Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
Allegro Design Flow Integration

Acknowledgments

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

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

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

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

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

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

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

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

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

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

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

About Allegro DFI

The Allegro Design Flow Integration (Allegro DFI) is an add-on to the Cadence Allegro Layout Platform and ADS. This tool is used to export the layout information defined in Allegro into ADS, where you can use the Momentum or FEM simulator to simulate the EM behavior. Using Allegro DFI, you can select specific nets, cross-section layers, and components from a specified area of the layout and then cut them from the design. The cut section of the layout can be saved in the EGS format along with a cross section description of the vertical stack and imported in ADS for EM analysis.

This section introduces the Allegro DFI tool, discusses its benefits, and provides an overview of the process used by Allegro DFI to take selected layout information from a design in Allegro and export it to ADS; where it can be used to run a physical simulation.

Note
It is strongly recommended that you carefully use the items of the Export to ADS menu, which is added to Allegro by the Allegro DFI tool. Go through each of them step by step before you export. You should be careful while selecting the portion of the layout that you need for EM analysis (critical nets and a cookie cut of the adjacent power and ground metal). Brute force, complete layout transfer from Allegro to ADS may cause speed, capacity, and performance issues.

Supported Software Versions

Agilent provides Allegro DFI support for the Cadence Allegro SPB platform with the following software versions:

- Versions 15.7, 16.01, 16.2, and 16.3
- Allegro DFI skill code version 4.1.0
- Tools: PCB Editor, Package Designer (APD), Cadence SiP

Note
Allegro DFI supports Cadence Allegro PCB version 15.7 and higher. Allegro Package Designer and Cadence SiP are supported for versions 16.01 and higher.

The ADS 2011.01 supported version of the Allegro Import Design Kit is version 3.0. Earlier versions will not work with ADS 2011.01 and cannot be converted to work with ADS 2011.01. Version 3.0 of the Allegro Import Design Kit remains compatible with ADS 2009 Update1 but with import support limited to the Allegro Import Design Kit version 2.7 behavior.

Standalone configurations of the Allegro DFI inside the Allegro Platform require access to a Python installation (http://www.python.org).
Supported Python versions are 2.5 and 2.6.

Updates in response to problems caused by Cadence ISR updates are provided under the Technical Support Documents and Examples in the section Allegro Design Flow Integration

Supported Operating Systems

See ADS installation guide for supported operating systems for the ADS components of the Allegro DFI.

For information about the supported operating systems for Allegro, refer to Cadence Allegro documentation or the Cadence website at:

http://www.cadence.com

Accessing Allegro DFI Documentation

You cannot access the Allegro DFI documentation from the Allegro PCB editor. To access Allegro DFI documentation you must either:

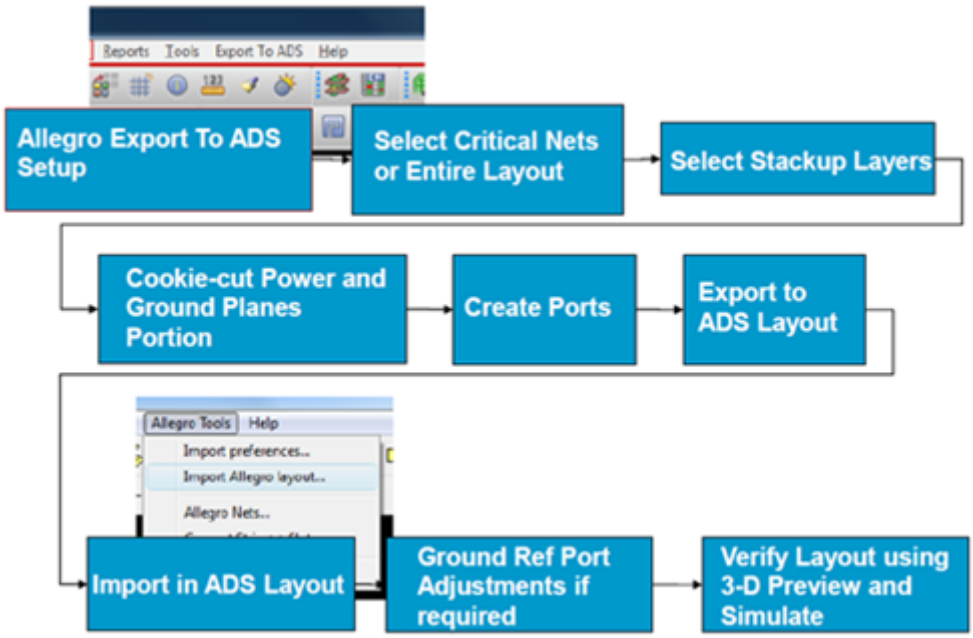
Open ADS, then from any ADS window select **Help** > **Topics and Index** > **I/O** > **Design Translation** > **Allegro Design Flow Integration**.

