperature (TEMPER). The type of variable L depends upon is indicated by the parameter “LTYPE”. Most commonly L depends upon I(Lxxx), which is assumed with the default of LTYPE=0 as explained below.
LTYPE	=	If inductance L is a function of I(Lxxx), set LTYPE to 0. Otherwise, set LTYPE to 1. The inductance flux is calculated differently depending on the value of LTYPE. LTYPE must be set properly to provide correct simulation results. Defining L as a function of multiple variables is not recommended. Default=0.
poly()	=	Polynomial function

Comments:

If SCALE is specified, Value=Value*scale

Unsupported Syntax and Parameters

PSpice:

PSpice Inductor model is supported by ADS but not translated yet.
lid n1 n2 [mname] value

HSpice:

Nonlinear inductor lid n1 n2 poly l0 l1 l2…
This is supported by ADS as a component called NonlinL.
The ADS Netlist Syntax should be:
NonlinL:lid n1 n2 coef=list(l0 l1 l2…) [ic=ic]

lid n1 n2 L=’equation’ ltype=[0 | 1] [R=val]
This syntax requires a Symbolically Defined Device (SDD) in ADS if ltype=0. This is not currently handled by the translator. Please contact customer support for assistance in writing the SDD to represent the current through the inductor.

Mxxxxxxx

Semiconductor MOSFET Device: This device is translated as either a nonlinear N-type or a nonlinear P-type MOSFET device.

Example SPICE Command Line:

```
m1 1 2 3 4 Mmodel L=10U W=5U AD=100P AS=100P PD=40U NRD=1 nrs=1
```

SPICE dialect and netlist syntax:

Spice2/3: **m***id nd ng ns nb mname* [**l=***l*] [**w=***w*] [**ad=***ad*] [**as=***as*]
+ [**pd=***pd*] [**ps=***ps*] [**nrd=***nrd*] [**nrs=***nrs*] [**off**]
+ [**ic=***vds, vgs, vbs*] [**temp=***temp*]

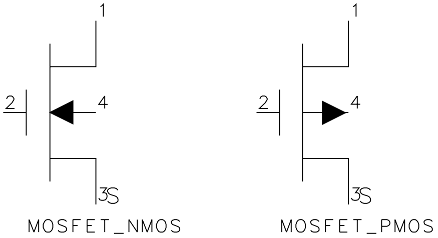
PSpice: **m***id nd ng ns nb mname* [**l=***l*] [**w=***w*] [**ad=***ad*] [**as=***as*]
+ [**pd=***pd*] [**ps=***ps*] [**nrd=***nrd*] [**nrs=***nrs*] [**nrg=***nrg*]
+ [**nrb=***nrb*] [**m=***mult*]

HSpice: **m***id nd ng ns nb mname* [[**l=**]*l*] [[**w=**]*w*] [**ad=***ad*] [**as=***as*]
+ [**pd=***pd*] [**ps=***ps*] [**nrd=***nrd*] [**nrs=***nrs*] [**off**]
+ [**rdc=***rdc*] [**rsc=***rsc*] [**ic=***vds, vgs, vbs*]
+ [**m=***mult*] [**dtemp=d***temp*] [**geo=***geo*] [**nqsmod=***nqsmod*]

ADS Netlist Syntax:

*mname***:m***id nd ng ns nb* [**Length=***l*] [**Width=***w*] [**Ad=***ad*] [**As=***as*] [**Pd=***pd*] [**Ps=***ps*]
+ [**Nrd=***nrd*] [**Nrs=***nrs*] [**_M=***m*ult] [**Region=**0] [**Temp=***temp*/[*temp*+*dtemp*]]
+ [**Geo=***geo*] [**Nqsmod=***nqsmod*]

ADS Schematic Symbol:



Instance Parameters:

mid = MOSFET element name

mname = Model name

nd = Drain node

ng = Gate node

ns = Source node

nb = Bulk (substrate) node

l, w = Channel length and width in meters.

ad, as = Area of the drain and source diffusions, in square meters.

pd, ps = Perimeters of the drain and source junctions, in meters.

nrd, nrs = Designate the equivalent number of squares of the drain and source diffusions; these values multiply the sheet resistance (RSH) specified on the .MODEL control line for an accurate representation of the parasitic series drain and source resistance of each transistor.

off = Indicates an initial condition on the device for ac analysis. Translates to Region=0.

temp = The temperature at which this device is to operate.

dtemp = Element and circuit temperature difference. Default=0.0

mult = Multiplier used to simulate multiple parallel devices.

geo = Source/drain sharing selector

nqsmod = Non-quasi static model selector

Comments:

The Netlist Translator looks up the model statement referenced by the model name on the instance line to determine which ADS MOSFET device to place. For more information on MOSFET Models, refer to “MOSFET Models” on page 8-54.

Oxxxxxxx

Lossy Transmission Lines: This device and model pair is translated as an ideal physical transmission line.

Example SPICE Command Line:

```
o1 1 0 3 0 Omodel
.model Omodel LTRA L=9.19e-9 C=3.65e-12 LEN=1 R=0 G=0
```

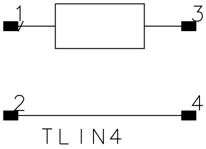
SPICE dialect and netlist syntax:

Spice3:	**o***id n1 n2 n3 n4 mname*
PSpice:	Does not exist
HSpice:	Does not exist

ADS Netlist Syntax:

TL:o*id* **Z=**sqrt(l/c) **L=***len* **V=**1/(c0*sqrt(l*c)) **R=***r* **G=***g* **F=**0

ADS Schematic Symbol:



Instance Parameters:

oid	=	TL element name
mname	=	Model name
n1, n2	=	Nodes at port 1
n3, n4	=	Nodes at port 2

Comments:

Parameters are read from the LTRA model statement, but no corresponding model is created in ADS. If no model is found, these defaults values are used:

Z=50Ohm L=1 V=1 R=0 G=0

For the schematic version, the TL component is not supported so the TLIN4 is used. TLIN4 does not support the R and G parameters so they are ignored. The following equations are used for TLIN4:

Z=sqrt(l/c) F=1 E=len/sqrt(l*c)*360

c0 = speed of light in free space. This is an ADS mathematical constant.

Qxxxxxxx

Semiconductor BJT Device: This device is translated as one of the following:

a nonlinear NPN or PNP BJT device
a nonlinear NPN or PNP BJT device with substrate
a NPN or PNP VBIC (Vertical Bipolar Inter-Company) device

For information on the BJT model, refer to "BJT_Model:Bipolar Transistor Model" on page 8-15. For information on the VBIC model, refer to "VBIC_Model:VBIC Model" on page 8-31.

Note For HSpice, if the referenced model has Level=4, the appropriate VBIC device is placed.

Example SPICE Command Line:

```
q1 1 2 3 Qmodel 2 OFF IC=0.6, 5.0
q2 1 2 3 4 Qmodel2 1.0
```

SPICE dialect and netlist syntax:

Spice2/3: **q**_id nc nb ne_ [_ns_] _mname_ [_area_] [**OFF**] [**IC=**_vbe_**,** _vce_] [**TEMP=**_temp_]

PSpice: **q**_id nc nb ne_ [_ns_] _mname_ [_area_]

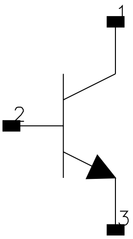
HSpice: **q**_id nc nb ne_ [_ns_] _mname_ [[**AREA=**]_area_] [**AREAB=**_area_]
+ [**AREAC=**_area_] [**OFF**] [[**IC=**_vbe_**,** _vce_] | [**VBC=**_vbc_ [**VCE=**_vce_]]]
+ [**M=**_mult_] [**DTEMP=**_temp_]

ADS Netlist Syntax:

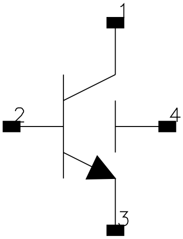
BJT[4] _mname_**: q**_id nc nb ne_ [_ns_] [**Area=**_area_] [**Temp=**_temp+dtemp_]
+ [**Region=**0] [**_M=**_mult_]

VBIC _mname_**: q**_id nc nb ne ns_ [**scale=**area]

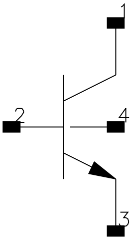
ADS Schematic Symbols:



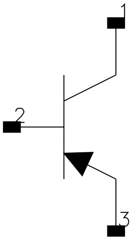
BJT_NPN



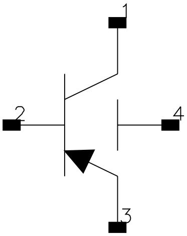
BJT4_NPN



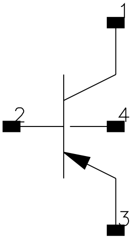
VBIC_NPN



BJT_PNP



BJT4_PNP



VBIC_PNP

Instance Parameters:

qid = BJT element name

mname = Model name

nc = Collector node

nb = Base node

ne = Emitter node

ns = Substrate node

area = Area factor (default=1)

off = Indicates an initial condition on the device for dc analysis. Translates to Region=0

mult = Multiplier used to simulate multiple parallel devices.

temp = The temperature at which this device is to operate.

Comments:

The referenced “BJT_Model:Bipolar Transistor Model” on page 8-15 or “VBIC_Model:VBIC Model” on page 8-31 model is read to determine which device to use:

NPN keyword indicates BJT_NPN, BJT4_NPN or VBIC_NPN
PNP keyword indicates BJT_PNP, BJT4_PNP or VBIC_PNP

If SCALE is specified, Value=Value*scale

If dtemp is provided, Temp=temp+dtemp

Rxxxxxxx

Resistor: This device is translated as a resistor device. For information on the resistor model, refer to “R_Model:Resistor Model” on page 8-4.

Example SPICE Command Line:

```
r1 1 2 1k
```

SPICE dialect and netlist syntax:

Spice2/3: **r***id n1 n2* [*value*] [*mname*] [**l=***l*] [**w=***w*] [**temp=***temp*]

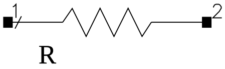
PSpice: **r***id n1 n2* [*mname*] *value* [**tc=***tc1* [,*tc2*]]

HSpice: **r***id n1 n2* [*mname*] [**r=**]*value* [[**tc1=**]*tc1* [[**tc2=**]*tc2*]]
+ [**scale=***s*] [**m=***mult*] [**ac=***ac*] [**dtemp=***dtemp*] [**l=***l*] [**w=***w*] [**c=***c*]

ADS Netlist Syntax:

R:r*id n1 n2* **R=***value* [**TC1=***tc1*] [**TC2=***tc2*] [**_M=***mult*]

ADS Schematic Symbol:



Instance Parameters:

rid = Resistor element name

mname = Model name. Use this name in elements to reference the model.

n1, n2 = Element nodes

value = Resistance value (in ohms). Reff=R*SCALE/M.

tc1 = First order temperature coefficient for resistor.

tc2 = Second order temperature coefficient for resistor.

scale = Element scale factor for resistance and capacitance. Default=1.0.

mult = Multiplier that simulates parallel resistors. For example, to represent two parallel instances of a resistor, set M=2 to multiply the number of resistors by 2. Default=1.0.

dtemp = Temperature difference between the element and the circuit. Default=0.0

l = Resistor length. Default=0.0, if L is not specified in the model.

w = Resistor width. Default=0.0, if W is not specified in the model.
SHRINK is a model parameter.
Wscaled=W*SHRINK*SCALE(option)

dw = Width narrowing due to etching in specified units.

dl = Length narrowing due to etching in specified units.

Comments:

If SCALE is specified, Value=Value*scale.

In the netlist, as in ADS, the value of R may be positive or negative but not zero.

HSpice: If the value of R is an expression which is a function of node voltages or independent variables, then the component is a dependent resistor. In ADS, the component is represented by a Symbolically Defined Device (SDD).

If one of the SDD parameters uses a variable containing node voltages, the translator needs to look up the node voltages on the resultant SDD to convert them to the correct ADS syntax and update the variable expression. For instance, the node voltage described as v(nodeX) in HSpice must be converted to _v1, if the first pair of nodes on the SDD is (node1, 0).

For translation purposes of this special case, the SDD and the variable used by it are expected to be defined in the same subcircuit, with the variable being used by only one SDD. If the variable were defined globally or used by different SDDs, the node voltage variables (_v1, etc.) might not match up correctly. For example, in the case described above one SDD might have (nodeX,0) as the first pair of nodes (_v1), while another SDD might have (nodeX,0) as the 2nd pair of nodes (_v2). The translator would not be able to replace v(nodeX) in the variable expression to define both SDDs correctly at the same time.

If this SDD is not found in the same subcircuit as the variable expression, a warning message will be written to the log file *spctoiff.log*. Also, the expression will not be converted automatically, and it will have to be fixed manually before simulation is attempted.

If dtemp is provided, Temp=temp+dtemp

The Netlist Translator fails to recognize the resistor model in the following syntax:

```
r1 1 2 rmodel
```

The translator assumes that it is a variable name for the resistor value. To correct your imported schematic, manually move the model name to the parameter Model. To correct the ADS Netlist, use the following netlist syntax:

```
rmodel:r1 1 2 [param=value]*
```

Txxxxxxx

Transmission Lines (Lossless): This device is translated as an ideal 4-terminal transmission line.

Example SPICE Command Line:

```
t1 1 0 2 0 z0=50 TD=10ns IC=1mv, 1ma, .2mv, .6ma
t2 1 0 2 0 z0=50 F=1.7GHz NL=0.3
t3 1 0 2 0 z0=50 F=1.2GHz
```

SPICE dialect and netlist syntax:

Spice2/3: **t***id n1 n2 n3 n4* **Z0=***zval* [[**TD=***tval*] | [**F=***fval* [**NL=***nval*]]]
+ **[IC=***v1***,** *i1***,** *v2***,** *i2*]

PSpice: **t***id n1 n2 n3 n4* [**model**] **Z0=***zval* [[**TD=***tval*] | [**F=***fval* [**NL=***nval*]]]
+ [**IC=***v1***,** *i1***,** *v2***,** *i2*]

HSpice: **t***id n1 n2 n3 n4* [**model**] **Z0=***zval* [[**TD=***tval* [**L=***l*]] |
+ [**F=***fval* [**NL=***nval*]]] [**IC=***v1***,** *i1***,** *v2***,** *i2*]

ADS Netlist Syntax:

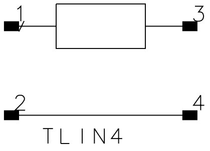
If TD is given:

TLIN4: t*id n1 n2 n3 n4* **Z=***zval* **F=**1 **E=***tval**360

If F is given:

TLIN4: t*id n1 n2 n3 n4* **Z=***zval* **F=***fval* **E=***nval**360

ADS Schematic Symbol:



Instance Parameters:

tid	=	TL element name
n1, n2	=	Nodes at port 1
n3, n4	=	Nodes at port 2

zval = Characteristic impedance

tval, fval = Length of line may be expressed in either of two forms. The transmission delay (TD) may be specified directly or frequency (F) may be given together with NL.

nval = Normalized electrical length of the transmission line with respect to the wavelength in the line frequency. If F is specified but NL is omitted, 0.25 is assumed.

v1, i1, v2, i2 = The voltage and current at each of the transmission line ports. Ignored translation.

Comments:

PSpice:

optional lossy format: n1 n2 n3 n4 LEN=len R=r L=l G=g C=c

R and G are not supported for the schematic version. For the netlist version, the alternate component TL will be used so these values are handled properly.

ADS Netlist Syntax:

TL:tid n1 n2 n3 n4 Z=sqrt(l/c) L=len V=1/(c0*sqrt(l*c))
R=r G=g F=0

If a model reference is used, the model is read and parameters are used on the device as needed. No model is placed and unused parameters are discarded.

HSpice:

optional syntax: tid n1 n2 n3 n4 Z=zval l=len

translates as: TLIN4:tid n1 n2 n3 n4 z=zval F=1 E=(len/c0)*360

c0 = speed of light in free space. This is an ADS mathematical constant.

Uxxxxxxx

Uniform Distributed RC Lines (Lossy): This device is translated as two ideal transformers and one ideal distributed RC line.

Example SPICE Command Line:

```
u1 1 2 3 URCmodel L=50u N=6
.model URCmodel URC RPERL=rperl CPERL=cperl
```

SPICE dialect and netlist syntax:

Spice2/3: **u***id nin nout nref mname* **L=***len* **N=***lumps*

PSpice: Does not exist

HSpice: Not supported

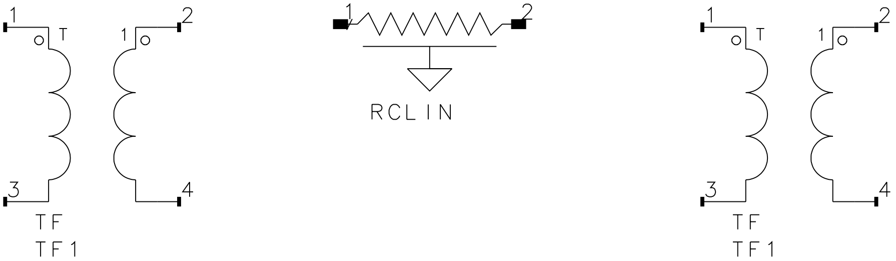
ADS Netlist Syntax:

RCLIN:u*id nin nout* **L=***len* **R=***rperl* **C=***cperl*

OR If node 3 is not ground:

tf:tf1 *nin nref* net1 0
RCLIN:u*id* net1 net2 **L=***len* **R=***rperl* **C=***cperl*
tf:tf2 net2 0 *nout nref*

ADS Schematic Symbol:



Instance Parameters:

uid = Element name

nin, nout = The two element nodes the RC line connects

nref = Node to which the capacitance is connected.

net1, net2 = Nodes inserted to connect RCLIN to transformers.

L = Length of the RC line in meters.

lumps = The number of lumped segments to use in modeling the RC line.

rperl = Resistance per unit length in Ohm/m (default=1000).

cperl = Capacitance per unit length in F/m (default=1.0e-15.

Comments:

N is ignored in translation.

Model parameters are read off the URC model, but no model is placed. Unused parameters are discarded. If the referenced model is not found, the default values specified above are used.

If node 3 is not ground, the transformers are added to float the RCLIN.

Vxxxxxxx

Independent Voltage Source: This device is translated as either a V_Source (ADS Netlist) or a VtUserDef (ADS Schematic).

Example SPICE Command Line:

```
v1 1 2 DC 6
v2 2 3 6 AC 1 90
v3 3 0 PULSE(-1 1 2NS 2NS 2NS 50NS 100NS)
v4 3 0 SIN(0 1 100MEG 1NS 1E10)
v5 3 0 EXP(-4 -1 2NS 30NS 60NS 40NS)
v6 7 5 PWL(0 -7 10NS -7 11NS -3 17NS -3 18NS -7 50NS -7)
v7 12 0 SFFM(0 1M 20K 5 1K)
```

SPICE dialect and netlist syntax:

Spice2/3: **v***id n+ n-* [[**dc**] *dcval*] [**ac** [*mag*[*phase*]]] [*tranfunc*]

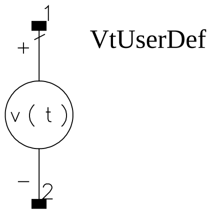
PSpice: **v***id n+ n-* [[**dc**] *dcval*] [**ac** [*mag*[*phase*]]] [*tranfunc*]

HSpice: **v***id n+ n-* [[**dc**] *dcval*] [**ac** [*mag*[*phase*]]] [*tranfunc*]

ADS Netlist Syntax:

V_Source:v*id n+ n-* **Vdc=***dcval* **Vac=polar(***mag,phase***)** **V_Tran=***tranfunc*

ADS Schematic Symbol:



Instance Parameters:

vid = Element name. Note that leading bold letter is included in the device name.

n+ = Positive node

n- = Negative node

dcval = The dc voltage

mag, phase = The magnitude and phase of the ac voltage.

tranfunc = Transient Source Functions. There are several built in ADS functions that mimic the SPICE transient source functions listed below.

SPICE	**Advanced Design System**
EXP()	exp_pulse(time, low, high, tdelay1, tau1, tdelay2, tau2)
PULSE()	pulse(time, low, high, delay, rise, fall, width, period)
PWL()	pwl(time, t1, x1,..., tn, xn)
SFFM()	sffm(time, offset, amplitude, carrier_freq, mod_index, signal_freq)
SIN()	damped_sin(time, offset, amplitude, freq, delay, damping)
HSpice	
PU()	pulse(time, low, high, delay, rise, fall, width, period)
AM()	Not translated.
PL()	Not translated.

Comments:

HSpice:

HSpice multiplier on Voltage sources - This is implemented by making an expression out of the given voltage value. Example:

HSpice: v1 n1 n2 500uv m=10
ADS: eqn0=(500uv)*(10)
V_Source: v1 n1 n2 Vdc=500uv*10

Unsupported HSpice transient functions - AM, PL

Unsupported HSpice syntax extensions -

Data driven PWL source
Repeat and Time Delay specifications in PWL transient function (R=repeat, TD=delay)

PSpice:

Unsupported PSpice syntax extensions

STIMULUS
corner points, TIME_SCALE_FACTOR, VALUE_SCALE_FACTOR

Xxxxxxxx

Subcircuit Reference: This device is translated as an instance of a subnetwork. For information on the parametric subnetwork command, refer to “.SUBCKT” on page 10-18.

Example SPICE Command Line:

```
x1 2 4 17 31 opamp
```

SPICE dialect and netlist syntax:

Spice2/3: **x***id* [*n1**] *subname*

PSpice: **x***id* [*n1**] *subname* [**OPTIONAL:** *interface node*=*default value*]
+ [**PARAMS:** *name*=*value**] [**TEXT:** *name*=*textvalue*]

HSpice: **x***id* [*n1**] *subname* **[***name*=*value**] [**m=***mult*]

ADS Netlist Syntax:

SUBNAME:x*id* [*n1**] [*name*=*value**] [**_M=***mult*]

Instance Parameters:

xid	=	Subnetwork element name
n1*	=	Node names
subname	=	Specifies the name of the subnetwork.
name	=	Parameter name or subcircuit definition.
value	=	A parameter value passed to the subnetwork.
mult	=	Multiplier used to simulate multiple parallel devices.

Comments:

PSpice:

OPTIONAL and TEXT parameters are ignored by the translator.