Or

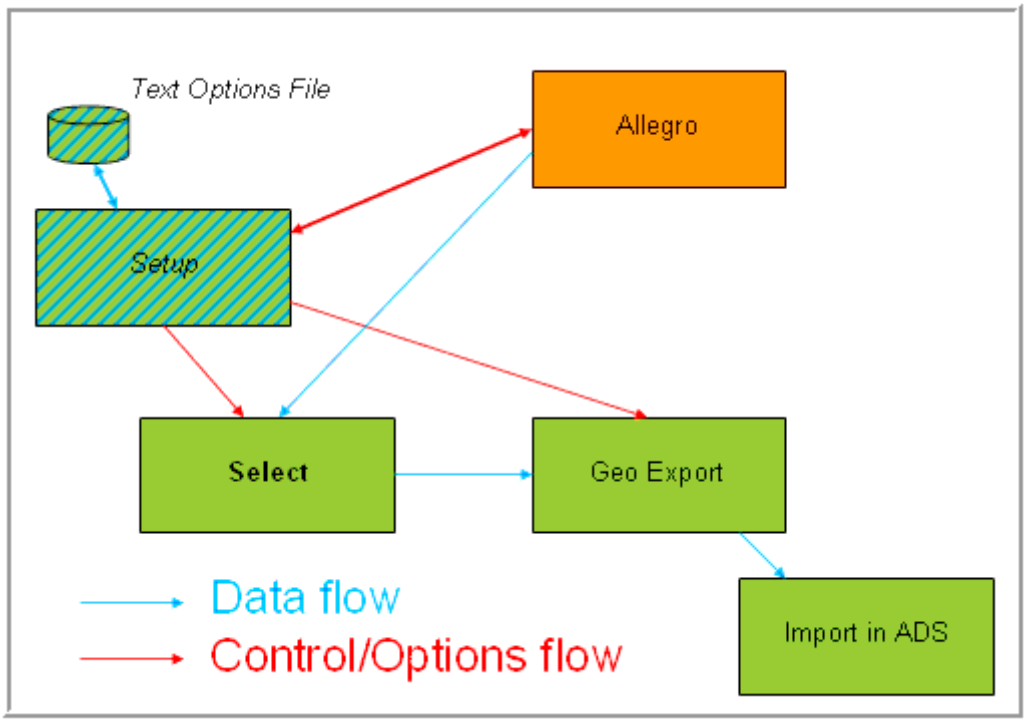
Access the Agilent EEsof EDA website:
http://eesof.tm.agilent.com/
Select **Support & Services** > **Product Documentation** > **ADS (Current and Archived doc)** > **ADS 2011.01**.

Allegro DFI Use Model Overview

The following diagram depicts the steps for transferring design information from Allegro into ADS:



The internal process flow of the Allegro DFI tool is illustrated in the following figure:



What's New in ADS 2011.01

ADS 2011.01 supports the following new features in the Allegro DFI:

- Supports Cadence SPB 16.01+
- Enables you to select unnamed electrical objects as RF Ground.
- Supports split ground planes.
- Supports component based selection of pins for port generation.
- Exports selected components in hierarchical design.
- Exports die stacks with bond wires in cdnsipand apd.
- Exports cookie cutter polygon and import of cutter polygon.
- Layers are ordered according to the cross section including drill holes at start of layer list.
- Layers set to 50% transparency.
- Consistent processing of holes in positive and negative shapes.
- Conversion between strips and slots in ADS.