Zxxxxxxx

Semiconductor MESFET Device: This device is translated as a nonlinear N-type GaAsFET device. For information on the GaAsFET model, refer to the "Statz_Model:Statz (Raytheon) GaAsFET Model" on page 8-99.

Example SPICE Command Line:

```
z1 1 2 3 Zmodel 2 OFF IC=0.6, 5.0
```

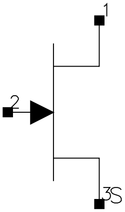
SPICE dialect and netlist syntax:

Spice2/3:	**z***id nd ng ns mname* [*area*] [**OFF**] [**IC=***vds*, *vgs*]
PSpice:	Does not exist
HSpice:	Does not exist

ADS Netlist Syntax:

mname: **z***id nd ng nd* **Area=***area* **Region=**0

ADS Schematic Symbol:



GaAsFET

Instance Parameters:

zid	=	GaAsFET element name
mname	=	Model name
nd	=	Drain node
ng	=	Gate node
ns	=	Source node
area	=	Area factor

OFF = Indicates an initial condition on the device for dc analysis. Translates to Region=0.

ic=vds, vgs = The initial condition specification intended for use with the UIC option on the .TRAN control line. Not translated.

Unsupported Devices

The following devices are not supported:

Axxxxxxx

Nxxxxxxx

Pxxxxxxx

Sxxxxxxx

Wxxxxxxx

Yxxxxxxx

The following additional information provides a suggested action for your translation, each of which requires using a Symbolically Defined Device (SDD). For information on SDD’s, refer to “Using a Symbolically Defined Device” on page 7-76.

Non-Linear Dependent Source (SPICE 3 only):

```
Bxxxxxxx N+ N- I=expression V=expression
```

Suggested Action

Insert an SDD to replace the unmapped element. Assign the appropriate wire labels to the SDD so you can maintain connectivity of the circuit. Insert the equation I=expression or V=expression into the correct current or voltage port definition.

Voltage Controlled Switch (SPICE 3 only):

```
Sxxxxxxx N+ N- NC+ NC- mname [ON] [OFF]
```

Suggested Action

Insert an SDD to replace the unmapped element. Assign the appropriate wire labels to the SDD so you can maintain connectivity of the circuit.

Current Controlled Switch (SPICE 3 only):

```
Wxxxxxxx N+ N- VNAM mname [ON] [OFF]
```

Suggested Action

Insert an SDD to replace the unmapped element. Assign the appropriate wire labels to the SDD so you can maintain connectivity of the circuit.

Using a Symbolically Defined Device

The symbolically-defined device (SDD) enables you to create equation based, user-defined, nonlinear components in ADS. The SDD is a multi-port device which is defined by specifying algebraic relationships that relate the port voltages, currents, and their derivatives, plus currents from certain other devices.

Controlled sources from SPICE are modeled in ADS using the symbolically-defined device. The SDD allows a voltage or current source to be constructed whose output can be described as an arbitrary combination of a number of controlling voltages or currents.

```
; e voltage controlled voltage source
SDD:name  nc1p nc1n  nc2p nc2n  outp outn \
I[1,0]=0 I[2,0]=0 F[3,0]=_v3-(output_voltage(_v1, _v2))

; g voltage controlled current source
SDD:name nc1p nc1n  nc2p nc2n  outp outn \
I[1,0]=0 I[2,0]=0 I[3,0]=(output_current(_v1, _v2))

; h current controlled voltage source
SDD:name  outp outn  C[1]=current1 C[2]=current2 \
F[1,0]=_v1-(output_voltage(_c1, _c2))

; f current controlled current source
SDD:name  outp outn  C[1]=current1 C[2]=current2 \
I[1,0]=(output_current(_c1, _c2))
```

The voltage controlled sources have 2(n+1) nodes, where n is the number of controls. The controlling voltage nodes are listed first in pairs (nc1p nc1n) at ports 1 through n, then the output node pair (outp outn) at port n+1. The controlling voltages have symbolic names like _v1. Each controlling voltage must have a current equation written for it (I[1,0]=0) so that no current flows between the controlling nodes through the SDD.

The current controlled sources have only 2 output nodes at port 1, while the name of the sources used to measure the controlling currents are specified using C[1]=”ivs1”. The controlling currents have symbolic names like _c1.

Current sources are written as a simple expression for the current at the output port. This expression is in SI units and is written in terms of the controls:

```
I[3,0]=_v1*1e-3+_v2*2e-3
```

A voltage source must have an implicit expression for the voltage at the output port such that F[3,0]=_v3-vout, where vout is the desired output voltage:

```
F[3,0]=_v3-(_v1*2+_v1*v2)
```

Controlled sources in SPICE may be linear or nonlinear and there may be one or more controlling voltages or currents. The general form of a controlled source is:

```
?id node1 node2 [poly(dim)] [sourcename | nodecp nodecn]*
+  value [value]*
```

The source has two nodes with the same polarity convention as the dependent sources. The method of specifying controls depends on whether the source is voltage controlled or current controlled. Current-controlled sources give the name of a source through which the current is measured; voltage-controlled sources provide two nodes with which to measure a differential voltage. The simple linear form for all four types:

```
eid n+ n- nc+ nc- value    ; vout = value * vcon
fid n+ n- vname   value    ; iout = value * icon
gid n+ n- nc+ nc- value    ; iout = value * vcon
hid n+ n- vname   value    ; vout = value * icon
```

A nonlinear source may be created whose output is a polynomial function of its control:

```
eid n+ n- nc+ nc- v0 v1 v2 v3 ... [ic=]
```

The polynomial is interpreted as:

v_{out}=v0+v1*v_{con}+v2*v^2_{con}+…

unless only a single value is given, in which case it reverts to a linear behavior:

v_{out}=v0*v_{con}

SPICE allows controlled sources with more than one control. When there is more than one control. the poly keyword must be used to identify the number of controls. The dimension refers to the number of controlling sources, not the order of the polynomial.

```
eid n+ n- poly(dim) [ nc+ nc- ]* values ... [ic=]
```

Coefficients for a polynomial of dimension two are interpreted as:

$v_{\mathrm{out}}=\mathrm{v0}$

$+\mathrm{v1}*v_{\mathrm{con1}}+\mathrm{v2}*v_{\mathrm{con2}}$
$+\mathrm{v3}*v^2_{\mathrm{con1}}+\mathrm{v4}*v_{\mathrm{con1}*}v_{\mathrm{con2}}+\mathrm{v5}*v^2_{\mathrm{con2}}$
$+\mathrm{v6}*v^3_{\mathrm{con1}}+\mathrm{v7}*v^2_{\mathrm{con1}}*v_{\mathrm{con2}}+\mathrm{v8}*v_{\mathrm{con1}*}v^2_{\mathrm{con2}}+\mathrm{v9}*v^3_{\mathrm{con2}}$
$+\ldots$

Coefficients for a polynomial of dimension three are interpreted as:

$v_{\mathrm{out}}=\mathrm{v0}$

$+\mathrm{v1}*v_{\mathrm{con1}}+\mathrm{v2}*v_{\mathrm{con2}}+\mathrm{v3}*v_{\mathrm{con3}}$
$+\mathrm{v4}*v^2_{\mathrm{con1}}+\mathrm{v5}*v_{\mathrm{con1}*}v_{\mathrm{con2}}+\mathrm{v6}*v_{\mathrm{con1}}*v_{\mathrm{con3}}+\mathrm{v7}*v^2_{\mathrm{con2}}+\mathrm{v8}*v_{\mathrm{con2}*}v_{\mathrm{con3}}+\mathrm{v9}*v^2_{\mathrm{con3}}$
$+\mathrm{v10}*v^3_{\mathrm{con1}}+\mathrm{v11}*v^2_{\mathrm{con1}*}v_{\mathrm{con2}}+\mathrm{v12}*v^2_{\mathrm{con1}}*v_{\mathrm{con3}}+\mathrm{v13}*v_{\mathrm{con1}}*v^2_{\mathrm{con2}}+\mathrm{v14}*v_{\mathrm{con1}*}v_{\mathrm{con2}}*$
v_{con3}
$+\mathrm{v15}*v_{\mathrm{con1}}*v^2_{\mathrm{con3}}+\mathrm{v16}*v^3_{\mathrm{con2}}+\mathrm{v17}*v^2_{\mathrm{con2}}*v_{\mathrm{con3}}+\mathrm{v18}*v_{\mathrm{con2}}*v^2_{\mathrm{con3}}+\mathrm{v19}*v^3_{\mathrm{con3}*}$
$+\mathrm{v20}*v^4_{\mathrm{con1}}+\ldots$

The SPICE syntax provides no simple way to specify just a couple of the individual terms. If only the `v14` term is needed, `v0` through `v13` must still be specified with zeros.

Initial conditions here are used to provide a guess at what the controlling values will be so the output of the dependent source can be evaluated; the initial values do not actually influence the controlling values. Enough initial conditions must be provided to match the dimensionality of the source.

Several translation examples for SPICE are presented below.

Example1

SPICE Netlist:

```
* linear vcvs
e01 vout 0 vc1p vc1n 0.05
```

ADS Netlist:

```
SDD:e01 vc1p vc1n vout 0 I[1,0]=0 F[2,0]=_v2-(0.05*_v1)
```

Example 2

SPICE Netlist:

```
* linear cccs
f01 vout 0 vname 0.05
```

ADS Netlist:

```
SDD:f01 vout 0 C[1]=vname I[1,0]=0.05*_c1
```

Example 3

SPICE Netlist:

```
* linear vccs
g01 vout 0 vc1p vc1n 0.05
```

ADS Netlist:

```
SDD:g01 vc1p vc1n vout 0 I[1,0]=0 I[2,0]=0.05*_v1
```

Example 4

SPICE Netlist:

```
* linear ccvs
h01 vout 0 vname 0.05
```

ADS Netlist:

```
SDD:h01 vout 0 C[1]=vname I[1,0]=0 F[1,0]=_v1-(0.05*_c1)
```

Example 5

SPICE Netlist:

```
* one input polynomial vcvs
* vout=0.5+0.1*vc1+0.01*vc1^2+0.001*vc1^3
e02 vout 0 nc1p nc1n 0.5 0.1 0.01 0.001
```

ADS Netlist:

```
SDD:e02 nc1p nc1n vout 0 I[1,0]=0 \
F[2,0]=_v2-(0.5+0.1*_v1+0.01*_v1^2+0.001*_v1^3)
```

Example 6

SPICE Netlist:

```
* two input polynomial cccs
* iout=0.0+0.2*i1+0.3*i2+0.01*i1^2+0.02*i1*i2+0.03*i2^2
f02 vout 0 poly(2) vname1 vname2 0.0 0.2 0.3 0.01 0.02 0.03
```

ADS Netlist:

```
SDD:f02 vout 0 C[1]=vname1 C[2]=vname2 \
I[1,0]=0.0+0.2*_c1+0.3*_c2+0.01*_c1^2+0.02*_c1*_c2+0.03*_c2^2
```

Example 7

SPICE Netlist:

```
* three input polynomial vccs
* iout=0.1*vc1+0.2*vc2+0.3*vc3
g02 vout 0 poly(3) vc1p vc1n vc2p vc2n vc3p vc3n
+ 0 0.1 0.2 0.3
```

ADS Netlist:

```
SDD:g02 vc1p vc1n vc2p vc2n vc3p vc3n vout 0 \
I[1,0]=0 I[2,0]=0 I[3,0]=0 \
F[4,0]=_v4-(0+0.1*_v1+0.2*_v2+0.3*_v3)
```

For more information, refer to the section on the "*Symbolically Defined Device*" in the ADS *Circuit Components* manual.

Chapter 8: Translating a Model

This chapter provides model translation information for each of the models supported by the Netlist Translator. The following are the supported ADS models:

Using Parameter Mapping Tables to Understand a Translation

Model parameter information is organized in tables with the ADS parameter, unit, and default value in the three left-hand columns of the table. The Spectre, Berkeley SPICE 2 & 3 (Spice2/3), PSpice and HSpice parameter translation information is described in the three right- hand columns. In some cases where ADS does not translate a particular SPICE dialect, the column for that SPICE dialect has been removed.

The parameters that are not translated are included in the tables. The parameters under *Parameters not in Spectre Model* are parameters that are supported in ADS and are not supported in Spectre. The parameters under *Parameters not in ADS Model* are parameters that are supported in Spectre and are not translated to ADS.

For each SPICE parameter in the Model Parameter table, the following conventions apply:

- An “x” indicates that the SPICE default value and units are the same as ADS. Any deviations are listed explicitly.
- A “-” indicates the parameter doesn’t exist in that particular SPICE dialect.
- Extra parameter names listed are aliases and are converted to the proper name in ADS by the translator.

ADS Model Parameters that are not listed are not supported by any SPICE dialects and will use standard ADS defaults.

Refer to the Advanced Design System *Circuit Components* manual for an alphabetical listing of ADS supported models and a detailed description of their parameters and default values.

R, C, L and Diode Models for Spectre

This section contains detailed parameter information for the Resistor (R), Capacitor (C), Inductor (I), and Diode models:

- “R_Model:Resistor Model” on page 8-4
- “C_Model:Capacitor Model” on page 8-6
- “L_Model:Inductor Model” on page 8-8

- “Diode_Model:PN-Junction Diode Model” on page 8-10

R_Model:Resistor Model

The Spectre resistor is translated to the ADS R_Model. For translation information on the resistor device, refer to "Resistor Device" on page 7-5.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model resmod resistor rsh=150 l=2u w=2u etch=0.05u tc1=0.1 tnom=27 kf=1
```

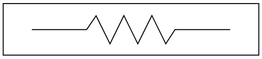
Spectre Netlist Syntax:

model *mname* **resistor** [param=*value*]*

ADS Netlist Syntax:

model *mname* **R_Model** [param=*value*]*

ADS Schematic Symbol:



R_Model

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-1. R_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
R	ohms	50	r	ohms	infinity
Rsh	ohms/sq	0.0	rsh	ohms/sq	infinity
Length	m	0.0	l	m	infinity
Width	m	0.0	w	m	1e-6
Dw	m	0.0	etch	m	0.0
Dl	m	0.0	etchl	m	0.0
Tnom	°C	25	tnom	°C	set by options

Table 8-1. R_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
TC1	1/°C	0.0	tc1	1/°C	0.0
TC2	1/°C^2	0.0	tc2	1/°C^2	0.0
Scale		1.0	scaler		1.0
Parameters not in Spectre Model					
Narrow	m	0.0			
wPmax	W	infinity			
wImax	A	infinity			
AllParams					
			Parameters not in ADS Model		
			trise	°C	0.0
			thres	ohms	1.0e-3
			coeffs=[...]		
			kf		0.0
			af		2.0

C_Model:Capacitor Model

The Spectre capacitor model is translated to the ADS C_Model. For translation information on the capacitor device, refer to "Capacitor Device" on page 7-7.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model proc_cap capacitor c=2u tc1=1.2e-8 tnom=25 w=4u l=4u cjsw=2.4e-10
```

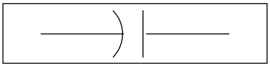
Spectre Netlist Syntax:

model *mname* **capacitor** [param=*value*]*

ADS Netlist Syntax:

model *mname* **C_Model** [param=*value*]*

ADS Schematic Symbol:



C_Model

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-2. C_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
C	F	1e-12 (1 pF)	c	F	0.0
Cj	F/m^2	0.0	cj	F/m^2	0.0
Cjsw	F/m	0.0	cjsw	F/m	0.0
Length	m		l	m	0.0
Width	m		w	m	0.0
Narrow	m	0.0	etch	m	0.0

Table 8-2. C_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Tnom	°C	25	tnom	°C	set by options
TC1	1/°C	0.0	tc1	1/°C	0.0
TC2	1/°C^2	0.0	tc2	1/°C^2	0.0
Scale		1	scalec		1
Parameters not in Spectre Model					
wBv	W	infinity			
AllParams					
			Parameters not in ADS Model		
			trise	°C	
			rforce	ohms	1.0
			coeffs=[...]		

L_Model:Inductor Model

The Spectre inductor is translated to the ADS L_Model. For translation information on the inductor device, refer to "Inductor Device" on page 7-9.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model ind inductor l=6e-9 r=1 tc1=1e-12 tc2=1e-12 tnom=25
```

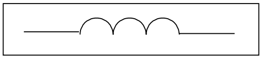
Spectre Netlist Syntax:

model *mname* **inductor** [param=*value*]*

ADS Netlist Syntax:

model *mname* **L_Model** [param=*value*]*

ADS Schematic Symbol:



L_Model

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-3. L_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
L	H	0.0	l	H	0.0
R	ohms	0.0	r	ohms	0.0
Tnom	°C		tnom	°C	set by options
TC1	1/°C		tc1	1/°C	0.0
TC2	1/°C^2		tc2	1/°C^2	0.0
Scale			scalei		1
Parameters not in Spectre Model					

Table 8-3. L_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
AllParams					
			Parameters not in ADS Model		
			trise	°C	
			rforce	ohms^2	1.0
			coeffs=[...]		
			kf		0.0
			af		2.0

Diode_Model:PN-Junction Diode Model

The Spectre diode model is translated to the ADS Diode_Model. For translation information on the diode device, refer to "Diode Device" on page 7-13.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model pdiode diode is=1.8e-8 rs=1.43 n=1.22 nz=2.31 gleak=6.2e-5 rsw=10
isw=6.1e-10 ibv=0.95e-3 tgs=2 ik=1.2e7 fc=0.5 cj=1.43e-3 pb=0.967 mj=0.337
cjsw=2.76e-9 vjsw=0.94 imax=1e20
```

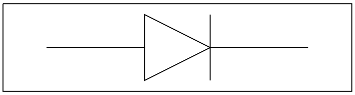
Spectre Netlist Syntax:

model *mname* **diode** [param=*value*]*

ADS Netlist Syntax:

model *mname* **Diode** [param=*value*]*

ADS Schematic Symbol:



Diode_Model

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-4. Diode_Model Parameters Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Level		1	level		1
Is	A	10e-14	is	A	10e-14
Rs	ohms	0.0	rs	ohms	0.0
N		1.0	n		1.0
Tt	sec	0.0	tt	sec	0.0
Cjo	F	0.0	cjo	F	0.0

Table 8-4. Diode_Model Parameters Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Vj	V	1.0	vj pb	V	1.0
M		0.5	m		0.5
Eg	eV	1.11	eg	eV	1.11
Imax	A	1.0	imax	A	1.0
Xti		3.0	xti		3.0
Kf		0.0	kf		0.0
Af		1.0	af		1.0
Fc		0.5	fc		0.5
Bv	V	infinity (0)	bv	V	infinity
Ibv	A	0.001	ibv	A	0.001
Ikf	A	infinity (0)	ik	A	infinity
Nbv		1.0	nz		1.0
Tnom	°C	25	tnom	°C	set by options
Jsw	A	0.0	isw	A	0.0
Cjsw	F	0.0	cjsw	F	0.0
Msw		0.33	mjsw		0.33
Vjsw	V	1.0	vjsw	V	1.0
Fcsw		0.0	fc		0.5
Area		0	area		1
Periph		0	perim		0
Tlev		0	tlev		0
Tlevc		00	tlevc		0
EgAlpha	V/°C	7.02e-4	gap1	V/°C	7.02e-4
EgBeta	K	1108	gap2	°C	1108
Tcjo	1/°C	0.0	cta	1/°C	0.0
Tcjsw	1/°C	0.0	ctp	1/°C	0.0
Tvj	1/°C		pta	V/°C	0.0
Tvjsw	1/°C		ptp	V/°C	0.0

Table 8-4. Diode_Model Parameters Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Trs	1/°C		trs	1/°C	0.0
Tbv	1/°C	0.0	tbv1	1/°C	0.0
wBv	V	infinity	bvj	V	infinity
Parameters not in Spectre Model					
Pb	V				
Isr	A				
Nr					
Isr	A	0.0			
Nr		2.0			
Ikr	A	0.0			
IkModel					
Ibvl	A	0.0			
Nbvl		1.0			
Ibvl	A				
Nbvl					
Ffe		1.0			
wBv	V	0.0			
Pbsw					
Length	m				
Width	m				
Dwl					
Shrink					
Pt					
Ttt1	1/°C				
Ttt2	1/°C^2				
Tm1	1/°C	0.0			
Tm2	1/°C^2				
wPmax	W	infinity			
ExpModel					

Table 8-4. Diode_Model Parameters Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Secured					
			Parameters not in ADS Model		
			minr	ohms	0.0
			ns		1
			cd	F	0.0
			imelt	A	imax
			ikp	A	ik
			allow_scaling		no
			etch	m	0.0
			etchl	m	etch
			trise	°C	0.0
			trs2	1/°C^2	0.0
			tbv2	1/°C^2	0.0
			dskip		yes
			gleak	Siemens	0.0
			gleaksw	Siemens	0.0
			tgs	1/°C^2	0.0
			tgs2	1/°C^2	0.0
			if	A/V^{nf}	1e-10
			ir	A/V^{nr}	if
			ecrf	V/m	2.55e10
			ecrr	V/m	ecrf
			nf		2
			nr		nf
			tox	m	1e-8

NPN, PNP, and VBIC BJT Models for Spectre

This section contains detailed parameter information for the BJT models:

- "BJT_Model:Bipolar Transistor Model" on page 8-15
- "VBIC_Model:VBIC Model" on page 8-31
- "LEVEL1_Model:LEVEL 1 MOSFET Model" on page 8-38