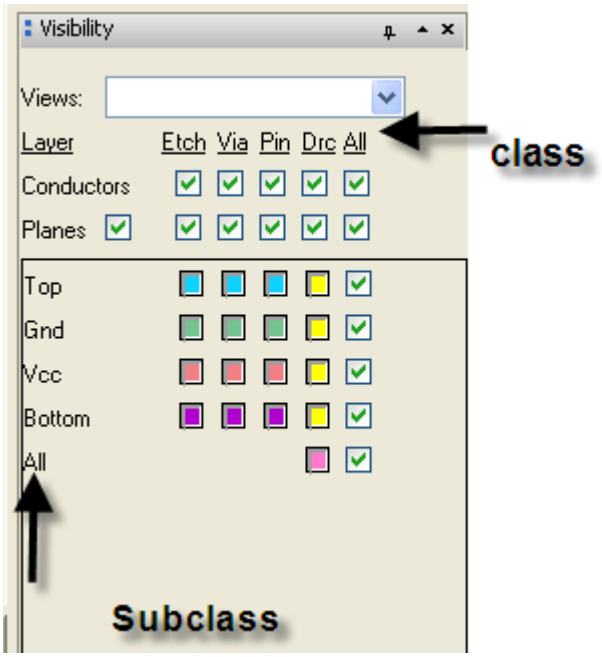
Allegro Platform Overview

Before using Allegro DFI for transferring design data to ADS, you need to setup a board, package, or module design and create at least a partial layout design in the Allegro Layout platform. The details of designing with the Allegro platform are outside the scope of this documentation, but all users of the Allegro DFI functionality need some basic knowledge about using the Allegro Layout Platform to complete transfer tasks.

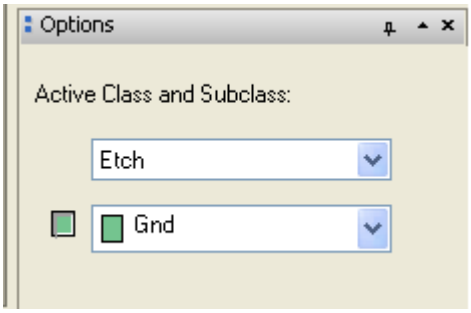
This section provides information about basic Allegro features, such as design parameters, cross-sections, classes, options, and edit functions that you need to use inside Allegro during a transfer of your Allegro Layout to ADS with the Allegro DFI.

Viewing Objects

In Allegro, database objects are usually organized in a class or subclass hierarchy. You can select the required view for showing classes and subclasses from the **Views** drop-down list. You can also deactivate the class and subclass hierarchy in Allegro. Classes and subclasses used in a cross-section are displayed, as shown in the following figure:



You can also view information about the current active operation in Allegro, as shown in the following figure:

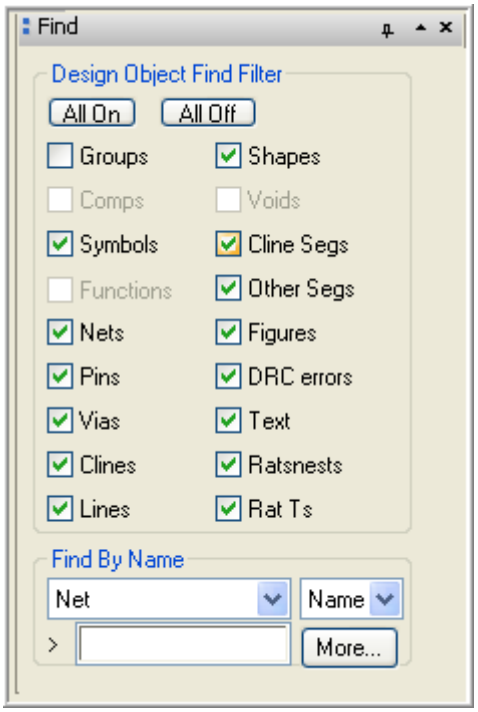


Selecting Objects

For selecting objects, Allegro uses the following operational model:

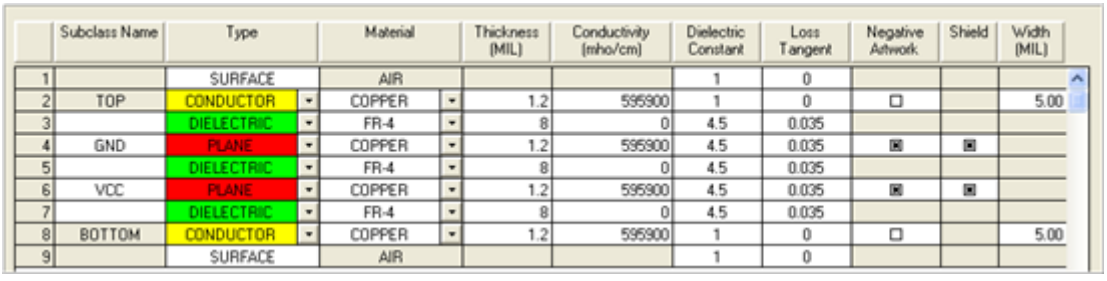
1. Choose an operation.
2. Select an object.
3. Apply the operation, choose the next object, or reject the operation.
4. Click **Done**.

The application mode may change depending on the command context. You can use the general edit options by right-clicking outside the design. To apply filters, use the **Find** dialog box. While choosing an action, the filter settings used in the last operation are automatically loaded.



Verifying Cross-section Settings

To specify the cross-section settings, select **Open Setup** > **Cross Section**.



	Subclass Name	Type	Material	Thickness (MIL)	Conductivity (mho/cm)	Dielectric Constant	Loss Tangent	Negative Artwork	Shield	Width (MIL)
1		SURFACE	AIR			1	0			
2	TOP	CONDUCTOR	COPPER	1.2	595900	1	0	☐		5.00
3		DIELECTRIC	FR-4	8	0	4.5	0.035			
4	GND	PLANE	COPPER	1.2	595900	4.5	0.035	☒	☒	
5		DIELECTRIC	FR-4	8	0	4.5	0.035			
6	VCC	PLANE	COPPER	1.2	595900	4.5	0.035	☒	☒	
7		DIELECTRIC	FR-4	8	0	4.5	0.035			
8	BOTTOM	CONDUCTOR	COPPER	1.2	595900	1	0	☐		5.00
9		SURFACE	AIR			1	0			

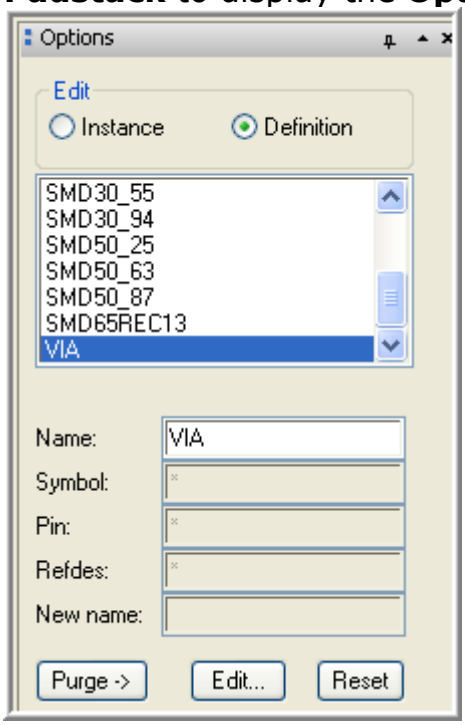
In the cross section:

- Negative Artwork is equivalent to Momentum slot mask, but the display shown is

always positive.
- Shield indicates that this is a plane, typically for pwr/gnd distribution. It informs the router to stay away from it and always exported as sheet conductor by Allegro DFI if strip export is chosen.
- TOP to BOTTOM defines a physical stack.
- Conducting layers (CONDUCTOR/PLANE) must be separated by dielectric. However, Allegro does not always enforce this rule. If this is not the case, always check the exported substrate stack in ADS because the automatic translation might be inaccurate.
- In traditional Allegro PCB design, only conducting layers get a subclass names. In APD/SiP, dielectric layer can get a name used to define, such as blind or buried vias. The material properties of such layers must be check carefully. The Allegro DFI export does not know the via material in this situation. Always verify the exported cross-section in this case.