BJT_Model:Bipolar Transistor Model

The Spectre bjt model is translated to the ADS NPN or PNP BJT_Model. For translation information on the BJT device, refer to "BJT Device" on page 7-15.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model npn_mod bjt type=npn is=10e-13 bf=200 va=58.8 ikf=5.63e-3 rb=700
rbm=86 re=3.2 cje=0.352e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12
pc=0.55
```

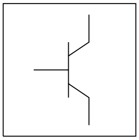
Spectre Netlist Syntax:

model *mname* **bjt type=**[**npn**|**pnp**] [param=*value*]*

ADS Netlist Syntax:

model *mname* **BJT NPN=**[**0**|**1**] **PNP=**[**0**|**1**] [param=*value*]*

ADS Schematic Symbol:



BJT_Model
NPN=yes
PNP=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-5. BJT_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
NPN †		yes	type		npn
PNP		no	type		npn
Is	A	1.0E-16	is	A	1.0E-16

Table 8-5. BJT_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Bf	A/A	100	bf	A/A	100
Nf		1.0	nf		1.0
Vaf	V	infinity	vaf	V	infinity
Ikf	A	infinity	ikf	A	infinity
Ise†	A	0.0	ise	A	0.0
Ne		1.5	ne		1.5
Br	A/A	1.0	br	A/A	1.0
Nr		1.0	nr		1.0
Var	V	infinity	var	V	infinity
Ikr	A	infinity	ikr	A	infinity
Isc†	A	0.0	isc	A	0.0
Nc		2.0	nc		2.0
Rb	Ohms	0.0	rb	Ohms	0.0
Irb	A	infinity	irb	A	infinity
Rbm	Ohms	0.0	rbm	Ohms	rb
Re	Ohms	0.0	re	Ohms	0.0
Rc	Ohms	0.0	rc	Ohms	0.0
Imax	A		imax	A	1000
Cje	F	0.0	cje	F	0.0
Vje	V	0.75	vje	V	0.75
Mje		1/3	mje		1/3
Cjc	F	0.0	cjc	F	0.0
Vjc	V	0.75	vjc	V	0.75
Mjc		1/3	mjc		1/3
Xcjc		1.0	xcjc		1.0
Cjs	F	0.0	cjs	F	0.0
Vjs	V	0.75	vjs	V	0.75
Mjs		0.0	mjs		0.0
Fc		0.5	fc		0.5

Table 8-5. BJT_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Tf	sec	0.0	tf	sec	0.0
Xtf		0.0	xtf		0.0
Vtf	V	infinity	vtf	V	infinity
Itf	A	0.0	itf	A	0.0
Ptf	degrees	0.0	ptf	degrees	0.0
Tr	sec	0.0	tr	sec	0.0
Kf		0.0	kf		0.0
Af		1.0	af		1.0
Kb		0.0	kb		0.0
Fb	Hz	1.0	bnoisefc	Hz	1.0
Iss	A	0.0	iss	A	0.0
Ns		1.0	ns		1.0

Lateral		for struct = vertical, default = no
		for struct = lateral, default = yes
			struct		for type = npn, default = vertical
					for type = pnp, default = lateral

RbModel		MDS	rbmod		spice
Tnom	°C		tnom	°C	Set by options
Tlev		0	tlev		0
Tlevc		0	tlevc		0
Eg	eV	1.11	eg	eV	1.11
EgAlpha	V/°C	7.04E-4	gap1	V/°C	7.02E-4
EgBeta	K		gap2	°C	1108
Tbf1	1/°C	0.0	tbf1	1/°C	0.0
Tbf2	1/°C^2	0.0	tbf2	1/°C^2	0.0
Tbr1	1/°C	0.0	tbr1	1/°C	0.0
Tbr2	1/°C^2	0.0	tbr2	1/°C^2	0.0
Tcbc	1/°C	0.0	ctc	1/°C	0.0

Table 8-5. BJT_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Tcbe	1/°C	0.0	cte	1/°C	0.0
Tccs	1/°C	0.0	cts	1/°C	0.0
Tikf1	1/°C	0.0	tikf1	1/°C	0.0
Tikf2	1/°C^2	0.0	tikf2	1/°C^2	0.0
Tikr1	1/°C	0.0	tikr1	1/°C	0.0
Tikr2	1/°C^2	0.0	tikr2	1/°C^2	0.0
Tirb1	1/°C	0.0	tirb1	1/°C	0.0
Tirb2	1/°C^2	0.0	tirb2	1/°C^2	0.0
Tis1	1/°C	0.0	tis1	1/°C	0.0
Tis2	1/°C^2	0.0	tis2	1/°C^2	0.0
Tisc1	1/°C	0.0	tisc1	1/°C	0.0
Tisc2	1/°C^2	0.0	tisc2	1/°C^2	0.0
Tise1	1/°C	0.0	tise1	1/°C	0.0
Tise2	1/°C^2	0.0	tise2	1/°C^2	0.0
Tiss1	1/°C	0.0	tiss1	1/°C	0.0
Tiss2	1/°C^2	0.0	tiss2	1/°C^2	0.0
Titf1	1/°C	0.0	titf1	1/°C	0.0
Titf2	1/°C^2	0.0	titf2	1/°C^2	0.0
Tmjc1	1/°C	0.0	tmjc1	1/°C	0.0
Tmjc2	1/°C^2	0.0	tmjc2	1/°C^2	0.0
Tmje1	1/°C	0.0	tmje1	1/°C	0.0
Tmje2	1/°C^2	0.0	tmje2	1/°C^2	0.0
Tmjs1	1/°C	0.0	tmjs1	1/°C	0.0
Tmjs2	1/°C^2	0.0	tmjs2	1/°C^2	0.0
Tnc1	1/°C	0.0	tnc1	1/°C	0.0
Tnc2	1/°C^2	0.0	tnc2	1/°C^2	0.0

Table 8-5. BJT_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Tne1	1/°C	0.0	tne1	1/°C	0.0
Tne2	1/°C^2	0.0	tne2	1/°C^2	0.0
Tnf1	1/°C	0.0	tnf1	1/°C	0.0
Tnf2	1/°C^2	0.0	tnf2	1/°C^2	0.0
Tnr1	1/°C	0.0	tnr1	1/°C	0.0
Tnr2	1/°C^2	0.0	tnr2	1/°C^2	0.0
Tns1	1/°C	0.0	tns1	1/°C	0.0
Tns2	1/°C^2	0.0	tns2	1/°C^2	0.0
Trb1	1/°C	0.0	trb1	1/°C	0.0
Trb2	1/°C^2	0.0	trb2	1/°C^2	0.0
Trc1	1/°C	0.0	trc1	1/°C	0.0
Trc2	1/°C^2	0.0	trc2	1/°C^2	0.0
Tre1	1/°C	0.0	tre1	1/°C	0.0
Tre2	1/°C^2	0.0	tre2	1/°C^2	0.0
Trm1	1/°C	0.0	trm1	1/°C	0.0
Trm2	1/°C^2	0.0	trm2	1/°C^2	0.0
Ttf1	1/°C	0.0	ttf1	1/°C	0.0
Ttf2	1/°C^2	0.0	ttf2	1/°C^2	0.0
Ttr1	1/°C	0.0	ttr1	1/°C	0.0
Ttr2	1/°C^2	0.0	ttr2	1/°C^2	0.0
Tvaf1	1/°C	0.0	tvaf1	1/°C	0.0
Tvaf2	1/°C^2	0.0	tvaf2	1/°C^2	0.0
Tvar1	1/°C	0.0	tvar1	1/°C	0.0
Tvar2	1/°C^2	0.0	tvar2	1/°C^2	0.0
Tvjc	1/°C	0.0	tvjc	1/°C	0.0
Tvje	1/°C	0.0	tvje	1/°C	0.0

Table 8-5. BJT_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Tvjs	1/°C	0.0	tvjs	1/°C	0.0
Xtb		0.0	xtb		0.0
Xti		3.0	xti		3.0
wVsubfwd	V	infinity	vsubfwd	V	0.2
wBvsub	V	infinity	bvsub	V	infinity
wBvbe	V	infinity	bvbe	V	infinity
wBvbc	V	infinity	bvbc	V	infinity
wVbcfwd	V	infinity	vbcfwd	V	0.2
wIbmax	A	infinity	imax	A	1E3
wIcmax	A	infinity	imax1	A	imax
Parameters not in Spectre Model					
Ab		1.0			
Secured					
Nk		0.5			
Ffe		1.0			
Approxqb		yes			
wPmax	W	infinity			
Null					
Temp†	°C	25			
			Parameters not in ADS Model		
			c2		0.0
			c4		0.0
			cb0	A	0.0
			gb0	Siemens	0.0
			vb0	V	0.0
			ke	1/V	0.0
			kc	1/V	0.0
			dope	cm^3	1E15
			cex		1.0

Table 8-5. BJT_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
			cco	A	1.0
			rvc	Ohm	0.0
			rcm	Ohm	0.0
			minr	Ohm	0.1
			rbnoi	Ohm	rb
			xcjc2		1.0
			td	sec	0.0
			tvtf1	1/°C	0.0
			tvtf2	1/°C^2	0.0
			txtf1	1/°C	0.0
			txtf2	1/°C^2	0.0
			trise	°C	0.0
			dskip		yes
			alarm		none
			imelt	A	imax
			bvce	V	infinity
			vbefwd	V	0.2

HICUM_Model:HICUM Bipolar Transistor Model

The SPICE Q model is translated to the ADS NPN or PNP BJT_Model. For translation information on the BJT device, refer to "BJT Device" on page 7-15.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model hicum NPN level=8
```

SPICE Dialect and Netlist Syntax:

Spice2/3: Does not exist.

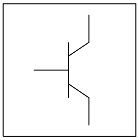
PSpice: Does not exist.

HSpice: **.model** *mname* **NPN | PNP LEVEL=8**[param=*value*]*

ADS Netlist Syntax:

model *mname* **HICUM NPN=**[**0**|**1**] **PNP=**[**0**|**1**] [param=*value*]*

ADS Schematic Symbol:



HICUM_model
HICUM_NPN
HICUM_PNP

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-6. HICUM_Model Parameters

ADS	Unit	Default	Spice2/3	PSpice	HSpice
C10		3.76e-32	x	x	3.76e-32
Qp0		2.78e-14	x	x	2.78e-14

Table 8-6. HICUM_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Ich	A	2.09e-02	x	x	2.09e-02
Hfe		1.0	x	x	1.0
Hfc		1.0	x	x	1.0
Hjei		0.0	x	x	0.0x
Hjci		1.0	x	x	1.0
Mcf		1.0	x	x	1.0x
Alit		0.45	x	x	0.45
Cjei0	F	8.11e-15	x	x	8.11e-15
Vdei	V	0.95	x	x	0.95
Zei		0.5	x	x	0.5
Aljei		1.8	x	x	1.8
Cjci0	F	1.16e-15	x	x	1.16e-15
Vdci	V	0.8	x	x	0.8
Zci		0.333	x	x	0.333
Vptci	V	416	x	x	416
T0	s	4.75e-12	x	x	4.75e-12
Dt0h	s	2.1e-12	x	x	2.1e-12
Tbvi	s	40e-12	x	x	40e-12
Tef0	s	1.8e-12	x	x	1.8e-12
Gtfe		1.4	x	x	1.4
Thcs	s	3.0e-11	x	x	3.0e-11
Alhc		0.75	x	x	0.75
Fthc		0.6	x	x	0.6
Alqf		0.225	x	x	0.225
Rci0	ohms	127.8	x	x	127.8
Vlim	V	0.7	x	x	0.7
Vpt	V	5.0	x	x	5.0
Vces	V	0.1	x	x	0.1
Tr	s	1.0e-9	x	x	1.0e-9
Ibeis	A	1.16e-20	x	x	1.16e-20

Table 8-6. HICUM_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Mbei		1.015	x	x	1.015
Ireis	A	1.16e-6	x	x	1.16e-6
Mrei		2.0	x	x	2.0
Ibcis	A	1.16e-20	x	x	1.16e-20
Mbci	V	1.015	x	PS	1.015
Favl	V^-1	1.186	x	x	1.186
Qavl		1.11e-14	x	x	1.11e-14
Rbi0	ohms	0	x	x	0
Fdqr0		0.0	x	x	0.0
Fgeo		0.73	x	x	0.73
Fqi		0.9055	x	x	0.9055
Fcri		0.0	x	x	0.0
Latb		3.765	x	x	3.765
Latl		0.342	x	x	0.342
Cjep0	F	2.07e-15	x	x	2.07e-15x
Vdep	V	1.05	x	x	1.05
Zep		0.4	x	x	0.4
Aljep		2.4	x	x	2.4
Ibeps	A	3.72e-21	x	x	3.72e-21
Mbep		1.015	x	x	1.105
Ireps	A	1e-30	x	x	1e-30
Mrep		2.0	x	x	2.0
Ibets	A	0	x	x	0
Abet		0.0	x	x	0.0x
Cjcx0	F	5.393e-15	x	x	5.393e-15
Vdcx	V	0.7	x	x	0.7
Zcx		0.333	x	x	0.333
Vptcx	V	100	x	x	100
Ccox	F	2.97e-15	x	x	2.97e-15
Fbc		0.1526	x	x	0.1526

Table 8-6. HICUM_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Ibcxs	A	4.39e-20	x	x	4.39e-20
Mbcx		1.03	x	x	1.03
Ceox	F	1.13e-15	x	x	1.13e-15
Rbx	ohms	0	x	x	0
Re	ohms	0	x	x	0
Rcx	ohms	0	x	x	0
Itss	A	0.0	x	x	0.0
Msf		0.0	x	x	0.0
Msr		1.0	x	x	1.0
Iscs	A	0.0	x	x	0.0
Msc		0.0	x	x	0.0
Tsf		0.0	x	x	0.0
Cjs0	F	3.64e-14	x	x	3.64e-14
Vds	V	0.6	x	x	0.6
Zs		0.447	x	x	0.447
Vpts	V	1000	x	x	1000
Rsu	ohms	0	x	x	x
Csu	F	0	x	x	0
Kf		1.43e-8	x	x	1.43e-8
Af		2.0	x	x	2.0
Krbi		1.17	x	x	1.17
Vgb	V	1.17	x	x	1.17
Alb	K^-1	6.3e-3	x	x	6.3e-3
Zetaci		1.6	x	x	1.6
Alvs	K^-1	1e-3	x	x	1e-3
Alt0	K^-1	0	x	x	0
Kt0	K^-1	0	x	x	0
Alces	K^-1	0.4e-3	x	x	0.4e-3
Zetarbi		0.588	x	x	0.588
Zetarbx		0.2060	x	x	0.2060

Table 8-6. HICUM_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Zetarcx		0.2330	x	x	0.2330
Zetare		0	x	x	0
Alfav	K^-1	8.25e-5	x	x	8.25e-5
Alqav	K^-1	1.96e-4	x	x	1.96e-4
Tnom	C	0	x	x	0
Rth	ohms	0	x	x	0
Cth	F	0	x	x	0

Mextram_Model:Mextram Bipolar Transistor Model

The SPICE Q model is translated to the ADS NPN or PNP BJT_Model. For translation information on the BJT device, refer to "BJT Device" on page 7-15.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model mex504 npn level=vers=504 tref=25.0 is=22.0e-18 ...
```

SPICE Dialect and Netlist Syntax:

Spice2/3: Does not exist.

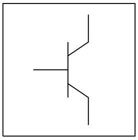
PSpice: Does not exist.

HSpice: **.model** *mname* **NPN | PNP LEVEL=6 VERS=504**[param=*value*]*

ADS Netlist Syntax:

model *mname* **MextramBJT NPN=**[**0**|**1**] **PNP=**[**0**|**1**] **Release=504** [param=*value*]*

ADS Schematic Symbol:



MEXTRAM_Model

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-7. Mextram_Model Parameters

ADS	Unit	Default	Spice2/3	PSpice	HSpice
LEVEL		6	x	x	6
MULT		1.0	x	x	

Table 8-7. Mextram_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Tref	C	25.0	x	x	25
DTA		0.0	x	x	
EXMOD		1	x	x	1
EXAVL		0	x	x	0
Is	A	22.0e-18	x	x	2.20E-17
Ik	A	0.1	x	x	0.1
Ver	V	2.5	x	x	2.5
Vef	V	44.0	x	x	44
Bf		215.0	x	x	215
Ibf	A	2.75e-15	x	x	2.70E-15
Mlf		2.0	x	x	2
Xibi		0.0	x	x	0
Bri		7.0	x	x	7.8
Ibr	A	1.0e-15	x	x	1.00E-15
Vlr	V	0.2	x	x	0.2
Xert		0.63	x	x	0.63
Wavl	gMeters	1.1e-6	x	x	1.00E-06
Vavl	V	3.0	x	x	3
Sfh		0.3	x	x	0.3
Re	ohms	5.0	x	x	5
Rbc	ohms	23	x	x	23
Rbv	ohms	18	x	x	18
Rcc	ohms	12	x	x	12
Rcv	ohms	150	x	x	150
SCRcv	ohms	1250	x	x	1250
Ihc	A	4e-3	x	x	4.00E-03
axi		0.3	x	x	0.3
Cje	farads	73e-15	x	x	73E-15
Vde	V	0.95	x	x	0.95
pe		0.4	x	x	0.4

Table 8-7. Mextram_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
XCje		0.4	x	x	0.4
Cbe0		0.0	x	x	0.0
Cjc	F	78e-15	x	x	7.80E-14
Vdc	V	0.68	x	x	0.68
pc		0.5	x	x	0.5
Xp		0.35	x	x	0.35
mc		0.5	x	x	10.5
XCjc		32e-3	x	x	32E-3
Cbc0		0	x	x	
mt		1	x	x	1
te	s	2e-12	x	x	2E-12
tb	s	4.2e-12	x	x	4.2E-12
tepi	s	41e-12	x	x	4.10E-11
tr	s	520e-12	x	x	5.20E12
dEg	Elec. V	0	x	x	0
Xrec	V	0	x	x	0
Aqb0		0.3	x	x	0.3
Ae		0	x	x	0
Ab		1	x	x	1
Aepi		2.5	x	x	2.5
Aex		0.62	x	x	0.62
Ac		2.0	x	x	2.0
dVgbf	V	0.05	x	x	5.00E-02
dVgbr	V	0.045	x	x	
Vgb	V	1.17	x	x	1.17
Vgj	V	1.15	x	x	1.15
dVgte	V	0.05	x	x	0.05
Af		2	x	x	2
Kf		2e-11	x	x	2.00E-11
Kfn		2e-11	x	x	2e-11

Table 8-7. Mextram_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Iss	A	4.8e-18	x	x	4.80E-17
Iks	A	2.5e-4	x	x	2.50E-04
Cjs	F	3.15e-13	x	x	3.15E-13
Vds	V	0.62	x	x	0.62
ps		0.34	x	x	0.34
Vgs	V	1.2	x	x	1.2
As		1.58	x	x	1.58
Rth		300	x	x	300
Cth		3e-9	x	x	300

Comments:

The Mextram 503 model is not currently supported in this translator.

VBIC_Model:VBIC Model

The Spectre vbic (Vertical Bipolar Inter-Company) model is translated to the ADS VBIC_Model. For translation information on the BJT device, refer to "BJT Device" on page 7-15.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model vbic_mod vbic type=npn
```

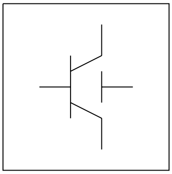
Spectre Netlist Syntax:

model *mname* **vbic type=**[**npn**|**pnp**] [param=*value*]*

ADS Netlist Syntax:

model *mname* **VBIC NPN=**[**0**|**1**] **PNP=**[**0**|**1**] [param=*value*]*

ADS Schematic Symbol:



VBIC_Model
NPN=yes
PNP=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-8. VBIC_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
NPN		yes	type		npn
PNP		no	type		npn
Tnom	°C	25	tnom	°C	set by options

Table 8-8. VBIC_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
(see *Temp* instance parameter)			trise†	°C	0.0
Rcx	ohms	0.0	rcx	ohms	0.0
Rci	ohms	0.0	rci	ohms	0.0
Vo	V	0.0	vo	V	0.0
Gamm		0.0	gamm	V	0.0
Hrcf		1.0	hrcf		1.0
Rbx	ohms	0.0	rbx	ohms	0.0
Rbi	ohms	0.0	rbi	ohms	0.0
Re	ohms	0.0	re	ohms	0.0
Rs	ohms	0.0	rs	ohms	0.0
Rbp	ohms	0.0	rbp	ohms	0.0
Is	A	10e-16	is	A	10e-16
Nf		1.0	nf		1.0
Nr		1.0	nr		1.0
Fc		0.9	fc		0.9
Cbeo	F	0.0	cbeo	F	0.0
Cje	F	0.0	cje	F	0.0
Pe		0.75	pe	V	0.75
Me		0.33	me		0.33
Aje		–0.5	aje		–0.5
Cbco	F	0.0	cbco	F	0.0
Cjc	F	0.0	cjc	F	0.0
Qco	C	0.0	qco	C	0.0
Cjep	F	0.0	cjep	F	0.0
Pc		0.75	pc		0.75
Mc		0.33	mc		0.33
Ajc		–0.5	ajc		–0.5
Cjcp	F	0.0	cjcp	F	0.0

Table 8-8. VBIC_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Ps		0.75	ps		0.75
Ms		0.33	ms		0.33
Ajs		–0.5	ajs		–0.5
Ibei	A	10e-18	ibei	A	10e-18
Wbe		1.0	wbe		1.0
Nei		1.0	nei		1.0
Iben	A	0.0	iben	A	0.0
Nen		2.0	nen		2.0
Ibci	A	10e-16	ibci	A	10e-16
Nci		1.0	nci		1.0
Ibcn	A	0.0	ibcn	A	0.0
Ncn		2.0	ncn		2.0
Isp	A	0.0	isp	A	0.0
Wsp		1.0	wsp		1.0
Nfp		1.0	nfp		1.0
Ibeip	A	0.0	ibeip	A	0.0
Ibenp	A	0.0	ibenp	A	0.0
Ibcip	A	0.0	ibcip	A	0.0
Ncip		1.0	ncip		1.0
Ibcnp	A	0.0	ibcnp	A	0.0
Avc1	1/V	0.0	avc1		0.0
Avc2	1/V	0.0	avc2		0.0
Ncnp		2.0	ncnp		2.0
Vef	V	0.0 (same as infinity)	vef	V	infinity
Ver	V	0.0 (same as infinity))	ver	V	infinity
Ikf	A	0.0 (same as infinity))	ikf	A	infinity

Table 8-8. VBIC_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Ikr	A	0.0 (same as infinity))	ikr	A	infinity
Ikp	A	0.0 (same as infinity))	ikp	A	infinity
Tf	sec	0.0	tf	sec	0.0
Qtf		0.0	qtf		0.0
Xtf		0.0	xtf		0.0
Vtf		0.0	vtf		0.0
Itf		0.0	itf		0.0
Tr	sec	0.0	tr	sec	0.0
Td	sec	0.0	td	sec	0.0
Kfn		0.0	kfn		0.0
Afn		1.0	afn		1.0
Bfn		1.0	bfn		1.0
Xre		0.0	xre		0.0
Xrb		0.0	xrb		0.0
Xrc		0.0	xrc		0.0
Xrs		0.0	xrs		0.0
Xvo		0.0	xvo		0.0
Ea	eV	1.12	ea	V	1.12
Eaie	eV	1.12	eaie	V	1.12
Eaic	eV	1.12	eaic	V	1.12
Eais	eV	1.12	eais	V	1.12
Eane	eV	1.12	eane	V	1.12
Eanc	eV	1.12	eanc	V	1.12
Eans	eV	1.12	eans	V	1.12
Xis		3.0	xis	V	3.0
Xii		3.0	xii	V	3.0
Xin		3.0	xin	V	3.0

Table 8-8. VBIC_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Tnf		0.0	tnf	V	0.0
Tavc		0.0	tavc	V	0.0
Rth	ohms	0.0	rth	ohms	0.0
Cth	F	0.0	cth	F	0.0
Imax	A	1.0	imax	A	1.0
wVsubfwd	V	infinity	vsubfwd	V	0.2
wBvsub	V	infinity	bvsub	V	infinity
wBvbe	V	infinity	bvbe	V	infinity
wBvbc	V	infinity	bvbc	V	infinity
wVbcfwd	V	infinity	vbcfwd	V	0.2
wIcmax	A	infinity	imax1	A	imax
Parameters not in Spectre Model					
wIbmax	A	infinity			
wPmax	W	infinity			
AllParams					
			Parameters not in ADS Model		
			imelt	A	10.0
			alarm		none
			selft		0
			dtmax	°C	500
			dskip		yes
			bvce	V	infinity
			vbefwd	V	0.2

MOSFET and JFET Models

The following information describes how various MOSFET models from Spectre are translated to the corresponding ADS models.