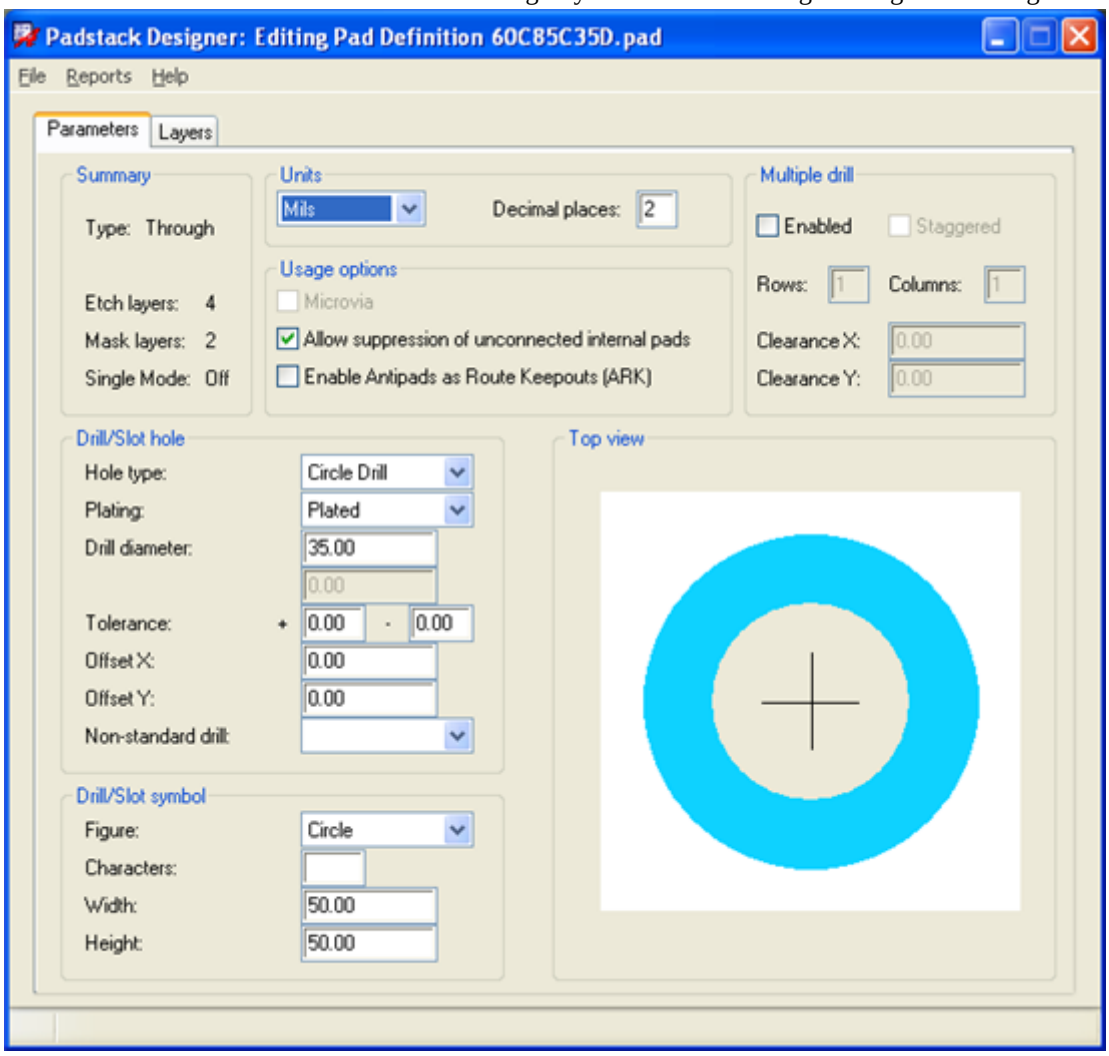
Defining Padstacks

To set your preferences for padstacks, choose **Tools > Padstack > Modify Design Padstack** to display the **Options** dialog box, as shown below:



Using the Options dialog box:

- You can choose any component in the design, for example a Via in the design.
- You can edit the properties associated with it by clicking **Edit** or right-click **Edit** to open the **Editing Padstack Definition**, as shown below:

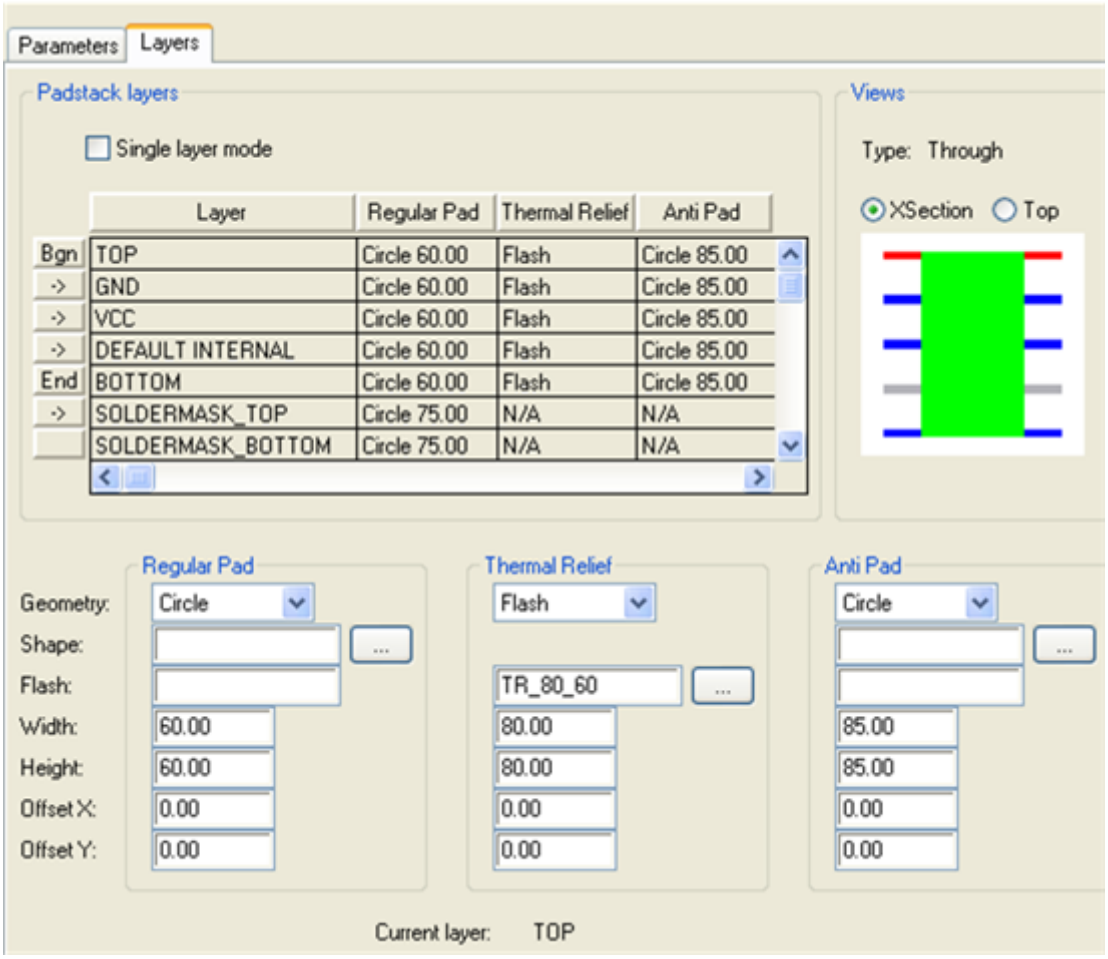


The **Padstack Designer** dialog box consists of **Parameters** and **Layers** tab. All the information related to component (Via in this case) is displayed in this dialog box. It allows you to suppress unconnected catch pads and plating information (electrical properties), which are specified in the *eemom.option* file.

Note
Multi-drill with staggered holes is not supported in Allegro DFI.

For setting layers, depending on the nets of the shapes that the via crosses, you can use a Regular pad, Thermal relief pad, or Anti pad.

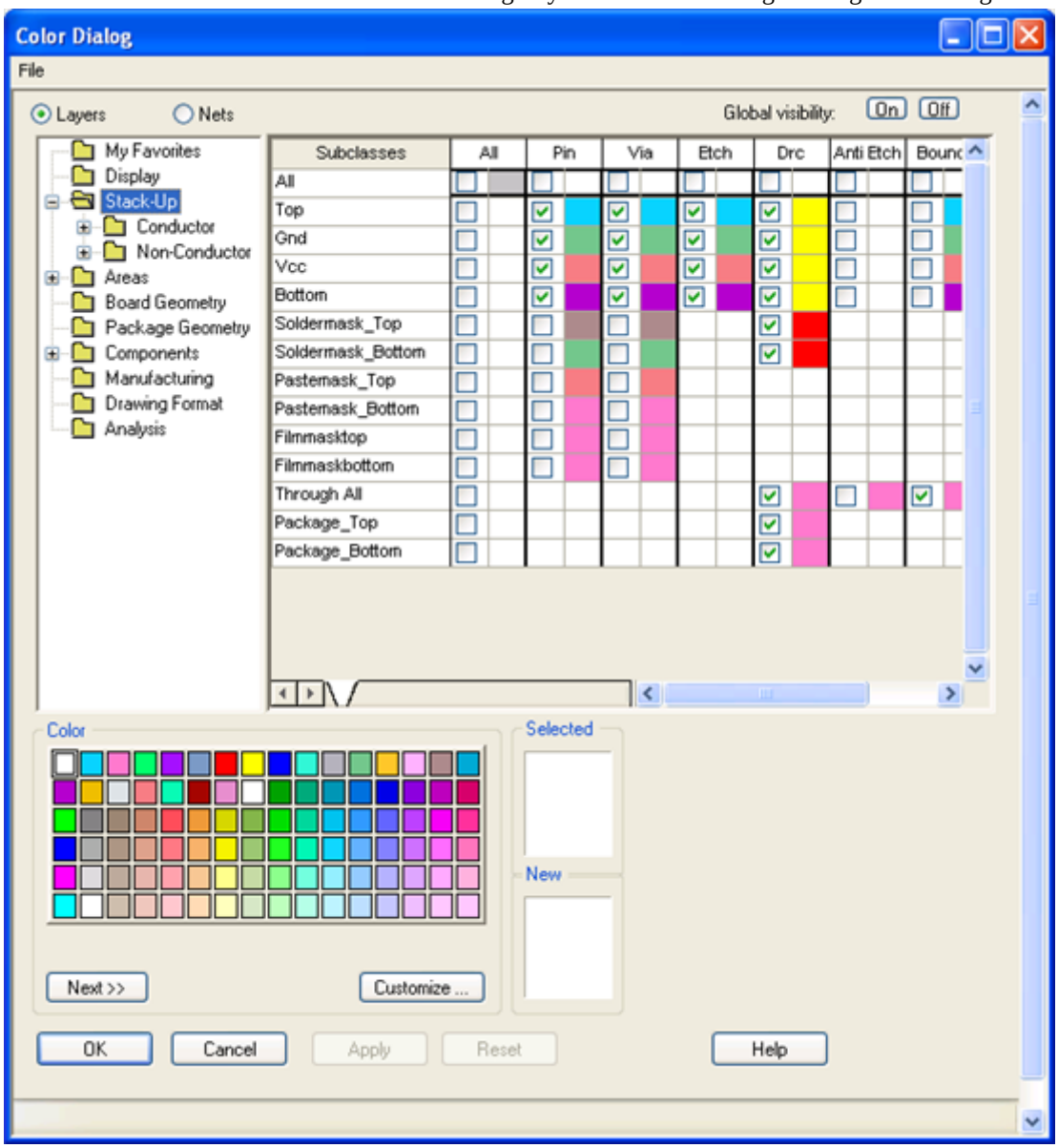
- Thermal relief is added when a via connects to a plane shape.
  - If a plane is all **positive**, relief is created by dynamic routing mechanism.
  - If a plane is **negative**, relief shape is inserted in the plane.



- Be very cautious with this when you switch between negative/positive in cross section. For example, Thermal relief as circle is working fine on the positive plane, but may disconnect the via when switching to the negative plane.
- The Allegro DFI option allows you to ignore thermal relief on negative layers.

Display Parameters

Select **Open Display** > **Color/Visibility** to change the display related parameters. The **Color Dialog** box is displayed, as shown in the following figure:



In **Color Dialog**:

- Allegro DFI uses a scratch layer for cookie cutter, etc. Usually it is called *MANUFACTURING/EEM_SCRATCH*. If you get an error message related to the scratch layer, toggle visibility of the Manufacturing layers and click **Apply**. This scratch layer is the only object that is not removed at the end of a Allegro DFI session.

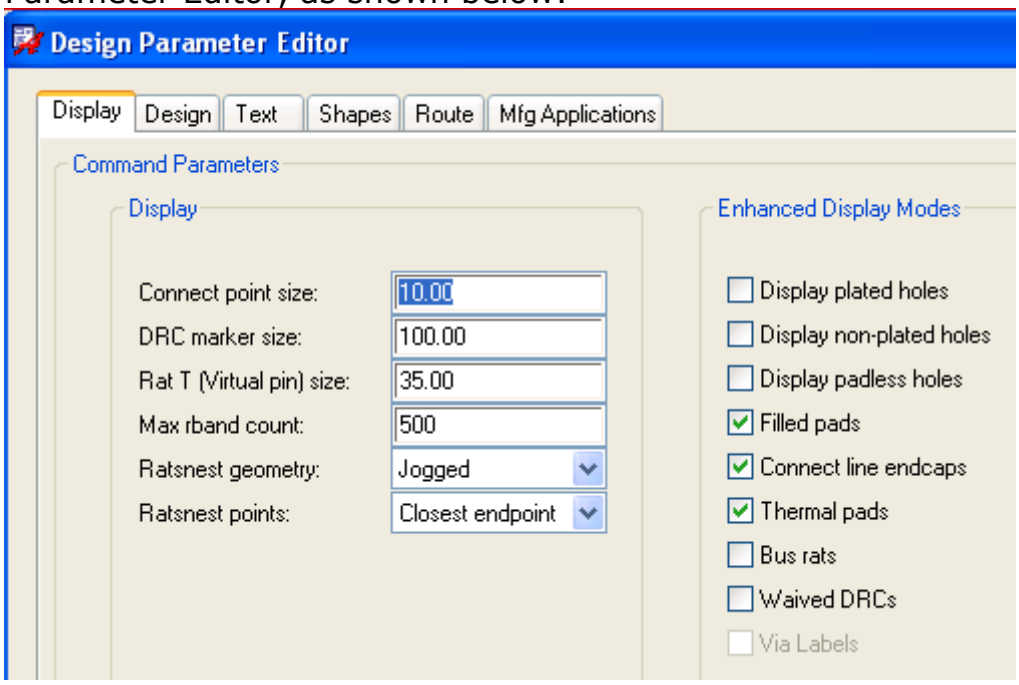
- You can also adjust display bundles, shadow mode, and shapes transparency.