All Mosfet devices in Spectre reference a model by its instance name. Each Mosfet model in Spectre has a name value pair **type=**[**npn**|**pnp**], as well as a Level parameter.

The *type* value pair keyword is used to determine what device to place in the schematic, MOSFET_NMOS or MOSFET_PMOS. The *Level* is used to determine which model is placed and what value is set for *Idsmod*.

In the ADS netlist, the model is always called MOSFET, with the appropriate keywords NMOS and PMOS set to [0|1], and the parameter Idsmod set as specified in Table 8-9.

The only exception to this is the Mosfet device which refers to the Phillips MOS902 model. In this case, the device placed in the schematic will be MM9_NMOS or MM9_PMOS and the model will be MOS_Model9_Process. The netlist component is called MOS9. For both the MM9_NMOS and the MM9_PMOS, the translator sets the parameter Type=2 to indicate that it is a process-based model.

Table 8-9. Spectre Level Parameters Mapping (MOSFET)

Spectre Level	ADS Schematic Model	ADS Netlist Idsmod	ADS BSIM3 Version
0	Not Translated		
1	LEVEL1_Model	1	
15	Not Translated		
2	Not Translated		
3	Not Translated		
30	Not Translated		
3002	Not Translated		
705	Not Translated		
902	MOS_Model9_Process		
903	Not Translated		
bsim3v3	BSIM3_Model	8	3.22

Dependence Parameters

There are certain model parameters listed in the BSIM Models that reference additional parameters. These additional parameters are denoted in parenthesis using the letters L, W and/or P. As an example, the Dwg (L,W, P) parameter in the BSIM3 Model defines four separate parameters;

1. Dwg: Coefficient of Weff's gate dependence.
2. LDwg: Length dependence of Dwg.
3. WDwg: Width dependence of Dwg.
4. PDwg: Cross dependence of Dwg.

Each letter indicates a sensitivity parameter that exists in Spectre and ADS. These parameters are length(L), Width(W) and Cross(P). Refer to your Spectre and ADS component documentation for details.

The following MOSFET and JFET models are translated in ADS:

- "LEVEL1_Model:LEVEL 1 MOSFET Model" on page 8-38
- "MOS_Model9_Process:Philips MOS Model 9 (Process Based)" on page 8-46
- "MOS_Model9_Process:Philips MOS Model 9 (Process Based)" on page 8-46
- "JFET_Model:Junction Field Effect Transistor Model" on page 8-51

Binning Process

In order for ADS to translate the binning process, each binning definition is translated as a separate model.

Binning Example

The following is an example Spectre binning statement for a bsim model:

```
model ModelName ModelType {
   1: <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2: <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3: <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

The Netlist Translator separates each binning reference into an ADS model. The example would result in three models with the names ModelName_1, ModelName_2, and ModelName_3. The models are then tied together by creating a "BinModel" and configuring the appropriate min/max values.

LEVEL1_Model:LEVEL 1 MOSFET Model

The Spectre Level 1 MOSFET (mos1) model is translated to the ADS MOSFET LEVEL1_Model. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model nchmod1 mos1 vto=0.78 gamma=0.56 kp=0.8675e-4 tox=0.21e-7
nsub=0.21e17 ld=0.55e-6 capmod=yang vmax=4e5 theta=0.19 cbs=11e-15
cbd=10e-15 lambda=0.1
```

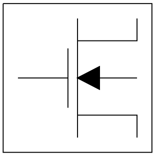
Spectre Netlist Syntax:

model *mname* **mos1 type=**[**npn**|**pnp**] [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=**[**0**|**1**] **PMOS=**[**0**|**1**] **Idsmod=1** [param=*value*]*

ADS Schematic Symbol:



LEVEL1_Model
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-10. MOSFET Level 1_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
NMOS		yes	type†		n (NMOS)
PMOS		no	type †		n (NMOS)
Capmod		1	capmod		bsim
Vto	V	0.0	vto	V	0.0
Kp	A/m^2	NMOS= 2.0718e-5 PMOS= 8.632e-5	kp	A/V^2	2.0718e-5
Gamma	√V	0.0	gamma	√ V	0.0
Phi	V	0.6	phi	V	0.7
Lambda	1/V	0.0	lamba	1/V	0.0
Rd	ohms	0.0	rd	ohms	0.0
Rs	ohms	0.0	rs	ohms	0.0
Cbd	F	0.0	cbd	F	0.0
Cbs	F	0.0	cbs	F	0.0
Is	A	1e-14	is	A	1e-14
Pb	V	0.8	pb	V	0.8
Cgso	F/m	0.0	cgso	F/m	0.0
Cgdo	F/m	0.0	cgdo	F/m	0.0
Cgbo	F/m	0.0	cgbo	F/m	0.0
Rsh	ohms/sq.	0.0	rsh	ohms/sq.	0.0
Cj	F/m^2	0.0	cj	F/m^2	0.0
Mj		0.5	mj		0.5
Cjsw	F/m	0.0	cjsw	F/m	0.0
Mjsw		0.33	mjsw		0.33
Js	A/m^2	0.0	js	A/m^2	0.0
Tox	m	1e-7	tox	m	1e-7

† If type = nmos then NMOS = yes and PMOS = no; if type = pmos then NMOS = no and PMOS = yes

†† Temp = trise + temp ("temp" is an ADS global variable)

Table 8-10. MOSFET Level 1_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Nsub	1/cm3	0.0	nsub	1/cm3	1.13e16
Nss	1/cm^2	0.0	nss	1/cm^2	0.0
Tpg		1	tpg		1
Ld	m	0.0	ld	m	0.0
Uo	cm^2/ (VxS)	600.0	uo	cm^2/ (VxS)	600.0
Kf		0.0	kf		0.0
Af		1.0	af		1.0
Fc		0.5	fc		0.5
Tnom	°C	25	tnom	°C	set by options
N		1.0	n		1.0
Imax	A	10.0	imax	A	1.0
wBvsub	V	infinite	bvj	V	infinite
wBvg	V	infinite	vbox	V	1e9 x tox
Parameters not in Spectre Model					
Temp [††]	°C	25			
Idsmod		1 (Level 1)			
Nlev		-1			
Gdwnoi		1			
Rg	ohms	0.0			
Rds	ohms	infinity			
Tt	sec	0.0			
Ffe		1.0			
wVsubfwd	V	infinity			
wBvds	V	infinite			
wIdsmax	V	infinite			

[†] If type = nmos then NMOS = yes and PMOS = no; if type = pmos then NMOS = no and PMOS = yes

[††] Temp = trise + temp ("temp" is an ADS global variable)

Table 8-10. MOSFET Level 1_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
wPmax	W	infinite			
AllParams					
			Parameters not in ADS Model		
			ld	m	
			vmax	m/s	infinity
			theta	1/V	0.0
			nfs	1/cm^2	0.0
			xw	m	0.0
			xl	m	0.0
			ai0	1/V	0.0
			lai0	µm/V	0.0
			wai0	µm/V	0.0
			bi0	V	0.0
			lbi0	µm x V	0.0
			wbi0	µm x V	0.0
			xpart		calculated
			xqc		0.0
			rsc	ohms	0.0
			rdc	ohms	0.0
			ldif	m	0.0
			hdif	m	0.0
			pbsw	V	0.8
			ute		–1.5
			tlev		0
			tlevc		0
			eg	eV	1.12452
			gap1	V/°C	7.02e-4

[†] If type = nmos then NMOS = yes and PMOS = no; if type = pmos then NMOS = no and PMOS = yes

[††] Temp = trise + temp ("temp" is an ADS global variable)

Table 8-10. MOSFET Level 1_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
			gap2	K	1108
			trs	1/°C	0.0
			trd	1/°C	0.0
			xti		3.0
			pta	V/°C	0.0
			ptp	V/°C	0.0
			cta	1/°C	0.0
			ctp	1/°C	0.0
			noisemod		1
			ef		1.0
			wmin	m	0.0
			wmax	m	1.0
			lmin	m	0.0
			lmax	m	1.0
			trise	°C	0.0
			wd	m	0.0
			meto	m	0.0
			rss	ohms x m	0.0
			rdd	ohms x m	0.0
			minr	ohms	0.1
			lgcs	m	0.0
			lgcd	m	0.0
			sc	m	infinity
			dskip		yes
			imelt	A	imax
			jmelt	A/m^2	jmax

† If type = nmos then NMOS = yes and PMOS = no; if type = pmos then NMOS = no and PMOS = yes

†† Temp = trise + temp ("temp" is an ADS global variable)

Table 8-10. MOSFET Level 1_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
			fcsw		0.5
			alarm		none
			jmax	A/m^2	1e8
			uto	°C	0.0
			flex		0.0
			lamex	1/°C	0.0
			ptc	V/°C	0.0
			tcv	V/°C	0.0
			ldd	m	0.0
			lds	m	0.0
			wnoi	m	1e-5
			degramod		spectre
			degradation		no
			dvthc	V	1.0
			dvthe		1.0
			duoc	Siemens	1.0
			duoe		1.0
			crivth	V	0.1
			criuo	%	10
			crigm	%	10
			criids	%	10
			wnom	m	5e-6
			lnom	m	1e-6
			vbsn	V	0.0
			vdsni	V	0.1
			vgsni	V	5.0

† If type = nmos then NMOS = yes and PMOS = no; if type = pmos then NMOS = no and PMOS = yes

†† Temp = trise + temp ("temp" is an ADS global variable)

Table 8-10. MOSFET Level 1_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
			vdsng	V	0.1
			vgsng	V	5.0
			esat	V/m	1.0e7
			esatg	1/m	2.5e6
			vpg		0.25
			vpb		-0.13
			subc1		2.24e-5
			subc2	1/V	-1.0e-6
			sube		6.4
			strc		1.0
			stre		1.0
			h0		1.0
			hgd	1/V	0.0
			m0		1.0
			mgd	1/V	0.0
			ecrit0	V/cm	1.1e5
			lecrit0	µm x V/cm	0.0
			wecrit0	µm x V/cm	0.0
			ecritg	1/cm	0.0
			lecritg	µm /cm	0.0
			wecritg	µm /cm	0.0
			ecritb	1/cm	0.0
			lecritb	µm /cm	0.0
			wecritb	µm /cm	0.0
			lc0		1.0
			llc0	µm	0.0

[†] If type = nmos then NMOS = yes and PMOS = no; if type = pmos then NMOS = no and PMOS = yes

[††] Temp = trise + temp ("temp" is an ADS global variable)

Table 8-10. MOSFET Level 1_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
			wlc0	µm	0.0
			lc1		1.0
			llc1	µm	0.0
			wlc1	µm	0.0
			lc2		1.0
			llc2	µm	0.0
			wlc2	µm	0.0
			lc3		1.0
			llc3	µm	0.0
			wlc3	µm	0.0
			lc4		1.0
			llc4	µm	0.0
			wlc4	µm	0.0
			lc5		1.0
			llc5	µm	0.0
			wlc5	µm	0.0
			lc6		1.0
			llc6	µm	0.0
			wlc6	µm	0.0
			lc7		1.0
			llc7	µm	0.0
			wlc7	µm	0.0

† If type = nmos then NMOS = yes and PMOS = no; if type = pmos then NMOS = no and PMOS = yes

†† Temp = trise + temp ("temp" is an ADS global variable)

MOS_Model9_Process:Philips MOS Model 9 (Process Based)

The Spectre mos902 model is translated to the ADS MOSFET MOS_Model9_Process. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model mos9pch mos902 ler=0.93e-6 wer=20e-6 tref=27 vtor=1.11 kr=0.54
phibr=0.66 vsbxr=100 their=0.19 slk=-0.215e-6 swk=98e-9 swthe3=7.8e-9
```

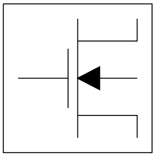
Spectre Netlist Syntax:

model *mname* **mos902 type=**[**p**|**n**] [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=[0|1] PMOS=[0|1] Type=2** [param=*value*]*

ADS Schematic Symbol:



MOS_Model9_Process
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-11. MOS Model9 Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
NMOS †		yes	type		n
PMOS		no	type		n
Type		2			

Table 8-11. MOS Model9 Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Ler	m	1e–4	ler	m	2.5e–6
Wer	m	1e–4	wer	m	25e–6
Lvar	m	0.0	lvar	m	0.3e–6
Lap	m	0.0	lap	m	0.1e–6
Wvar	m	0.0	wvar	m	3e–6
Wot	m	0.0	wot	m	1e–6
Tr	°C	300.15	tr tref	°C	set by option
Vtor	V	0.87505	vtor	V	0.8
Stvto	V/K	0.0	stvto	V/K	0.01
Slvto	V×m	0.0	slvto	V×m	0.5e–6
Sl2vto	V×m^2	0.0	sl2vto	V×m^2	0.0
Swvto	V m	0.0	Swvto	V m	5e–6
Kor	√V	0.74368	kor	V	0.5
Slko	√(V×m)	0.0	slko	V×m	1e–6
Swko	√V	0.0	swko	V×m	10e–6
Kr	√V	0.55237	kr	V	0.1
Slk	√(V×m)	0.0	slk	V×m	0.5e–6
Swk	√(V×m)	0.0	swk	V×m	5e–6
Phibr	V	0.65	phibr	V	0.65
Vsbxr	V	0.63304	vsbxr	V	0.9
Slvsbx	V×m	0.0	slvsbx	V×m	0.5e–6
Swvsbx	V×m	0.0	swvsbx	V×m	5e–6
Betaq	A/V^2	0.12069e–3	betaq	A/V^2	0.1e–3
Etabet		0.0	etabet		0.5
Thelr	1/V	0.99507e–1	thelr	1/V	0.05
Stthelr	1/(V×K)	0.0	stthelr	1/(V×K)	3e–3

Table 8-11. MOS Model9 Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Slthe1r	m/V	0.0	slthelr	m/V	50e–9
Stlthe1	m/(V×K)	0.0	stlthel	m/(V×K)	5e–9
Swthe1	m/V	0.0	swthe1	m/V	1e–6
Wdog	m	0.0	wdog	m	0.0
Fthe1		0.0	fthe1		0.0
The2r	$\sqrt{V}$	0.43225e–1	the2r	1/V	17e–3
Stthe2r	$\sqrt{(V/K)}$	0.0	stthe2r	1/(V×K)	0.1e–3
Slthe2r	$m/\sqrt{V}$	0.0	slthe2r	m/V	5e–9
Stlthe2	$m/\sqrt{(V/K)}$	0.0	stlthe2	m/(V×K)	0.5e–9
Swthe2	$m/\sqrt{V}$	0.0	swthe2	m/V	0.1e–6
The3r	1/V	0.0	the3r	1/V	37e–3
Stthe3r	1/V/K	0.0	stthe3r	1/(V×K)	0.1e–3
Slthe3r	m/V	0.0	slthe3r	m/V	5e–9
Stlthe3	m/V/K m/(V×K)?	0.0	stlthe3	m/(V×K)	0.5e–9
Swthe3	m/V	0.0	swthe3	m/V	0.1e–6
Gam1r	$V^{(1-Etads)}$	0.38096e–2	gam1r	$V^{(1-etads)}$	40e–3
Slgam1	$V^{(1-Etads)}$ ×m	0.0	slgam1	$V^{(1-etads)}$ ×m	0.1e–6
Swgam1	$V^{(1-Etads)}$ ×m	0.0	swgam1	$V^{(1-etads)}$ ×m	1e–6
Etadsr		0.6	etadsr		0.6
Alpr		0.1e–1	alpr		4e–3
Etaalp		0.0	etaalp		0.5
Slalp	$m^{(Etaalp)}$	0.0	slalp	$m^{(etaalp)}$	0.14e–3
Swalp	m	0.0	swalp	m	0.1e–6

Table 8-11. MOS Model9 Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Vpr	V	0.67876e1	vpr	V	0.25
Gamoor		0.29702 e–4	gamoor		1.1e–3
Slgamoo	m^2	0.0	slgamoo	m^2	10e–15
Etagamr		2.0	etagamr		2.0
Mor		0.44	mor		0.3
Stmo	1/K	0.0	stmo	1/K	0.01
Slmo	$\sqrt{m}$	0.0	slmo	$\sqrt{m}$	1.4e–3
Etamr		2.0	etamr		2.0
Zet1r		0.2015e1	zet1r		0.7
Etazet		0.0	etazet		0.5
Slzet1	$m^{(Etazet)}$	0.0	slzet1	$m^{(etazet)}$	0.14e–6
Vsbtr	V	0.61268e1	vsbtr	V	99
Slvsbt	m×V	0.0	slvsbt	m×V	10e–6
A1r		0.20348e2	a1r		22
Sta1	1/K	0.0	sta1	1/K	0.1
Sla1	m	0.0	Sla1	m	10e–6
Swa1	m	0.0	swa1	m	0.1e–3
A2r	V	0.33932e2	a2r	V	33
Sla2	m×V	0.0	sla2	m×V	10e–6
Swa2	m×V	0.0	swa2	m×V	0.1e–3
A3r		0.10078e1	a3r		0.6
Sla3	m	0.0	sla3	m	1e–6
Swa3	m	0.0	swa3	m	10e–6
Tox	m	1e–6	tox	m	20e–9
Col	F/m	0.0	col	F/m	50e–12
Ntr	J	0.0	ntr	J	21e–21
Nfr	V^2	0.0	nfr	V^2	16e–12

Table 8-11. MOS Model9 Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Parameters not in Spectre Model					
Nfar	1/(V×m)	7.15e22			
Nfbr	1/(V×m^2)	2.16e7			
Nfcr	1/V	0.0			
Vr	V	0.0			
Jsgbr	A/m^2	1e–14			
Jsdbr	A/m^2	1e–14			
Jsgsr	A/m	1e–14			
Jsdsr	A/m	1e–14			
Jsggr	A/m	1e–14			
Jsdgr	A/m	1e–14			
Cjbr	F/m^2	0.0			
Cjsr	F/m	0.0			
Cjgr	F/m	0.0			
Vdbr	V	0.8			
Vdsr	V	0.8			
Vdgr	V	0.8			
Pb		0.5			
Ps		0.5			
Pg		0.5			
Nb		1.0			
Ns		1.0			
Ng		1.0			
			Parameters not in ADS Model		
			tnom	°C	set by option
			dta	K	0.0
			trise	K	0.0

JFET_Model:Junction Field Effect Transistor Model

The Spectre jfet model is translated to the ADS JFET_Model. For translation information on the JFET device, refer to "JFET Device" on page 7-22.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example Spectre Command Line:

```
model jmod jfet beta=9e-5 lambda=0 type=n vt0=-18.7 rd=10 rs=10
cgs=1.3e-13 pb=0.65
```

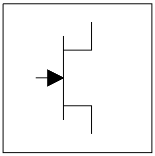
Spectre Netlist Syntax:

model *mname* **jfet type=[n|p]** [param=*value*]*

ADS Netlist Syntax:

model *mname* **JFET NFET=[0|1] PFET=[0|1]** [param=*value*]*

ADS Schematic Symbol:



JFET_Model
NFET=yes
PFET=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-12. JFET_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
NFET		yes	type		n (NFET)
PFET		no	type		n (NFET)
Vto	V	–2.0	vto	V	–2.0

Table 8-12. JFET_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
Beta	A/V^2	1E-4	beta	A/V^2	1E-4
Lambda	1/V	0.0	lambda	1/V	0.0
Rd	Ohms	0.0	rd	Ohms	0.0
Rs	Ohms	0.0	rs	Ohms	0.0
Is	A	1E-14	is	A	1E-14
Cgs	F	0.0	cgs	F	0.0.
Cgd	F	0.0	cgd	F	0.0
Pb	V	1.0	pb	V	1.0
Fc		0.5	fc		0.5
Tnom	°C	25	tnom	°C	Set by defaults
see *Temp* instance parameter)			trise[†]	°C	0.0
Kf		0.0	kf		0.0
Af		1.0	af		1.0
Imax	A	1.6	imax	A	1.0
N		1.0	n		1.0
Alpha	1/V	0.0	ai	1/V	0.0
Vk	V	0.0	bi	V	0.0
Vtotc	V/°C	0.0	tcv	1/°C	0.0
Xti		3.0	xti		3.0
wBvgs	V	infinity	bvj	V	infinity
wBvgd	V	infinity	bvj	V	infinity
Parameters not in Spectre Model					
Isr	A	0.0			
Nr		2.0			
M		0.5			
Ffe					
Betatce	%/°C	0.0			

Table 8-12. JFET_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
wVgfwd	V	infinity			
wBvds	V	infinity			
wIdsmax	A	infinity			
wPmax	W	infinity			
Secured					
			Parameters not in ADS Model		
			level		1
			lambda1	1/V	0.0
			rg	Ohms	0.0
			rb	Ohms	0.0
			minr	Ohms	0.1
			tt	Sec	0.0
			mj		0.5
			kfd		0.0
			afg		1.0
			imelt	A	imax
			dskip		yes
			np		2.0
			alpha		2.0
			io	A	0.0
			ns		1.0
			tlev		0.0
			tlevc		0.0
			eg	V	1.12452
			gap1	V/°C	7.02E-4
			gap2	°C	1108
			lto	°C	0.0
			lte		0.0
			tc1	1/°C	0.0

Table 8-12. JFET_Model Parameter Mapping

ADS Name	Unit	Default	Spectre Name	Unit	Default
			tc2	1/°C^2	0.0
			alarm		none
			vtop	V	0.6
			vtos	V	1.2
			vtoe	V	0.33
			vtoc	V	3.3
			isb	A	1E-14
			nb		1.0
			cgbs	F	0.0
			cgbd	F	0.0
			mjb		0.5
			pbb	V	1.0
			bto	°C	0.0
			bte		0.0

MOSFET Models

The following information describes how the various MOSFET models from SPICE are translated to the corresponding ADS models.

All Mosfet devices in SPICE reference a model by its instance name. Each Mosfet model in SPICE has a keyword NMOS or PMOS, as well as a Level parameter.

The NMOS/PMOS keyword is used to determine what device to place in the schematic, MOSFET_NMOS or MOSFET_PMOS. The *Level* is used to determine which model is placed and what value is set for *Idsmod*.

In the ADS netlist, the model is always called MOSFET, with the appropriate keywords NMOS and PMOS set to [0|1], and the parameter Idsmod set as specified in Table 8-13.

The only exception to this is the Mosfet device which refers to an HSpice Level 50 model, the Phillips MOS9 model. In this case, the device placed in the schematic will be MM9_NMOS or MM9_PMOS and the model will be MOS_Model9_Process. The netlist component is called MOS9. For both the MM9_NMOS and the MM9_PMOS, the translator sets the parameter Type=2 to indicate that it is a process-based model.

Table 8-13. SPICE Level Parameters Mapping Table (MOSFET)

Spice2/3 Level	PSpice Level	HSpice Level	ADS Schematic Model	ADS Netlist Idsmod	ADS BSIM3 Version
1	1	1	LEVEL1_Model	1	
2	2	2	LEVEL2_Model	2	
3	3	3	LEVEL3_Model	3	
4	4	13, 28	BSIM1_Model	4	
5	-	39	BSIM2_Model	5	
	7		BSIM3_Model	8	3.1
		49, 53	BSIM3_Model	8	legal BSIM3 Versions are: 3.0, 3.1, 3.2, 3.21, and 3.22

Dependence Parameters

There are certain model parameters listed in the BSIM Models that reference additional parameters. These additional parameters are denoted in parenthesis using the letters L, W and/or P. As an example, the Dwg (L,W, P) parameter in the BSIM3 Model defines four separate parameters;

1. Dwg: Coefficient of Weff’s gate dependence.
2. LDwg: Length dependence of Dwg.
3. WDwg: Width dependence of Dwg.
4. PDwg: Cross dependence of Dwg.

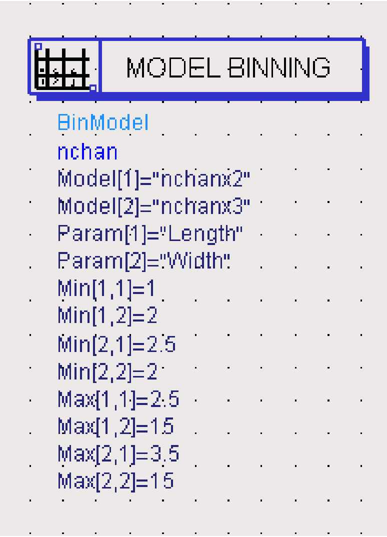
Each letter indicates a sensitivity parameter that exists in HSpice and ADS. These parameters are length(L), Width(W) and Cross(P). Refer to your HSpice and ADS component documentation for details.

HSpice Automatic Model Selection

Automatic Model Selection, also known as *binning*, is an HSpice feature that allows for the definition of a library of Mosfets over a range of lengths and widths. The following netlist fragment is an example of automatic model selection in HSpice. It is currently used only for Mosfets. Notice that the model name is a root name plus an extension, separated by a period (that is: NCHAN.2). Also note the special model parameters, LMIN, LMAX, WMIN and WMAX. They are used only for automatic model selection.

```
.OPTION WL

M1 1 2 3 4 NCHAN 10 2
M2 1 2 3 4 NCHAN 10 3

$$$$$$$ FOR CHANNEL LENGTH SELECTION
.MODEL NCHAN.2 NMOS LEVEL=2 VTO=2.0 UO=800 TOX=500
+NSUB=1E15
+ RD=10 RS=10 CAPOP=5
+ LMIN=1 LMAX=2.5 WMIN=2 WMAX=15

.MODEL NCHAN.3 NMOS LEVEL=2 VTO=2.2 UO=800 TOX=500
+NSUB=1E15
+ RD=10 RS=10 CAPOP=5
+ LMIN=2.5 LMAX=3.5 WMIN=2 WMAX=15
```

The ADS equivalent of this functionality is provided by a component called *BinModel*. Since ADS component names cannot include a period, the models are renamed nchanx2 and nchanx3. Then a BinModel component is placed, which lists the models by name, as well as listing the special length and width parameter ranges, as defined by LMIN, LMAX, WMIN and WMAX. The name of the BinModel component is the same as the root name of the models, in this case nchan. The ranges of all models with the same root name will be recorded on one BinModel component as shown in Figure 8-1.



BinModel
nchan
Model[1]="nchanx2"
Model[2]="nchanx3"
Param[1]="Length"
Param[2]="Width"
Min[1,1]=1
Min[1,2]=2
Min[2,1]=2.5
Min[2,2]=2
Max[1,1]=2.5
Max[1,2]=15
Max[2,1]=3.5
Max[2,2]=15

Figure 8-1. The ADS BinModel Component

Also note that the Mosfet devices that are placed in the schematic do not refer to the Mosfet models as they normally would. Instead they refer to the BinModel. During simulation, the length and width specified on the device are sent to the BinModel and checked against the ranges specified for each model, to determine which model to use. If the length and width on the device fall within the range of the length and width on more than one model, the first one that matches is used.

LEVEL1_Model:LEVEL 1 MOSFET Model

The SPICE Level 1 MOSFET model is translated to the ADS MOSFET LEVEL1_Model. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model nch nmos level=1 vto=0 kp=2e-5 ...
```

SPICE Dialect and Netlist Syntax:

Spice2/3: **.model** *mname* **NMOS | PMOS LEVEL=1** [param=*value*]*

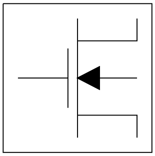
PSpice: **.model** *mname* **NMOS | PMOS LEVEL=1** [param=*value*]*

HSpice: **.model** *mname* **NMOS | PMOS LEVEL=1** [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=[0|1] PMOS=[0|1] Idsmod=1** [param=*value*]*

ADS Schematic Symbol:



LEVEL1_Model
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-14. LEVEL1_Model Parameters

ADS	**Unit**	**Default**	**Spice2/3**	**PSpice**	**HSpice**
Idsmod		1	Level=1	Level=1	Level=1
Capmod		1			
Vto	V	0.0	x	x	VT
Kp	A/m^2	NMOS= 2.0718e-5 PMOS= 8.632e-5	x	x	BET, BETA
Gamma	$\sqrt{V}$	0.0	x	x	.5276
Phi	V	0.6	x	x	.576 V
Lambda	1/V	0.0	x	x	LAM, LA
Rd	ohms	0.0	x	x	-
Rs	ohms	0.0	x	x	-
Cbd	F	0.0	x	x	-
Cbs	F	0.0	x	x	-
Is	A	1e-14	x	x	-
Pb	V	0.8	x	x	-
Cgso	F/m	0.0	x	x	-
Cgdo	F/m	0.0	x	x	-
Cgbo	F/m	0.0	x	x	-
Rsh	ohms/sq.	0.0	x	x	-
Cj	F/m^2	0.0	x	x	-
Mj		0.5	x	x	-
Cjsw	F/m	0.0	x	x	-
Mjsw		0.33	Spice3 (0.50)	x	-
Js	A/m^2	0.0	x	x	-
Tox	m	1e-7	x	x	x
Nsub			x	x	DNB, NB (1e15)
Nss	1/Cm^2	0.0	x	x	-
Tpg		1	x	x	-
Ld	m	0.0	x	x	DLAT, LATD

Table 8-14. LEVEL1_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Uo	Cm^2/(VxS)	600.0	x	x	x
Kf		0.0	x	x	-
Af		1.0	x	x	-
Fc		0.5	x	x	-
N		1.0	-	x	-
Tt	sec	0.0	-	x	-

Comments:

ADS does not support NRD, RDC, NRS or RSC MOSFET parameters. If RD or RS are not given, they will be calculated as follows:

RD=NRD*RSH+RDC

RS=NRS*RSH+RSC

LEVEL2_Model:LEVEL 2 MOSFET Model

The SPICE Level 2 MOSFET model is translated to the ADS MOSFET LEVEL2_Model. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model nch nmos level=2 vto=0 kp=2e-5 ...
```

SPICE Dialect and Netlist Syntax:

Spice2/3: **.model** *mname* **NMOS | PMOS LEVEL=2** [param=*value*]*

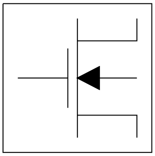
PSpice: **.model** *mname* **NMOS | PMOS LEVEL=2** [param=*value*]*

HSpice: **.model** *mname* **NMOS | PMOS LEVEL=2** [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=[0|1] PMOS=[0|1] Idsmod=2** [param=*value*]*

ADS Schematic Symbol:



LEVEL2_Model
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-15. LEVEL2_Model Parameters

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Idsmod		2	Level=2	Level=2	Level=2
Capmod		1			
Vto	V	0.0	x	x	VT
Kp	A/V^2	2e-5	x	x	BET, BETA
Gamma	$\sqrt{V}$	0.0	x	x	(.5276)
Phi	V	0.6	x	x	(.576)
Lambda	1/V	0.0	x	x	LAM, LA
Rd	ohms	0.0	x	x	-
Rs	ohms	0.0	x	x	-
Cbd	F	0.0	x	x	-
Cbs	F	0.0	x	x	-
Is	A	1e-14	x	x	-
Pb	V	0.8	x	x	-
Cgso	F/m	0.0	x	x	-
Cgdo	F/m	0.0	x	x	-
Cgbo	F/m	0.0	x	x	-
Rsh	ohms/sq	0.0	x	x	-
Cj	F/m^2	0.0	x	x	-
Mj		0.5	x	x	-
Cjsw	F/m	0.0	x	x	-
Mjsw		0.33	x	x	-
Js	A/m^2	0.0	x	x	-
Tox	m	1e-7	x	x	x
Nsub			x	x	DNB, NB (1e15)
Nss	$1/Cm^2$	0.0	x	x	-
Nfs	$1/Cm^2$	0.0	x	x	DFS, NF, DNF
Tpg		1	x	x	-
Xj	m	0.0	x	x	x

Table 8-15. LEVEL2_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Ld	m	0.0	x	x	DLAT, LATD
Uo	$Cm^2/(V\times s)$	600.0	x	x	UB, UBO
Ucrit	V/Cm	1e4	x	x	x
Uexp		0.0	x	x	F2
Vmax	m/s	0.0	x	x	VMX, VSAT
Neff		1.0	x	x	-
Xqc		1.0	-	x	-
Kf		0.0	x	x	-
Af		1.0	x	x	-
Fc		0.5	x	x	-
Delta		0.0	x	x	x
Rg	ohms	0.0	-	x	-
Rds	ohms	infinity	-	x	-
N		1.0	-	x	-
Tt	sec	0.0	-	x	-

Comments:

ADS does not support NRD, RDC, NRS or RSC MOSFET parameters. If RD or RS are not given, they will be calculated as follows:

RD=NRD*RSH+RDC

RS=NRS*RSH+RSC

LEVEL3_Model:LEVEL 3 MOSFET Model

The SPICE Level 3 MOSFET model is translated to the ADS MOSFET LEVEL3_Model. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model nch nmos level=3 vto=0 kp=2e-5 ...
```

SPICE Dialect and Netlist Syntax:

Spice2/3: **.model** *mname* **NMOS | PMOS LEVEL=3** [param=*value*]*

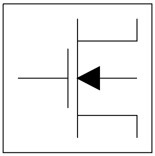
PSpice: **.model** *mname* **NMOS | PMOS LEVEL=3** [param=*value*]*

HSpice: **.model** *mname* **NMOS | PMOS LEVEL=3** [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=[0|1] PMOS=[0|1] Idsmod=3** [param=*value*]*

ADS Schematic Symbol:



LEVEL3_Model
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-16. LEVEL3_Model Parameters

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Idsmod		3	Level=3	Level=3	Level=3
Capmod		1			
Vto	V	0.0	x	x	VT
Kp	A/V^2	2e-5	x	x	BET, BETA
Gamma	$\sqrt{V}$	0.0	x	x	(.5276)
Phi	V	0.6	x	x	(.576)
Rd	ohms	0.0	x	x	-
Rs	ohms	0.0	x	x	-
Cbd	F	0.0	x	x	-
Cbs	F	0.0	x	x	-
Is	A	1e-14	x	x	-
Pb	V	0.8	x	x	-
Cgso	F/m	0.0	x	x	-
Cgdo	F/m	0.0	x	x	-
Cgbo	F/m	0.0	x	x	-
Rsh	ohms/sq	0.0	x	x	-
Cj	F/m^2	0.0	x	x	-
Mj		0.5	x	x	-
Cjsw	F/m	0.0	x	x	-
Mjsw		0.33	x	x	-
Js	A/m^2	0.0	x	x	-
Tox	m	1e-7	x	x	x
Nsub			x	x	DNB, NB (1e15)
Nss	$1/cm^2$	0.0	x	x	-
Nfs	$1/cm^2$	0.0	x	x	DFS, NF, DNF
Tpg		1	x	x	-
Xj	m	0.0	x	x	x
Ld	m	0.0	x	x	DLAT, LATD

Table 8-16. LEVEL3_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Uo	$cm^2/(V \times s)$	600.0	x	x	UB, UBO (PMOS def=250)
Vmax	m/s	0.0	x	x	VMX
Xqc		1.0	-	x	-
Kf		0.0	x	x	-
Af		1.0	x	x	-
Fc		0.5	x	x	-
Delta		0.0	x	x	x
Theta	1/V	0.0	x	x	x
Eta		0.0	x	x	x
Kappa		0.2	x	x	x
Rg	ohms	0.0	-	x	-
Rds	ohms	infinity	-	x	-
N		1.0	-	x	-
Tt	sec	0.0	-	x	-

Comments:

ADS does not support NRD, RDC, NRS or RSC MOSFET parameters. If RD or RS are not given, they will be calculated as follows:

RD=NRD*RSH+RDC

RS=NRS*RSH+RSC

BSIM1_Model:BSIM1 MOSFET Model

The SPICE BSIM1 MOSFET model is translated to the ADS MOSFET BSIM1_Model. For translation information on the MOSFET device, refer to “MOSFET Device” on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model nch nmos level=13 rsh=0 js=0 ...
```

SPICE Dialect and Netlist Syntax:

Spice2/3: **.model** *mname* **NMOS | PMOS LEVEL=4** [param=*value*]*

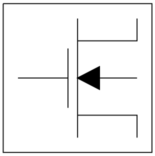
PSpice: **.model** *mname* **NMOS | PMOS LEVEL=4** [param=*value*]*

HSpice: **.model** *mname* **NMOS | PMOS LEVEL=13** [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=[0|1] PMOS=[0|1] Idsmod=4** [param=*value*]*

ADS Schematic Symbol:



BSIM1_Model
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to “Using Parameter Mapping Tables to Understand a Translation” on page 8-2.

Table 8-17. BSIM1_Model Parameters

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Idsmod		4	Level=4	Level=4	Level=13, 28
Rsh	ohms/sq	0.0	x	x	Level 13 only RSHM
Js	A/m^2	0.0	x	x	Level 13 only IJS
Temp	°C	25	x	x	-
Muz	cm^2/Vsec	600	x	x	x
Dl		0.0	x	x	XL, LDEL
Dw		0.0	x	x	XW, WDEL
Vdd	V	5.0	x	x	-
Vfb (L,W)	V	–0.3	x	x	VFB0
Phi (L,W)	V	0.6	x	x	PHI0 (0.7)
K1 (L,W)	$\sqrt{V}$	0.5	x	x	x
K2 (L,W)		0.0	x	x	x
Eta (L,W)		0.0	x	x	ETA0
U0 (L,W)	1/V	670.0	x	x	U00 (0.0 1/V)
U1 (L,W)	μm/V	0.0	x	x	x
X2mz (L,W)	cm^2/V^2	0.0	x	x	X2M
X2e (L,W)	1/V	-0.07	x	x	x
X3e (L,W)	1/V	0.0	x	x	x
X2u0 (L,W)	1/V^2	0.0	x	x	x
X2u1 (L,W)	μm/V^2	0.0	x	x	x
X3u1 (L,W)	μm/V^2	0.0	x	x	x
Mus (L,W)	cm^2/Vs	1082	x	x	Level 13 only
X2ms (L,W)	cm^2/V^{2}s	0.0	x	x	Level 13 only
X3ms (L,W)	cm^2/V^{2}s	0.0	x	x	x
N0 (L,W)		0.5	x	x	x
Nb (L,W)	1/V	0.0	x	x	NB0

Table 8-17. BSIM1_Model Parameters (continued)

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Nd (L,W)	1/V	0.0	x	x	ND0
Tox	μm	1e-7	x	x	TOXM
Cj	F/m^2	0.0	x	x	Level 13 only CJM
Mj		0.5	x	x	Level 13 only MJ0
Cjsw	F/m	0.0	x	x	Level 13 only CJW
Mjsw		0.33	x	x	Level 13 only MJW
Pb	V	0.8	x	x	Level 13 only PJ
Pbsw	V	1.0	x (Def PB)	x	Pjw, Php
Cgso	F/m	0.0	x	x	CGSOM (1.5e-9)
Cgdo	F/m	0.0	x	x	CGDOM (1.5e-9)
Cgbo	F/m	0.0	x	x	CGBOM (2.0e-10)
Xpart		1.0	x	x	x
Kf		0.0	-	x	-
Af		1.0	-	x	-
Rg	ohms	0	-	x	-
N		1.0	-	x	-

Comments:

Tox > 1 is in angstroms in HSpice. Translator will convert it to um.

ADS does not support NRD, RDC, NRS or RSC MOSFET parameters. If RD or RS are not given, they will be calculated as follows:

RD=NRD*RSH+RDC

RS=NRS*RSH+RSC

BSIM2_Model:BSIM2 MOSFET Model

The SPICE BSIM2 MOSFET model is translated to the ADS MOSFET BSIM2_Model. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model nch nmos level=39 rsh=0 js=0
```

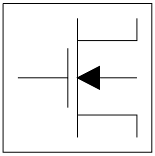
SPICE Dialect and Netlist Syntax:

Spice2/3:	**.model** *mname* **NMOS | PMOS LEVEL=5** [param=*value*]*
PSpice:	Does not exist
HSpice:	**.model** *mname* **NMOS | PMOS LEVEL=39** [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=[0|1] PMOS=[0|1] Idsmod=5** [param=*value*]*

ADS Schematic Symbol:



BSIM2_Model
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-18. BSIM2_Model Parameters

ADS	Unit	Default	Spice2/3	HSpice
Idsmod		5	Level=5	Level=39
Rsh		0.0	x	x
Js	A/m^2	0.0	x	x
Mu0	cm^2/V-s	600	x	Def=400
Dl	µm	0.0	x	x
Dw	µm	0.0	x	x
Vdd	V	5.0	x	x
Vgg	V	5.0	x	x
Vbb	V	-5.0	x	x
Temp	^{o}C	25	x	-
Tox	µm	1e-7	x	x
Cj	F/m^2	5.0	x	Def=0
Mj		0.5	x	x
Cjsw	F/m	0.0	x	x
Mjsw		0.33	x	x
Pb	V	0.8	x	x
Pbsw	V	1.0	x	Def=PB
Cgso	F/m	0.0	x	x
Cgdo	F/m	0.0	x	x
Cgbo		0.0	x	x
Xpart		1.0	x	-
Vfb (L, W)	V	-0.1	x	Def=-0.3
Phi (L, W)	V	0.6	x	Def=0.8
K1 (L, W)	√ V	0.5	x	x
K2 (L, W)		0.0	x	x
Eta0 (L, W)		0.08	x	Def=0
Ua0 (L, W)	1/V	670.0	x	Def=0
U10 (L, W)	µm/V	0.0	x	x

Table 8-18. BSIM2_Model Parameters

ADS	Unit	Default	Spice2/3	HSpice
Mu0b (L, W)	cm^2/V^2s	0.0	x	x
Etab (L, W)	1/V	-0.07	x	Def=0
Uab (L, W)	$1/V^2$	0.0	x	x
U1b (L, W)	$\mu m/V^2$	0.0	x	x
U1d (L, W)	$\mu m/V^2$	0.0	x	x
Mus0 (L, W)	cm^2/Vs	600.0	x	x
Musb (L, W)	cm^2/V^2s	0.0	x	x
N0 (L, W)		0.5	x	x
Nb (L, W)	1/V	1.0	x	x
Nd ((L, W)	1/V	0.0	x	x
Mu20 (L, W)		0.0	x	x
Mu2b (L, W)	1/V	0.0	x	x
Mu2g (L, W)	1/V	0.0	x	x
Mu30 (L, W)	cm^2/V^2s	0.0	x	x
Mu3b (L, W)	cm^2/V^3s	0.0	x	x
Mu3g (L, W)	cm^2/V^3s	0.0	x	x
Mu40 (L, W)	cm^2/v^3s	0.0	x	x
Mu4b (L, W)	cm^2/v^4s	0.0	x	x
Mu4g (L, W)	cm^2/v^4s	0.0	x	x
Ub0 (L, W)	$1/V^2$	0.0	x	x
Ubb (L, W)	$1/V^3$	0.0	x	x
Vof0 (L, W)	V	0.0	x	x
Vofb (L, W)	1/V	0.0	x	x
Vofd (L, W)	1/V	0.0	x	x
Ai0 (L, W)		0.0	x	x
Aib (L, W)	1/V	0.0	x	x
Bi0 (L, W)	V	0.0	x	x

Table 8-18. BSIM2_Model Parameters

ADS	Unit	Default	Spice2/3	HSpice
Bib (L, W)		0.0	x	x
Vghigh (L, W)	V	0.0	x	x
Vglow (L, W)	V	-0.15	x	Def=0

Comments:

Tox > 1 is in angstroms in HSpice. Translator will convert it to um.

For HSpice and ADS, most parameters have associated width and length sensitivity parameters. Consult your HSpice manual or refer to the ADS *Circuit Components* manual for more information.

ADS does not support NRD, RDC, NRS or RSC MOSFET parameters. If RD or RS are not given, they will be calculated as follows:

RD=NRD*RSH+RDC

RS=NRS*RSH+RSC

BSIM3_Model:BSIM3 MOSFET Model

The SPICE BSIM3 MOSFET model is translated to the ADS MOSFET BSIM3_Model. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model nch nmos level=7 rg=0 rsh=0 ...
```

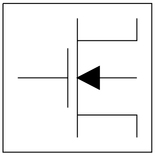
SPICE Dialect and Netlist Syntax:

Spice2/3:	Does not exist
PSpice:	**.model** *mname* **NMOS | PMOS LEVEL=7** [param=*value*]*
HSpice:	**.model** *mname* **NMOS | PMOS LEVEL=49** [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=[0|1] PMOS=[0|1] Idsmod=8** [param=*value*]*

ADS Schematic Symbol:



BSIM3_Model
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-19. BSIM3_Model Parameters

ADS	Units	Default	PSpice	HSpice
Idsmod		8	Level=7	Level=49, 53
Version		3.1	x	x
Mobmod		1	x	x
Capmod		1	Def=2	3 Level 53 def=3
Noimod		1	x	x
B3qmod		0	-	See comments
Paramchk		0	x	x
Binunit		1	x	x
Rg	ohms	0	x	-
Rsh	ohms/sq	0.0	x	x
Nj		1.0	x	N
Xti		3.0	-	x
Js	A/m^2	1e-4	x	Def=0
Jsw	A/m^2	0.0	x	x
Lint	m	0.0	x	x
Ll	m^{Lln}	0.0	x	x
Lln		1.0	x	x
Lw	m^{Lwn}	0.0	x	x
Lwn		1.0	x	x
Lwl	$m^{(Lwn+Lln)}$	0.0	x	x
Wint	m	0.0	x	x
Wl	m^{Wln}	0.0	x	x
Wln		1.0	x	x
Ww	m^{Wwn}	0.0	x	x
Wwn		1.0	x	x
Wwl	$m^{(Wwn+Wln)}$	0.0	x	x
Tnom	°C	25	Def=27	-

Table 8-19. BSIM3_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Tox	m	1.5e-8	x	x
Toxm	m	Tox	-	x
Cj	F/m^2	5e-4	x	Def=5.79e-4
Mj		0.5	x	x
Cjsw	F/m	5.0e-10	x	Def=0
Mjsw		0.33	x	x
Pb	V	1.0	x	PHIB
Pbsw	V	1.0	x	x
Xt (L,W,P)	m	1.55e-7	x	x
Vbm (L,W,P)	V	–5	Def=-3.0	Def=-3
Vbx (L,W,P)	V		x	x
Vfbcv (L,W,P)	V	-1	x	x
Vfb (L,W,P)	V	calculated	-	x
Xj (L,W,P)	m	0	x	Def=1.5e-7
Dwg (L,W,P)	m/V	0.0	x	x
Dwb (L,W,P)	m/V$^{(1/2)}$	0.0	x	x
Nch (L,W,P)	1/cm^3	1.7e17	-	x
Nsub (L,W,P)	1/cm^3	6.0e16	-	x
Ngate (L,W,P)	1/cm^3		x	x
Gamma1 (L,W,P)	V$^{(1/2)}$		x	x
Gamma2 (L,W,P)	V$^{(1/2)}$		x	x
Alpha0 (L,W,P)	m/V	0.0	x	x
Alpha1 (L,W,P)	1/V	0.0	-	x
Acde (L,W,P)	m/V	1.0	-	x
Moin (L,W,P)	m/V	15	-	x
Tpb	V/K	0.0	-	x
Tpbsw	V/K	0.0	-	x
Tpbswg	V/K	0.0	-	x

Table 8-19. BSIM3_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Tcj	V/K	0.0	-	x
Tcjsw	V/K	0.0	-	x
Tcjswg	V/K	0.0	-	x
Llc		Ll	-	x
Lwc		Lw	-	x
Lwlc		Lwl	-	x
Wlc		Wl	-	x
Wwc		Ww	-	x
Wwlc		Wwl	-	x
Elm (L,W,P)		5.0	x	x
Beta0 (L,W,P)	V	30.0	x	x

Vth0 (L,W,P)	V		NMOS=0.7
			PMOS=-0.7
				0.7

K1 (L,W,P)	$V^{(1/2)}$		x	0.50
K2 (L,W,P)			x	0.0186
K3 (L,W,P)		80.0	x	x
K3b (L,W,P)	1/V	0.0	x	x
W0 (L,W,P)	m	2.5e-6	-	x
Nlx (L,W,P)	m	1.74e-7	x	x
Dvt0 (L,W,P)		2.2	x	x
Dvt1 (L,W,P)		0.53	x	x
Dvt2 (L,W,P)	1/V	–0.032	x	-
Dvt0w (L,W,P)	1/m	0.0	x	x
Dvt1w (L,W,P)	1/m	5.3e6	x	x
Dvt2w (L,W,P)	1/V	–0.032	x	x
Cgso	F/m	0	x	2.07188e-10
Cgdo	F/m	0	x	2.07188e-10
Cgbo	F/m	0.0	x	x
Xpart		0.0	x	1 Level 53 def=0

Table 8-19. BSIM3_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Drout (L,W,P)		0.56	x	x
Dsub (L,W,P)		Drout	x	x
Ua (L,W,P)	m/V	2.25e-9	x	x
Ua1 (L,W,P)	m/V	4.31e-9	x	x
Ub (L,W,P)	$(m/V)^2$	5.87e-19	x	x
Ub1 (L,W,P)	$(m/V)^2$	-7.61e-18	x	x
Uc (L,W,P)	m/V^2 1/V	-0.0456	x	-4.56e-11
Uc1 (L,W,P)	m/V^2 1/V	-0.056	x	-5.69e-11
U0 (L,W,P)	$cm^{2/Vs}$	670.0 NMOS 250.0 PMOS	x	x
Ute (L,W,P)		–1.5	x	x
Rdsw (L,W,P)	$ohms \times \mu m^{Wr}$	0.0	x	x
Prwg (L,W,P)	1/V	0.0	x	x
Prwb (L,W,P)	1/V	0.0	x	x
Wr (L,W,P)		1.0	-	x
Prt (L,W,P)	$ohms \times \mu m$	0.0	x	x
Vsat (L,W,P)	m/s	8.0e4	x	x
At (L,W,P)	m/s	3.3e4	x	x
A0 (L,W,P)		1.0	x	x
Keta (L,W,P)	1/V	–0.047	x	x
Ags (L,W,P)	1/V	0.0	x	x
A1 (L,W,P)	1/V	0.0	x	x
A2 (L,W,P)		1.0	x	x
B0 (L,W,P)	m	0.0	x	x
B1 (L,W,P)	m	0.0	x	x
Voff (L,W,P)	V	-0.08	x	x
Voffcv (L,W,P)		0.0	-	x
Noff (L,W,P)		1.0	-	x

Table 8-19. BSIM3_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Nfactor (L,W,P)		1.0	x	x
Ijth	A	.1	-	x
Cdsc (L,W,P)	F/m^2	2.4e-4	x	x
Cdscb (L,W,P)	F/V/m^2	0.0	x	x
Cdscd (L,W,P)	F/V/m^2	0.0	x	x
Cit (L,W,P)	F/m^2	0.0	x	x
Eta0 (L,W,P)		0.08	x	x
Etab (L,W,P)	1/V	–0.07	x	x
Pclm (L,W,P)		1.3	-	x
Pdiblc1 (L,W,P)		0.39	x	x
Pdiblc2 (L,W,P)		0.0086	x	x
Pdiblcb (L,W,P)	1/V	0	x	x
Pscbe1 (L,W,P)	V/m	4.24e8	x	x
Pscbe2 (L,W,P)	V/m	1e-5	x	x
Pvag (L,W,P)		0.0	x	x
Delta (L,W,P)	V	0	x	0.01
Kt1 (L,W,P)	V	–0.11	x	0.0
Kt1l (L,W,P)	m-V	0.0	x	x
Kt2 (L,W,P)		0.022	x	x
Cgsl (L,W,P)	F/m	0.0	x	x
Cgdl (L,W,P)	F/m	0.0	-	x
Ckappa (L,W,P)	F/m	0.6	x	x
Cf (L,W,P)	F/m	calculated	x	x
Clc (L,W,P)	m	0.1e-6	x	x
Cle (L,W,P)		0.6	x	x
Dlc	m	Lint	x	x
Dwc	m	Wint	x	x
Kf		0.0	x	x
Af		1.0	x	x

Table 8-19. BSIM3_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Ef		1.0	x	x
Em	V/m	4.1e-7	x	4.1e+7
Noia		1.0e20 NMOS 9.9e18 PMOS	x	x
Noib		5.0e4 NMOS 2.4e3 PMOS	x	x
Noic		-1.4e-12 NMOS 1.4e-12 PMOS	x	x
Acm		10	-	x
Calcacm		0	-	x
Hdif	m	0	-	x
Ldif	m	0	-	x
Wmlt		1	-	x
Xw	m	0	-	x
Xl	m	0	-	x
Rdc	Ohms	0	-	x
Rsc	Ohms	0	-	x
Cjswg	F/m	Cjsw	-	CJGATE
Pbswg	V	Mjsw	-	PHP
Mjswg		1/3	-	.33
Is	A	1e-14	-	x
Rd	Ohms	0	-	x
Rs	Ohms	0	-	x
Tlev		0	-	x
Tlevc		0	-	x
Eg	eV	1.16	-	x
Gap1	V/°C	7.04e-4	-	x
Gap2	K	1108	-	x
Cta	1/°C	0	-	x
Ctp	1/°C	0	-	x

Table 8-19. BSIM3_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Pta	1/°C	0	-	x
Ptp	1/°C	0	-	x
Trd	1/°C	0	-	x
Trs	1/°C	0	-	x
Nqsmod		0	-	x
Nlev		-1	-	x
Gdsnoi			-	x
Vfbflag		0	-	x

Comments:

The binning parameters Wmin, Wmax, Lmin and Lmax have been added to the model. They are not used by the simulator; they are supported solely for making models self-documenting. To perform binning, the ADS BinModel must be used. The translator automatically sets up the BinModel and moves these parameters to that BinModel. Therefore, the translator does not set these parameters on the BSIM3 model, just the BinModel.

Hspice only: If Level=49 and Capmod=0, the translator sets B3qmod=1 to turn on a special Hspice compatibility mode. Otherwise, the default value of B3qmod=0.

BSIM4_Model:BSIM4 MOSFET Model

The SPICE BSIM3 MOSFET model is translated to the ADS MOSFET BSIM3_Model. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model nch nmos versiion=4.0.0 ...
```

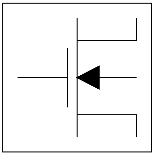
SPICE Dialect and Netlist Syntax:

Spice2/3:	Does not exist
PSpice:	Does not exist
HSpice:	**.model** *mname* **NMOS | PMOS VERSION=4.0.0** [param=*value*]*

ADS Netlist Syntax:

model *mname* **BSIM4 NMOS=[0|1] PMOS=[0|1] Idsmod=8** [param=*value*]*

ADS Schematic Symbol:



BSIM4_Model
NMOS=yes
PMOS=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-20. BSIM4_Model Parameters

ADS	Units	Default	PSpice	HSpice
Capmod		2	x	2
Diomod		1	x	1
Rdsmod		0	x2	0
Trnqsmod		0	x	0
Acnqsmod		0	x	0
Mobmod		1	x	1
Rbodymod		0	x	0
Rgatemod		0	x	0
Geomod		0	x	0
Fnoimod		1	x	1
Tnoimod		0	x	0
Igbmod		0	x	0
Parachk		1	x	1
Binunit		1	x	1
Version		4.0.0	x	4.0.0
Toxe	m	3.0e-9	x	3.0e-9
Toxp		TOXE	x	TOXE
Toxm		TOXE	x	TOXE
Toxref	m	3.0e-9	x	3.0e-9
Dtox	m	0.0	x	0.0
Epsrox	Si02	3.9	x	3.9
Cdsc	F/(V*m^2)	2.4e-4	x	2.4e-4
Cdscb	F/(V*m^2)	0.0	x	x
Cdscd	F/(V*m^2)	1.0	x	x
Cit	F/(V*m^2)	0.0	x	x
Nfactor		1	x	1
Xj	m	1.5e-7	x	1.5e-7
Vsat	m/s	8.0e-4	x	8.0e-4
At	m/s	3.3e-4	x	3.3e-4
A0		1	x	1

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Ags	V^-1	0.0	x	0.0
A1	V^-1	0.0	x	0.0
A2		1	x	1
Keta	V^-1	-0.047	x	-0.047
Nsub	cm^-3	6.0e-16	x	6.0e-16
Ndep	cm^-3	1.7e-17	x	1.7e-17
Nsd	cm^-3	1.0e-20	x	1.0e-20
Phin	V	0.0	x	0.0
Ngate	cm^-3	1	x	1
Gamma1	V^(1/2)	calculated	x	calculated
Gamma2	V^(1/2)	calculated	x	calculated
Vbx	V	calculated	x	calculated
Vbm	V	-3.0	x	-3.0
Xt	m	1.55e-7	x	1.55e-7
K1	V^(1/2)	0.5	x	0.5
Kt1	$V^{(1/2)}$	-0.11	x	-0.11
Kt1l	V*m	0.0	x	0.0
Kt2		0.022	x	0.022
K2		0	x	0
K3		80	x	80
K3b	V^-1	0.0	x	0.0
W0	m	2.5e-6	x	2.5e-6
Dvtp0	m	0.0	x	0.0
Dvpt1	V^1	0.0	x	0.0
Lpe0	m	1.74e-7	x	1.74e-7
Lpeb	m	0.0	x	0.0
Dvt0		2.2	x	2.2
Dvt1		0.53	x	0.53
Dvt2	V^-1	-0.032	x	-0.032
Dvt0w		0	x	0

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Dvt1w	m^-1	5.3e-6	x	5.3e-6
Dvt2w	V^-1	-0.032	x	-0.032
Drout		0.56	x	0.56
Dsub		Drout	x	Drout
Vth0	V (NMOS)	0.7	x	0.7
Vtho	V (PMOS)	-0.7	x	-0.7
Ua	m/V	1.0e-9 for MOBMOD=0 and 1; 1.0e-15 for MOBMOD=2	x	1.0e-9 for MOBMOD=0 and 1; 1.0e-15 for MOBMOD=2
Ua1	m/V	1.0e-9	x	1.0e-9
Ub	(m/V)^2	1.0e-19	x	1.0e-19
Ub1	(m/V)^2	1.0e-18	x	1.0e-18
Uc	V^-1and m/V^2	-0.0465 V^-1for MOBMOD=1; -0.0465e-9 m/V^2 for MOBMOD=0 and 2	x	-0.0465 V^-1for MOBMOD=1; -0.0465e-9 m/V^2 for MOBMOD=0 and 2
Uc1	V^-1and m/V^2	0.067 V^-1for MOBMOD=1; 0.025 m/V^2 for MOBMOD=0 and 2	x	0.067 V^-1for MOBMOD=1; 0.025 m/V^2 for MOBMOD=0 and 2
U0	m^2/(V*s)	0.067 for MOBMOD=1; 0.025 for MOBMOD=0 and 2	x	0.067 for MOBMOD=1; 0.025 for MOBMOD=0 and 2
Eu		1.67 (NMOS); 1.0(PMOS)	x	1.67 (NMOS); 1.0(PMOS)
Ute		-1.5	x	-1.5
Voff	V	-0.08	x	-0.08
Minv		0	x	0
Voffl	V*m	0.0	x	0.0

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Tnom	C	27	x	27
Cgso	F/m	calculated	x	calculated
Cgdo	F/m	calculated	x	calculatedx
Xpart		0	x	0
Delta	V	0.01	x	0.01
Rsh	ohms^2	0.0	x	0.0
Rdsw	$\text{ohms} \times \mu\text{m}^{Wr}$	200.0	x	200.0
Rdswmin	$\text{ohms} \times \mu\text{m}^{Wr}$	0.0	x	0.0
Rsw	$\text{ohms} \times \mu\text{m}^{Wr}$	100.0	x	100.0
Rdw	$\text{ohms} \times \mu\text{m}^{Wr}$	100.0	x	100.0
Rdwmin	$\text{ohms} \times \mu\text{m}^{Wr}$	0.0	x	0.0
Rswmin	$\text{ohms} \times \mu\text{m}^{Wr}$	0.0	x	0.0
Prwg	V^-1	1.0	x	1.0
Prwb	V^(-1/2)	0.0	x	0.0
Prt	$\text{ohms} \times \mu\text{m}^{Wr}$	0.0	x	x
Eta0		0.08	x	0.08
Etab	V^-1	-0.07	x	-0.07
Pclm		1.3	x	1.3
Pdiblc1		0.39	x	0.39
Pdiblc2		0.0086	x	0.0086
Pdiblcb	V^-1	0.0	x	0.0
Fprout	V/m^(1/2)	0.5	x	0.5
Pdits	V^-1	0.0	x	0.0
Pditsl	m^-1	0.0	x	0.0
Pditsd	V^-1	0.0	x	0.0
Pscbe1	V/m	4.24e-8	x	4.24e-8
Pscbe2	m/V	1.0e-5	x	1.0e-5
Pvag		0	x	0

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Jss	A/m^2	1.0e-4	x	1.0e-4
Jsw	A/m	0.0	x	0.0
Jswgs	A/m	0.0	x	0.0
Pbs	V	1.0	x	1.0
Njs		Njs=1.0; Njd=Njs	x	Njs=1.0; Njd=Njs
Xtis		Xtis=3.0; Xtid=Xtis	x	Xtis=3.0; Xtid=Xtis
Mjs		0.5	x	0.5
Pbsws	V	1.0	x	1.0
Mjsws		0.33	x	0.33
Pbswgs	V	Pbswgs=Pbsws	x	Pbswgs= Pbsws
Mjswgs		Mjswgs=Mjsws	x	Mjswgs=Mjsws
Cjs	F/m^2	5.0e-4	x	5.0e-4
Cjsws	F/m	5.0e-10	x	5.0e-10
Cjswgs	F/m	Cjswgs=Cjsws	x	Cjswgs=Cjsws
Jsd	A/m^2	Jsd=Jss	x	Jsd=Jss
Jswd	A/m	Jswd=Jswgs	x	Jswd=Jswgs
Pbd	V	Pbd=Pbs	x	Pbd=Pbs
Njd		1	x	1
Xtid		Xtis=3.0; Xtid=Xtis	x	Xtis=3.0; Xtid=Xtis
Mjd		Mjd=Mjs	x	Mjd=Mjs
Pbswd	V	Pbswd=Pbsws	x	Pbswd=Pbsws
Mjswd		Mjswd=Mjsws	x	Mjswd=Mjsws
Pbswgd	V	Pbswgd=Pbsws	x	Pbswgd= Pbsws
Mjswd		Mjswd=Mjsws	x	Mjswd=Mjsws
Cjd	F/m^2	Cjd=Cjs	x	Cjd=Cjs
Cjswd	F/m	Cjswd=Cjsws	x	Cjswd=Cjsws
Cjswgd	F/m	Cjswgd=Cjsws	x	Cjswgd=Cjsws

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Vfbcv	V	-1.0	x	-1.0
Vfb	V	-1.0	x	-1.0
Tpb	V/K	0.0	x	0.0
Tcj	K^-1	0.0	x	0.0
Tpbsw	V/K	0.0	x	0.0
Tcjsw	K^-1	0.0	x	0.0
Tpbswg	V/K	0.0	x	0.0
Acde	m/V	1.0	x	1.0
Moin		15	x	15
Noff		1	x	1
Voffcv	V	0.0	x	0.0
Dmcg	m	0.0	x	0.0
Dmci	m	Dmcg	x	Dmcg
Dmdg	m	0.0	x	0.0
Dmcgt	m	0.0	x	0.0
Xgw	m	0.0	x	0.0
Xgl	m	0.0	x	0.0
Rshg	Ohms^2	0.1	x	0.1
Ngcon	cm	0.0	x	x
Xrcrg1		12	x	12
Xrcrg2		1	x	1
Lint	m	0.0	x	0.0
Ll	m	0.0	x	0.0
Llc	m		x	
Lln		1	x	1
Lw	m	0.0	x	0.0
Lwc	m	Lw	x	Lw
Lwn		1	x	1
Lwl	m	0.0 mLwn+Lln	x	0.0 mLwn+Lln
Lwlc	m	Lwl	x	Lwl

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Lmin	m	0.0	x	0.0
Lmax	m	0.0	x	0.0
Wr		1	x	1
Wint	m	0.0	x	0.0
Dwg	m/V	0.0	x	0.0
Dwb	m/V^(1/2)	0.0	x	0.0
Wl	m	0.0mWln	x	0.0mWln
Wlc	m	Wl	x	Wl
Wln		1	x	1
Ww	m	0.0mWln	x	0.0mWln
Wwc	m	Ww	x	Ww
Wwn		1	x	1
Wwl	m	0.0mWwn+Wln	x	0.0mWwn+Wln
Wwlc	m	Wwl	x	Wwl
Wmin	m	0.0	x	0.0
Wmax	m	0.0	x	0.0
B0	m	0.0	x	0.0
B1	m	0.0	x	0.0
Cgsl	F/m	0.0	x	0.0
Cgdl	F/m	0.0	x	0.0
Ckappas	V	0.6	x	0.6
Ckapped	V	Ckappas	x	Ckappas
Cf	F/m	calculated	x	calculated
Clc	m	1.0e-7	x	1.0e-7
Cle		0.6	x	0.6
Dwc	m	Wint	x	Wint
Dlc	m	Lint	x	Lint
Dlcig	m	Lint	x	Lint
Dwj		Dwc (in CVmodel)	x	Dwc (in CVmodel)

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Alpha0	A*m/V	0.0	x	0.0
Alpha1	A/V	0.0	x	0.0
Beta0	V	30.0	x	30.0
Agidl	ohms^-1	0.0	x	0.0
Bgidl	V/m	2.3e-9	x	2.3e-9
Cgidl	V^3	0.5	x	0.5
Aigc	(F s 2/g)0.5m-1V-1	0.054 (NMOS) and 0.31 (PMOS)	x	0.054 (NMOS) and 0.31 (PMOS)
Bigc	(F s 2/g)0.5m-1V-1	0.054 (NMOS) and 0.24 (PMOS)	x	0.054 (NMOS) and 0.24 (PMOS)
Cigc	V^-1	0.075(NMOS) and 0.03(PMOS)	x	0.075(NMOS) and 0.03(PMOS)
Aigsd	(F s 2/g)0.5m-1	0.43(NMOS) and (0.31PMOS)	x	0.43(NMOS) and (0.31PMOS)
Bigsd	(F s 2/g)0.5m-1V-1	0.054(NMOS) and 0.024(PMOS)	x	0.054(NMOS) and 0.024(PMOS)
Cigsd	V^-1	0.054(NMOS) and 0.024(PMOS)	x	0.054(NMOS) and 0.024(PMOS)
Aigbacc	(F s 2/g)0.5m-1	0.43	x	0.43
Bigbacc	(F s 2/g)0.5m-1V-1	0.054	x	0.054
Cigbacc	V^-1	0.075	x	0.075
Aigbinv	(F s 2/g)0.5m-1	0.35	x	0.35
Bigbinv	(F s 2/g)0.5m-1V-1	0.03	x	0.03
Cigbinv	V^-1	0.0006	x	0.0006
Nigc		3	x	3
Nigbinv		1	x	1
Nigbacc		1	x	1
Ntox		1	x	1

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Eigbinv	V	1.1	x	1.1
Pigcd		1	x	1
Poxedge		1	x	1
ljthdfwd		ljthdfwd=ljthsfwd	x	jthdfwd= ljthsfwd
ljthsfwd	A	0.1	x	0.1
ljthdrev	A	ljthdrev=ljthsrev	x	ljthdrev= ljthsrev
ljthsrev	A	0.1	x	0.1
Xjbvd		Xjbvd=Xjbvs	x	Xjbvd=Xjbvs
Xjbvs		1.0	x	1.0
Bvd	V	Bvd=Bvs	x	Bvd=Bvs
Bvs	V	10.0	x	10.0
Gmin	ohms^-1	1.0e-12	x	1.0e-12
Rbdb	ohms	50.0	x	50.0
Rbpb	ohms	50.0	x	50.0
Rbsb	ohms	50.0	x	50.0
Rbps	ohms	50.0	x	50.0
Rdpd	ohms	50.0	x	50.0
Noia	(eV)-1S1-EFm-3	6.24e41 for NMOS; 6.188e40 for PMOS	x	6.24e41 for NMOS; 6.188e40 for PMOS
Noib	(eV)-1S1-EFm-3	3.125e26 for NMOS; 1.5e25 for NMOS	x	3.125e26 for NMOS; 1.5e25 for NMOS
Noic	(eV)-1S1_EFm	8.75	x	8.75
Tnoia		1.5	x	1.5
Tnoib		3.5	x	3.5
Ntnoi		1	x	1
Em	V/m	4.1e7	x	4.1e7

Table 8-20. BSIM4_Model Parameters (continued)

ADS	Units	Default	PSpice	HSpice
Ef		1	x	1
Af		1	x	1
Kf	A2-EFs1-EFF	0.0		0.0

MOS_Model9_Process:Philips MOS Model 9 (Process Based)

The SPICE MOS9 MOSFET model is translated to the ADS MOSFET MOS_Model9_Process. For translation information on the MOSFET device, refer to "MOSFET Device" on page 7-18.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model nch nmos level=50 ler=1e-6 wer=10e-6 ...
```

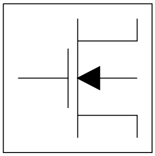
SPICE Dialect and Netlist Syntax:

Spice2/3:	Not available
PSpice:	Not available
HSpice:	**.model** *mname* **NMOS | PMOS LEVEL=50** [param=*value*]*

ADS Netlist Syntax:

model *mname* **MOSFET NMOS=[0|1] PMOS=[0|1] Type=2** [param=*value*]*

ADS Schematic Symbol:



MOS_Model9_Process
NMOS=yes
PMOS=no

Model Parameters:

All parameter names for this model are the same between HSpice and ADS except for the addition of Type=2 to indicate the process level model. However, while the parameter names are the same, most default values are different. Additionally, HSpice supports different default values between the NMOS and PMOS implementations.

The ADS defaults are from an earlier version of the Phillips model. This should not affect the translation because the translator will fill in the proper HSpice defaults.

Consult your HSpice manual for the HSpice defaults if needed. All defaults supplied by the translator are from HSpice V99.2.

GaAsFET and JFET Models for SPICE

The following information describes how the various GaAsFET models from SPICE are translated to the corresponding ADS models.

All GaAsFET devices in SPICE reference a model by its instance name. GaAsFET models in SPICE have a keyword

- **NMF | PMF** for Spice3,
- **GASFET** for PSpice and
- **NJF | PJF** for HSpice.

PSpice and HSpice also use a *Level* parameter to help identify the appropriate model to select. ADS uses the *Idsmod* parameter to identify the model to use. The SPICE *Level* is used by the Netlist Translator to determine what value to set for *Idsmod* and which model to place. Both the Level and Idsmod parameters are listed where available.

Table 8-21 displays Spice3 MESFET device and model information and the corresponding device and model information for schematics and netlists in ADS.

Table 8-21. Spice3 MESFET Mapping Table

Format	Device	Model	ADS Idsmod
Spice3 Netlist	Zxxxxxxx	NMF | PMF	-
ADS Schematic	GaAsFET	“Statz_Model:Statz (Raytheon) GaAsFET Model” on page 8-99	Idsmod = 3
ADS Netlist	-	GaAs	

Table 8-22 displays the initial and translated devices, models and Idsmod values for each of the individual formats listed for the PSpice GaAsFET. PSpice level parameter values are also listed.

Table 8-22. PSpice GaAsFET Mapping Table

PSpice Level	Format	Device	Model	ADS Idsmod
All	PSpice Netlist	Bxxxxxxx	GASFET	-
Level = 1	ADS Schematic	GaAsFET	"Advanced_Curtice2_Model:Advanced Curtice-Quadratic GaAsFET" on page 8-97	Idsmod =1
	ADS Netlist	-	GaAs	
Level = 2	ADS Schematic	GaAsFET	"Statz_Model:Statz (Raytheon) GaAsFET Model" on page 8-99	Idsmod = 3
	ADS Netlist	-	GaAs	
Level = 3	ADS Schematic	TOM	"TOM_Model:Triquint Scalable Nonlinear GaAsFET Model" on page 8-101	Idsmod = 7
	ADS Netlist	-	GaAs	
Level = 4	Not translated.			
Level = 5	Not translated.			

Table 8-23 displays the device, model and Idsmod (where used) for each of the individual formats listed for the HSpice JFET and GaASFET. The HSpice level parameters and SAT parameters are also listed where appropriate.

Table 8-23. HSpice JFET & GaAsFET Mapping Table

HSpice Level	Format	Device	Model	ADS Idsmod
All	HSpice Netlist	Jxxxxxxx	NJF | PJF	-
Level = 1	ADS Schematic	JFET_NFET or JFET_PFET	"JFET_Model:Junction Field Effect Transistor Model" on page 8-51	-
	ADS Netlist	-	JFET	
Level = 2	Not translated.			

Table 8-23. HSpice JFET & GaAsFET Mapping Table

HSpice Level	Format	Device	Model	ADS Idsmod

Level = 3
SAT = 0
	ADS Schematic	GaAsFET	"Advanced_Curtice2_ Model:Advanced Curtice-Quadratic GaAsFET" on page 8-97	Idsmod =1

	ADS Netlist	-	GaAs	

Level = 3
SAT = 2
	ADS Schematic	GaAsFET	"Statz_Model:Statz (Raytheon) GaAsFET Model" on page 8-99	Idsmod = 3

	ADS Netlist	-	GaAs	

Level = 3
SAT = 1,3
	Not translated.			

Note HSpice uses the additional SAT (saturation factor) parameter to determine the appropriate model. The combination of the Level and SAT parameters determine the model used.

Advanced_Curtice2_Model:Advanced Curtice-Quadratic GaAsFET

The PSpice Level 1 GASFET and HSpice Level 3, SAT 0 JFET or MESFET models are translated to the ADS Advanced_Curtice2_Model. For translation information on the GaAsFET device, refer to "Gxxxxxxx" on page 7-43.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
PSpice: .model GNOM GASFET Level=1 [param=value]
HSpice: .model Jmodel NJF level=3 sat=0 [param=value]
```

SPICE Dialect and Netlist Syntax:

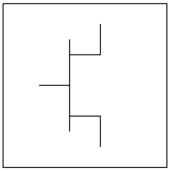
PSpice: **.model** *mname* **GASFET LEVEL=1** [param=*value*]*

HSpice: **.model** *mname* **NJF | PJF LEVEL=3 SAT=0** [param=*value*]*

ADS Netlist Syntax:

model *mname* **Advanced_Curtice2_Model NFET=[0|1] PFET=[0|1]** [param=*value*]*

ADS Schematic Symbol:



Advanced_Curtice2_Model
NFET=yes
PFET=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-24. Advanced_Curtice2_Model Parameters

ADS	Unit	Default	PSpice	HSpice
Idsmod		1	Level=1	Level=3 SAT=0
Vto	V	–2	-2.5	x
Beta	A/V^2	1e-4	0.1	x
Lambda	1/V	0.0	x	x
Alpha	1/V	2.0	x	x
Tau	sec	0.0	x	-
Ucrit		0	-	x
Vgexp		2	-	x
Gamds		-0.01	-	Gamma (def=0)
Vtotc	V/°C	0.0	x	-
Betatce	%/°C	0.0	x	x
Cgs	F	0.0	x	x
Cgd	F	0.0	x	x
Fc		0.5	x	x
Rd	ohms	0.0	x	x
Rg	ohms	0.0	x	x
Rs	ohms	0.0	x	x
Cds	F	0.0	x	-
Vbi	V	0.85	1.0 V	1.0 V
Is	A	1e-14	x	x
Xti		3.0	0	0
Eg	eV	1.11	x	x
N		1.0	x	x

Statz_Model:Statz (Raytheon) GaAsFET Model

The PSpice Level 2 GASFET and HSpice Level 3, SAT2 JFET or MESFET models are translated to the ADS Statz_Model. For translation information on the GaAsFET device, refer to "Zxxxxxxx" on page 7-74. For translation information on the JFET device, refer to "JFET Device" on page 7-22.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

Spice2/3: `.model Zmodel NMF VTO=2.- BETA=-1.0e-3...`

PSpice: `.model GNOM GASFET LEVEL=2 [param=value]...`

HSpice: `.model Jmodel NJF LEVEL=3 SAT=2 CAPOP=1...`

SPICE Dialect and Netlist Syntax:

Spice2/3: **.model** *mname* **NMF | PMF** [param=*value*]*

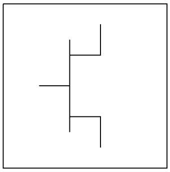
PSpice: **.model** *mname* **GASFET LEVEL=2** [param=*value*]*

HSpice: **.model** *mname* **NJF | PJF LEVEL=3 SAT=2** [param=*value*]*

ADS Netlist Syntax:

model *mname* **GaAs NFET=[0|1] PFET=[0|1] Idsmod=3** [param=*value*]*

ADS Schematic Symbol:



STATZ_Model
NFET=yes
PFET=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-25. Statz_Model Parameters

ADS	Unit	Default	Spice2/3	PSpice	HSpice
Idsmod		3	-	Level=2	Level=3, SAT=2
Vto	V	–2	x	-2.5	x
Beta	A/V^2	1e-4	x	0.1	x
Lambda	1/V	0.0	x	x	x
Alpha	1/V	2.0	x	x	x
B	1/V	0.3	x	x	-
Vbi	V	0.85	-	1.0	1.0
Tau	sec	0.0	-	x	-
Betatce	%/°C	0.0	-	x	x
Cgs	F	0.0	x	x	x
Cgd	F	0.0	x	x	x
Vmax	V	0.5	-	x	-
Fc		0.5	x	x	x
Rd	ohms	0.0	x	x	x
Rg	ohms	0.0	-	x	x
Rs	ohms	0.0	x	x	x
Cds	F	0	-	x	-
Is	A	1e-14	-	x	x
Xti		3.0	-	0.0	0.0
N		1	-	x	x
Eg	eV	1.11	-	x	1.16
Vtotc	V/°C	0.0	-	-	TVC

TOM_Model:Triquint Scalable Nonlinear GaAsFET Model

The PSpice Level 3 GASFET model is translated to the ADS TOM_Model. For translation information on the GaAsFET device, refer to "Bxxxxxxx" on page 7-32.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
    .model Bmodel GASFET level=3
```

SPICE Dialect and Netlist Syntax:

Spice2/3: Does not exist

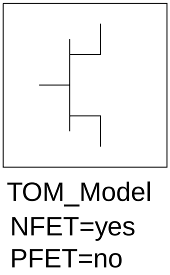
PSpice: **.model** *mname* **GASFET LEVEL=3** [param=*value*]*

HSpice: Does not exist

ADS Netlist Syntax:

model *mname* **TOM_Model NFET=[0|1] PFET=[0|1] Idsmod=7** [param=*value*]*

ADS Schematic Symbol:



Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-26. TOM_Model Parameters

ADS	Unit	Default	PSpice
Idsmod		7	Level=3
Vto	V	–2	-2.5 V
Alpha	1/V	2.0	x
Beta	A/V^2	1e-4	0.1
Q		2.0	x
Tau	sec	0.0	x
Vtotc	V/°C	0.0	x
Betatce	%/°C	0.0	x
Cgs	F	0.0	x
Cgd	F	0.0	x
Vbi	V	0.85	1.0
Vmax	V	-	0.5
Fc		0.5	x
M		0.5	x
Is	A	1e-14	x
N		1.0	x
Eg	eV	1.11	x
Xti		3.0	0.0
Rg	ohms	0.0	x
Rd	ohms	0.0	x
Rs	ohms	0.0	x
Trg1	1/°C	0	x
Trd1	1/°C	0	x
Trs1	1/°C	0	x
Cds	F	0.0	x

JFET_Model:Junction Field Effect Transistor Model

The SPICE J model is translated to the ADS JFET_Model. For translation information on the JFET device, refer to "JFET Device" on page 7-22.

For more information on the ADS model, place the model in a schematic and choose **Edit > Component > Edit Component Parameters** to view the model parameters. You can also click **Help** in the component editor dialog box for additional information.

Example SPICE Command Line:

```
.model Jmodel1 NJF Vto=-2.0
```

SPICE Dialect and Netlist Syntax:

Spice2/3: **.model** *mname* **NJF | PJF** [param=*value*]*

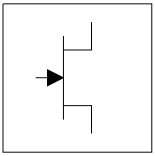
PSpice: **.model** *mname* **NJF | PJF** [param=*value*]*

HSpice: **.model** *mname* **NJF | PJF LEVEL=1** [param=*value*]*

ADS Netlist Syntax:

model *mname* **JFET NFET=[0|1] PFET=[0|1]** [param=*value*]*

ADS Schematic Symbol:



JFET_Model
NFET=yes
PFET=no

Model Parameters:

For information on parameter format, refer to "Using Parameter Mapping Tables to Understand a Translation" on page 8-2.

Table 8-27. JFET_Model Parameters

ADS	Unit	Default	Spice2/3	PSpice	HSpice
					Level=1
Vto	V	–2.0	x	x	x
Beta	A/V^2	1e-4	x	x	x
Lambda	1/V	0.0	x	x	x
Rd	ohms	0.0	x	x	x
Rs	ohms	0.0	x	x	x
Is	A	1e-14	x	x	x
Cgs	F	0.0	x	x	x
Cgd	F	0.0	x	x	x
Pb	V	1.0	x	x	0.8 V
Fc		0.5	x	x	x
Tnom	°C	25	x	T_measured	-
Kf		0.0	x	x	x
Af		1.0	x	x	x
N		1.0	-	x	x
Isr	A	0.0	-	x	-
Nr		2.0	-	x	-
Alpha	1/V	0.0	-	x	-
Vk	V	0.0	-	x	-
M		0.5	-	x	MJ
Vtotc	V/°C	0.0	-	x	TVC
Betatce	%/°C	0.0	-	x	-
Xti		3.0	-	x	0.0

Chapter 9: Adding User Defined Model Translations for Spectre

This chapter provides Spectre model translation mapping information for user defined model translation supported by the Netlist Translator.

Note This document refers to *source dialect* to indicate a generic modelling dialect. The rules and examples relate to any modelling dialect, including Spectre.

The ADS2001 supported Spectre mapping rules reside a Perl file called *spectre.rul*. The file can be edited or copied for your customizing needs. Customizing can consist of modifying the mapping rules for existing model translations or adding new models to the translation portfolio.

OpenTranslator Mapping Rules

Type

This refers to the type of entry in the map file. This is necessary because some spice dialects can have the same *InputName* for components and models. This distinguishes those cases from each other.

The following are possible entries:

- **element**
  Indicates this entry is a component type.
- **model**
  Indicates this entry is a model type.

Input Name

This is the identifier that source dialect uses to indicate the type of component.

Output Name

This is the reference used to indicate the type of ADS component. This is a comma delimited list to support the mapping of many names. This is useful, for example, if you need to map to a different name depending upon if you are generating IFF versus Netlist output.

Pin Mapping

The pin mapping is straight forward. Each pin has a comma delimiter and groups have a semi-colon delimiter. The following is an example:

```
|1,2;2,1|
```

Here source dialect pin #1 maps to ADS pin #2 and source dialect pin #2 maps to ADS pin #1

Parameter Mapping

Parameters are mapped in the order of source dialect parameter, ADS parameter, then source dialect default value. If the source dialect and ADS names are the same, only list the source dialect name. If the source dialect and ADS default values are the same, do not list the source dialect default value. The following is an example:

```
|c, C, 0.0|
```

Here “c” is the source dialect parameter name, “C” is the ADS parameter name (case sensitive), and the default is set to 0.0.

Multiple Parameters Mapping to a Single ADS Parameter

If there are multiple parameters that map to the same ADS parameter name, they will be separated with a tilde. The following is the general rule:

| *X*[~*Y*~*Z*],[*W*],[*value*];... |

The following is an example:

```
|R,,50;W~Width,Width,;...|
```

Here the source dialect parameter "R" maps to ADS parameter "R" and it will use the default value of 50. Note that only list the source dialect name is listed since ADS uses the same name. In the second case, either "W" or "Width" can be the source dialect parameter and they both map to the ADS parameter "Width". Note that the default value is not specified since source dialect and ADS use the same value.

Perl Callbacks

There are two entries for Perl callbacks. The first is the Perl file and the second is the Perl routine to be called.

The order of the callbacks are as follows:

[1] Component name

[2] Pin operations

[3] Parameter name

[4] Model operations

[5] Last Chance

The order of the callbacks is the same as the order of the map rule structure with the addition of the two callbacks, *Model operations* and *Last Chance*.

Component name

The Component name callback will map source dialect component names to the appropriate ADS name. This is necessary to support either Netlist output or IFF output from the translator.

Pin operations

The *Pin operations* callback can handle more sophisticated pin manipulation than just straight pin mappings. For example, some BJT devices require either three or four pins depending upon the configuration.

Parameter name

The *Parameter name* callback will translate the source dialect parameter names to ADS names that are listed in the rules file.

Model operations

The *Model operations* callback handles cases where more model processing is needed that just a straight across mapping.

Last Chance

The *Last Chance* callback is user configurable and is the last called in the sequence of callbacks.

Perl Callback Syntax

The following is the rule syntax for callbacks:

|*filename*, *callback*; [*filename*,]*callback*;...|

Note Any entry that does not have *filename* listed is assumed to be a callback in the last known filename.

The following is an example:

```
|bsim.pl,model_mapping;,param_mapping|
```

Here, at the appropriate time during post processing the Perl code will call *bsim.pl* to source the perl file and then call the routine *model_mapping*.

The next routine that will be called is *param_mapping* and it is assumed to be in the same file.

The following paths will be searched to locate the Perl files:

- .
- $HOME/hpeesof/links/spice/perl/*dialect*
- $HPEESOF_DIR/links/spice/perl/*dialect*
- ../perl/*dialect*

Where *dialect* could be anything you want to call it; *dialect* will be set from the command line argument -pl <*dialect*>.

Note The continuation character \ can be used at any time to keep the line from getting excessively long.

Also, comment lines, comment characters, //, *, #, and in-line comments are not allowed.

Chapter 10: Translating Commands and Functions

This chapter provides information on translating individual commands, functions, and the binning process from Spectre and SPICE to Advanced Design System.

Commands, Functions, and Binning for Spectre

The following table displays a list of commands and functions that are supported by the Netlist Translator.

Table 10-1. Commands, Functions, and Binning

Command / Function	Description
"ends" on page 10-2	End of Subcircuit
"check" on page 10-2	Ignored by ADS
"include" on page 10-2	Include File
"info" on page 10-3	Ignored by ADS
"library" on page 10-3	Library File
"parameters" on page 10-4	Parameter
"subckt, ends" on page 10-5	Subcircuit
"Other Functions" on page 10-6	Function Mapping in the *spectrefunc.rul* File

If the translator reads a command that is not recognized or supported, the translation log file will include a warning such as:

```
WARNING: Skipping unsupported statement.ALTER, line 10.
```

alter and altergroup

The ADS simulator does not have a comparable command. Translation will terminate when this line is found. Everything before it will be translated. Everything after it will be ignored.

check

All check statements will be ignored by the Netlist Translator.

ends

See “subckt, ends” on page 10-5.

include

This line identifies another Spectre file that needs to be read by the translator. The contents are integrated directly into the translated design as appropriate.

Spectre Netlist Syntax:

include *file_name*

ADS Netlist Syntax:

There is no ADS syntax since the referenced file is read and the contents are processed as part of the parent file.

ADS Schematic Symbol:

There is no ADS symbol since the referenced file is read and the contents are processed as part of the parent file.

Parameters:

name	=	Legal file name including file extension.

Comments:

The include statement is used in Spectre to insert the contents of another file. In ADS, separate include statements are not created. Instead, the collection of files are processed as if they were one long file. All circuits defined in an include statement are read by the translator, but a schematic will only be created for those that are referenced in the design hierarchy.

To force creation of all schematics for all subcircuits in an include file, import the include file separately. Note that the top line of an include file may not be a comment line, so to read it as a stand-alone file, insert a comment line or a blank line as line 1, if the existing line 1 has information that needs to be processed. Alternately, you may uncheck the box in the import dialog options page to indicate that the first line is *not* a comment (refer to “Setting the Import Options” on page 2-5), or use the command line argument -l (refer to “Executing the Nettrans Command” on page 3-5) to deselect the *First line is a comment* option.

For tips on debugging the translation of a large set of nested include files, refer to Chapter 6, Troubleshooting.

info

All info statements will be ignored by the Netlist Translator.

library

This line indicates that another Spectre file is to be read by the translator. The contents are integrated directly into the translated design as appropriate.

Spectre Netlist Syntax:

```
library library_name
section section_name
    <statements>
  endsection [section_name]
  section [another_name]
    <statements>
  endsection [another_name]
library library_name
```

ADS Netlist Syntax:

No ADS netlist syntax because the contents are just read and processed like the include file.

ADS Schematic Symbol:

No ADS Schematic Symbol because the contents are just read and processed like the include file.

Parameters:

entryname = Name of library within specified file

filename	=	Legal file name including extension.
dirpath	=	Optional path to library file. There is no space between dirpath and filename when dirpath is given.

Comments:

The library statement is used in Spectre to insert the contents of another file. In ADS, separate include statements are not created. Instead, the collection of files are processed as if they were one long file. All circuits defined in a library statement are read by the translator, but a schematic will only be created for those that are referenced in the design hierarchy.

To force creation of all schematics for all subcircuits in a library file, import the library file separately. Note that the top line of a library file may not be a comment line, so to read it as a stand-alone file, insert a comment line or a blank line as line 1, if the existing line 1 has information that needs to be processed. Alternately, you may uncheck the box in the import dialog options page to indicate that the first line is *not* a comment (refer to "Setting the Import Options" on page 2-5), or use the command line argument -l (refer to "Executing the Nettrans Command" on page 3-5) to deselect the *First line is a comment* option.

For a description of the -*models* command, refer to "Executing the Nettrans Command" on page 3-5. This command can be used to read a model file and create a separate ADS netlist file for each model.

Not supported:

The ADS translator does not automatically read nom.lib and it does not use the search path. All libraries must be specified explicitly with a full path if the library file is not in the working directory. If the file nom.lib is needed, an include statement should be added to the main source file to instruct the translator to read that file.

The .OPTIONS SEARCH statement is not supported by the translator. All library paths must be explicitly specified.

parameters

This line is translated as a name=value parameter.

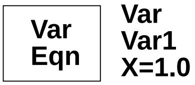
Spectre Syntax:

parameter *name=value*

ADS Netlist Syntax:

name=*value*

ADS Schematic Symbol:



Parameters:

name = Parameter name

value = Parameter value

subckt, ends

This line is translated as a parametric subnetwork.

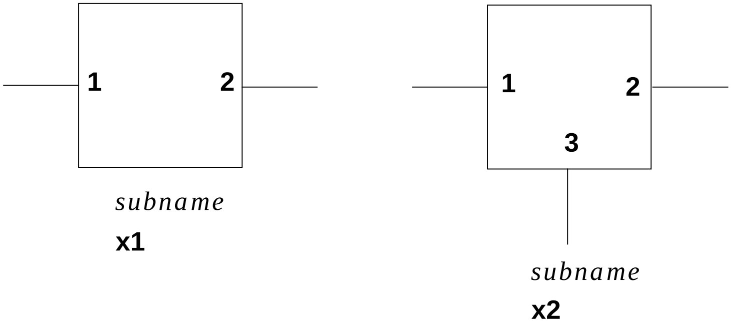
Spectre Netlist Syntax:

subckt *subname* (*node1…nodeN*)
parameters *name*=*value*
ends *subname*

ADS Netlist Syntax:

define *subname* (*n1…*)
parameters *name*=*value*
end *subname*

ADS Schematic Symbol:



Parameters:

subname	=	Subnetwork name
n1*	=	Node names
name	=	Parameter name
value	=	Parameter default value
ends	=	Spectre subcircuit definitions must end with an **ends** statement.

Comments:

For more information on defining subcircuit parameters, refer to "Using the NetlistInclude Component" on page 2-25.

Other Functions

Functions such as log10(x) and ln(x) are mapped in the *spectrefunc.rul* file. Any function that requires mapping into the equivalent ADS function is entered into this file. For more information about function mapping, see Chapter 9, Adding User Defined Model Translations for Spectre.

Commands and Functions for SPICE

The following table displays a list of commands and functions that are supported by the Netlist Translator.

Table 10-2. Commands and Functions

Command / Function	Description
".ALIASES and .ENDALIASES" on page 10-8	Aliases and Endaliases
"alter and altergroup" on page 10-1	Alternate
".END" on page 10-8	End of Circuit
".ENDL" on page 10-8	End of Library
".ENDS or .EOM" on page 10-8	End of Subcircuit
".IC" on page 10-8	Initial Bias Point Condition
".INC[LUDE]" on page 10-9	Include File
".LIB, .ENDLIB" on page 10-11	Library File
".MACRO, .ENDM" on page 10-12	Macro
".MODEL" on page 10-12	Model
".NODESET" on page 10-13	Set the Approximate Node Voltage for the Bias Point
".NOISE" on page 10-14	Noise Analysis
".OP" on page 10-15	Bias Point
"parameters" on page 10-4	Analysis Options
"parameters" on page 10-4	Parameter
"subckt, ends" on page 10-5	Subcircuit
".TEMP" on page 10-20	Temperature
".TRAN" on page 10-21	Transient Analysis

If the translator reads a command that is not recognized or supported, the translation log file will include a warning such as:

```
WARNING: Skipping unsupported statement.ALTER, line 10.
```

.ALIASES and .ENDALIASES

This command block is recognized but not translated. Everything between the .ALIASES and .ENDALIASES lines is ignored.

.ALTER

The ADS simulator does not have a comparable command. Translation will terminate when this line is found. Everything before it will be translated. Everything after it will be ignored.

.DCVOLT

The .DCVOLT statement is processed exactly the same as a .IC statement. For more information, refer to “.IC” on page 10-8.

.END

All processing stops when a .END statement is encountered. Comments and any other information following the .END statement are discarded.

.ENDL

See “.LIB, .ENDLIB” on page 10-11.

.ENDS or .EOM

See “subckt, ends” on page 10-5 or “.MACRO, .ENDM” on page 10-12.

.IC

This line is translated as an initial condition control element. It sets the initial conditions for a transient simulation.

Example SPICE Command Line:

```
      .IC V(11)=5 V(4)=-5 V(2)=2.2
```

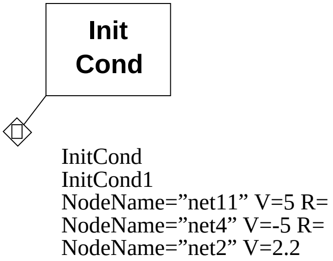
SPICE Netlist Syntax:

.ic v(_node_**)=**_val_ [**v(**_node_**)=**_val_]*]

ADS Netlist Syntax:

InitCond:_instanceName_ **NodeName[**_1_**]=**_node_ **V[**_1_**]=**_val_
NodeName[_2_**]=**_node_ **V[**_2_**]=**_val_

ADS Schematic Symbol:



Parameters:

node = Node number

val = Voltage at the specified node.

instanceName = Unique component name. This is not provided by SPICE so it is automatically generated by the translator.

Comments:

The following PSpice syntax variations are not translated:

.IC V(inPlus,inMinus)=1e-3

.IC I(L1)=2uAmp

.INC[LUDE]

This line identifies another SPICE file that needs to be read by the translator. The contents are integrated directly into the translated design as appropriate.

Example SPICE Command Line:

```
.INC vars.inc
```

SPICE Netlist Syntax:

.INC *name*

ADS Netlist Syntax:

There is no ADS syntax since the referenced file is read and the contents are processed as part of the parent file.

ADS Schematic Symbol:

There is no ADS symbol since the referenced file is read and the contents are processed as part of the parent file.

Parameters:

name	=	Legal file name including file extension.

Comments:

The .INC statement is used in SPICE to insert the contents of another file. In ADS, separate include statements are not created. Instead, the collection of files are processed as if they were one long file. All circuits defined in a .INC statement are read by the translator, but a schematic will only be created for those that are referenced in the design hierarchy.

To force creation of all schematics for all subcircuits in an include file, import the include file separately. Note that the top line of an include file may not be a comment line, so to read it as a stand-alone file, insert a comment line or a blank line as line 1, if the existing line 1 has information that needs to be processed. Alternately, you may uncheck the box in the import dialog options page to indicate that the first line is *not* a comment (refer to “Setting the Import Options” on page 2-5), or use the command line argument -l (refer to “Executing the Nettrans Command” on page 3-5) to deselect the *First line is a comment* option.

For tips on debugging the translation of a large set of nested include files, refer to Chapter 6, Troubleshooting.

.LIB, .ENDLIB

This line indicates that another SPICE file is to be read by the translator. The contents are integrated directly into the translated design as appropriate.

Example SPICE Command Line:

PSpice:	`.LIB linear.lib`
HSpice library call:	`.LIB ‘file1’ mos8`
HSpice library definition:	`.LIB mos8`

SPICE Dialect and Netlist Syntax:

Spice2/3:	Not supported
PSpice:	**.LIB** *name*
HSpice library call:	**.LIB** ‘<*dirpath*>*filename*’ *entryname*
HSpice library definition:	**.LIB** *entryname*

ADS Netlist Syntax:

No ADS netlist syntax because the contents are just read and processed like the include file.

ADS Schematic Symbol:

No ADS Schematic Symbol because the contents are just read and processed like the include file.

Parameters:

entryname = Name of library within specified file

filename = Legal file name including extension.

dirpath = Optional path to library file. There is no space between dirpath and filename when dirpath is given.

Comments:

The .LIB statement is used in SPICE to insert the contents of another file. In ADS, separate include statements are not created. Instead, the collection of files are processed as if they were one long file. All circuits defined in a .LIB statement are read by the translator, but a schematic will only be created for those that are referenced in the design hierarchy.

To force creation of all schematics for all subcircuits in a library file, import the library file separately. Note that the top line of a library file may not be a comment line, so to read it as a stand-alone file, insert a comment line or a blank line as line 1, if the existing line 1 has information that needs to be processed. Alternately, you may uncheck the box in the import dialog options page to indicate that the first line is *not* a comment (refer to "Setting the Import Options" on page 2-5), or use the command line argument -l (refer to "Executing the Nettrans Command" on page 3-5) to deselect the *First line is a comment* option.

For a description of the -*models* command, refer to "Executing the Nettrans Command" on page 3-5. This command can be used to read an HSpice model file and create a separate ADS netlist file for each model.

Not supported:

PSpice has a master library file *nom.lib* that is used by default in PSpice if no library is specified. It also has a LIBPATH variable in a file called *msim.ini*. This search path is used in PSpice if the library is not in the working directory. The ADS translator does not automatically read *nom.lib* and it does not use the search path. All libraries must be specified explicitly with a full path if the library file is not in the working directory. If the file *nom.lib* is needed, an include statement should be added to the main SPICE file to instruct the translator to read that file.

The HSpice .OPTIONS SEARCH statement is not supported by the translator. All library paths must be explicitly specified.

.MACRO, .ENDM

The .MACRO statement is processed exactly the same as a .SUBCKT statement. For more information, refer to "subckt, ends" on page 10-5.

.MODEL

A .MODEL statement is translated to a corresponding model if a compatible model exists in ADS. For translation details of specific models, refer to Chapter 8, Translating a Model.

Example SPICE Command Line:

```
.MODEL dmodel D is=1e-14
```

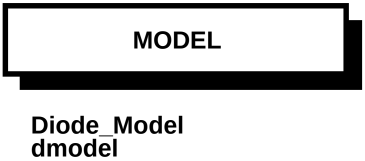
SPICE Netlist Syntax:

.model *name* **type**[*param=value**]

ADS Netlist Syntax:

model *name* **type**[*param=value**]

ADS Schematic Symbol:



.NODESET

This line is translated as a NodeSet control element.

Example SPICE Command Line:

```
.NODESET V(12)=4.5 V(4)=2.23
```

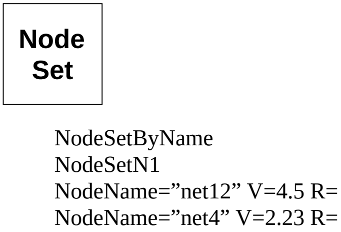
SPICE Netlist Syntax:

.nodeset v(_node_**)=**_val_ [**v(**_node_**)=**_val_]*

ADS Netlist Syntax:

NodeSet:_instanceName_ **NodeName[**_1_**]=**_node_ **V[**_1_**]=**_val_
[**NodeName[**_2_**]=**_node_ **V[**_2_**]=**_val_]

ADS Schematic Symbol:



Parameters:

node	=	Node name
val	=	Voltage at the specified node.
instanceName	=	Unique component name. This is not provided by SPICE so it is automatically generated by the translator.

Comments:

Unsupported syntax:

HSpice .NODESET V(5:SETX)=3.5V

.NOISE

This line is translated as a noise stimulus control element.

Example SPICE Command Line:

```
.NOISE V(5) VIN DEC 10 1KHz 100MHz
```

SPICE Netlist Syntax:

Spice2/3: **.NOISE v(***out***)** <,ref> *srcname* **[dec | oct | lin]** *numpoints fstart fstop* [*pts_out*]

PSpice: **.NOISE v(***out***,**<,ref>) *name=value*

HSpice: **.NOISE v(***out***)** *srcname numpoints*

ADS Netlist Syntax:

**AC:cmp99 SweepVar=”freq” SweepPlan=”freqstim” CalcNoise=yes \
NoiseNode=”***out***”**
SweepPlan:freqstim Start=*fstart* **Stop=***fstop* [**Dec=***ptsperdec* **| Oct=***ptsperoct***/log(2) | Lin=***numpoints*]

ADS Schematic Symbol:

Parameters:

dec	=	Sweep by decades
oct	=	Sweep by octaves
lin	=	Linear sweep

numpoints = Integer number of points in the sweep. If sweeptype is dec or oct, then this is the number of points per decade or octave.

fstart = Start frequency

fstop = Stop frequency

poi = List of points

Comments:

HSpice has extended syntax formats that are not supported by the translator. All extra keywords and values are ignored.

The srcname allows SPICE to compute the output noise referred back to an input node: this usage is ignored by the translator. The optional *pts_out* parameter causes a printed output to occur for one of every *pts_out* frequency points; this is also ignored by the translator.

.OP

This line is translated as a DC analysis control.

Example SPICE Command Line:

```
.OP
```

SPICE Netlist Syntax:

```
.OP
```

ADS Netlist Syntax:

```
DC:cmp1
```

ADS Schematic Symbol:

Parameters:

None. Any optional parameters are ignored.

Comments:

The .OP function in SPICE invokes the simulator to calculate the DC operating point of the circuit. It is sometimes placed in a netlist just to perform the simplest possible analysis to verify that a dc solution can be found. In ADS, this is accomplished with a simple dc analysis with no parameters.

.OPTIONS

The only options currently translated are used to set other values on specific components.

Example SPICE Command Line:

```
.OPTIONS TNOM=25 DCAP=1 WL
```

SPICE Netlist Syntax:

```
.OPTIONS [option name]* [option_name=value]*
```

Parameters:

tnom = Default nominal temperature. Used for models if TNOM is not specified. ADS does not have a global TNOM parameter. If TNOM is specified on the .OPTIONS statement, the value is copied to all models if the nominal temperature is not specified directly on the model. Nominal temperature parameters are as follows:

BSIM1, BSIM2........Temp
MOS_Model9...........Tr
All others.....................Tnom or Tref

dcap = Controls the forward bias capacitance of the diode. ADS does not have an equivalent global parameter, so this is used by the translator to set Fc on the BJT model and Fc and Fcsw on the Diode model. If DCAP is not set to 1, (.OPTIONS DCAP=1), Fc and Fcsw will be set to 0 on the Diode and BJT models.

wl = If values without parameter names are given for Mosfet length and width parameters, the first is assumed to be length. If the wl parameter is present with no value, or the value is set to 1, the order is reversed.

The following parameters are default values for mosfets. If no value is given on a mosfet device for a certain parameter, the SPICE simulator uses the default given here. However, the value of the MOSFET ACM parameter affects the recognition of these values. There is no global ADS equivalent for these parameters and they are not currently translated.

defl	=	Default value for l
defw	=	Default value for w
defad	=	Default value for ad
defas	=	Default value for as
defnrd	=	Default value for nrd
defnrs	=	Default value for nrs
defpd	=	Default value for pd

The following SPICE tolerances have counterparts in ADS on the Options control element but are not set by the translator. They can be set manually after translation.

SPICE	ADS
pivtol	PivAbsThresh
pivrel	PivotRelThresh
vntol	V_AbsTol
abstol	I_AvsTol
reltol	V_RelTol

Comments:

For large RFIC designs, the I_RelTol and V_RelTol should be set to a larger value than the default 1e-6. 1e-3 is a better start value and is the default value for SPICE based simulators. The translator will not set a default value for these.

.PARAM

This line is translated as a name=value parameter.

Example SPICE Command Line:

```
.PARAM name=value
```

SPICE Dialect and Netlist Syntax:

Spice2/3: **.PARAM** *name=value*

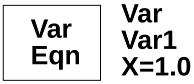
PSpice: **.PARAM** *name=value* (expressions must be in curly brackets)

HSpice: **.PARAM** *name=value* (expressions must be in single quotes)

ADS Netlist Syntax:

name=value

ADS Schematic Symbol:



Parameters:

name = Parameter name

value = Parameter value

.SUBCKT

ENDS

This line is translated as a parametric subnetwork.

Example SPICE Command Line:

```
.SUBCKT OPAMP 1 2 3 4
.ENDS
```

SPICE Dialect and Netlist Syntax:

Spice2/3: **.SUBCKT** *subname* [*n1**]
.ENDS [*subname*]

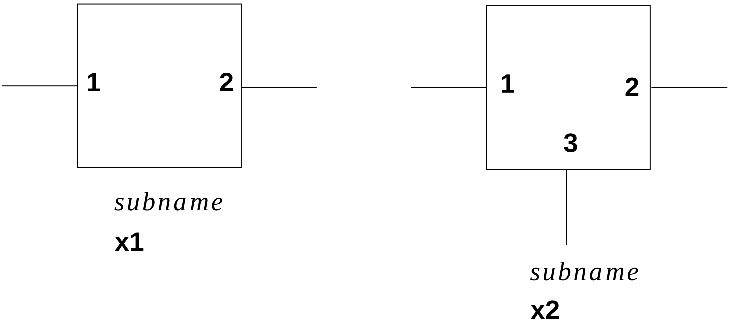
PSpice: **.SUBCKT** *subname* [*n1**] [**OPTIONAL:** *interface node*=*default value*]
[**PARAMS:** *name*=*value**] [**TEXT:** *name*=*textvalue*]
.ENDS

HSpice: **.SUBCKT** *subname* [*n1**] [*name*=*value**]
.ENDS

ADS Netlist Syntax:

define *subname* (*n1*…)
parameters *name*=*value*
end *subname*

ADS Schematic Symbol:



Parameters:

subname	=	Subnetwork name
n1*	=	Node names
name	=	Parameter name
value	=	Parameter default value
.ENDS	=	SPICE subcircuit definitions must end with a .ENDS statement.

Comments:

Spice 3:

Subcircuits may be nested in the SPICE file; however, ADS does not allow this. The translator handles this by writing each subcircuit definition separately in order of occurrence. Line by line comparison with the original netlist will be harder in this case since the lines will not match up directly.

PSpice:

OPTIONAL and TEXT parameters are ignored by the translator.

For more information on defining subcircuit parameters, refer to "Using the NetlistInclude Component" on page 2-25.

.TEMP

Note The translator does not currently translate the TEMP statement. Temp enables you to maintain control over this setting at the highest level of your design.

To set the simulation temperature, place an OPTIONS element in the highest level of your schematic design, or use an Options statement in the netlist as shown below in ADS Netlist Syntax.

Example SPICE Command Line:

```
.TEMP 25
```

SPICE Netlist Syntax:

.TEMP *tempval*

ADS Netlist Syntax:

Options:options1 Temp=*tempval*

ADS Schematic Symbol:

Options
Options1
Temp=25

Parameters:

tempval = Temperature in degrees Celsius.

Comments:

.TRAN

This line is translated as a transient analysis control element.

Example SPICE Command Line:

```
.TRAN 1nS 1000nS 500nS 1nS
```

SPICE Dialect and Netlist Syntax:

Spice2/3: **.TRAN** *tstep tstop* [*tstart*[*tmax*]] [uic]

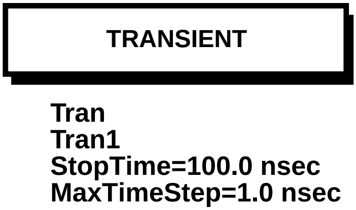
PSpice: **.TRAN[/op]** *tstep tstop* [*tstart*[*tmax*]] [SKIPBP]

HSpice: **.TRAN** *var1* **START=***tstart* **STOP=***tstop* **STEP=***tstep*
OR **.TRAN** *tstep tstop* [*tstep2 tstop2 ...*] [START=*tstart*] [UIC]
OR **.TRAN DATA=***dataname*

ADS Netlist Syntax:

Tran:Tran1 StopTime=*tstop* **MaxTimeStep=***tmax* [**StartTime=***tstart*]

ADS Schematic Symbol:



Parameters:

tstep = Time step between output points.

tstop = Final time.

tstart = Initial time, assumed zero if omitted.

tmax	=	Maximum step size between simulation points.
var1	=	Voltage or current source (ignored).

Comments:

.tran is used to perform a time-domain transient analysis.

Optional UIC keyword is ignored by the translator.

MaxTimeStep is calculated by the translator as the smallest of tstep, (tstop-tstart/50 * tmax)

PSpice optional /OP and SKIPBP are ignored by the translator.

HSpice tstep and tstop are ignored for n>1. SWEEP, SWEEP MONTE, OPTIMIZER and DATA keywords are ignored by the translator.

Index