On some graphics cards, OpenGL fails. You need to change the settings by performing the following steps:

1. Choose **Setup** > **User Preferences**.
2. Choose **Display** > **Opengl**.
3. Disable the **opengl** flag and restart Allegro.

Design Parameters

To set design parameters, choose **Setup** > **Design Parameters**. This displays the Design Parameter Editor, as shown below:

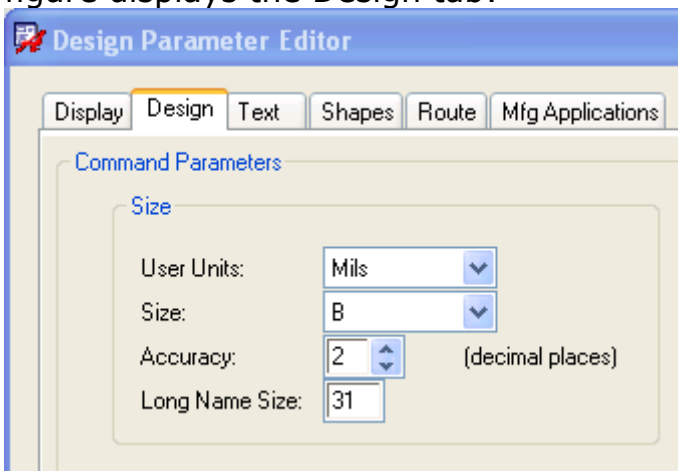


Some of the Design Parameter Editor tabs are described below:

- **Display**: This tab consists of **Display** and **Enhanced Display Modes** sections. It controls how padstacks are displayed. You can set view preferences by selecting the require options. Click **Apply** to implement each option.

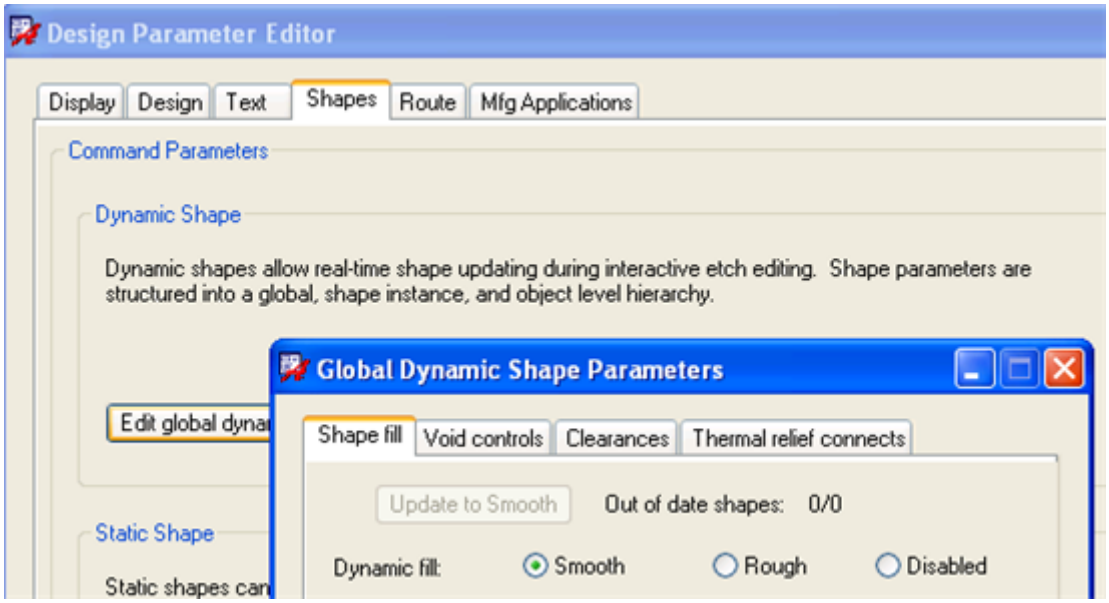
  Note
  You should note the difference on a board with all positive layers versus negative layers.

- **Design**: This tab controls the accuracy settings, which corresponds with the ADS layout resolution. The accuracy settings also control substrate export. The following figure displays the Design tab:



- **Shapes**: You can select the Dynamic and Static Shape options to edit your design in real time or in non-real time.
  - Click **Edit global dynamic shape** to check the number of *out of date shapes*.

This must be 0.



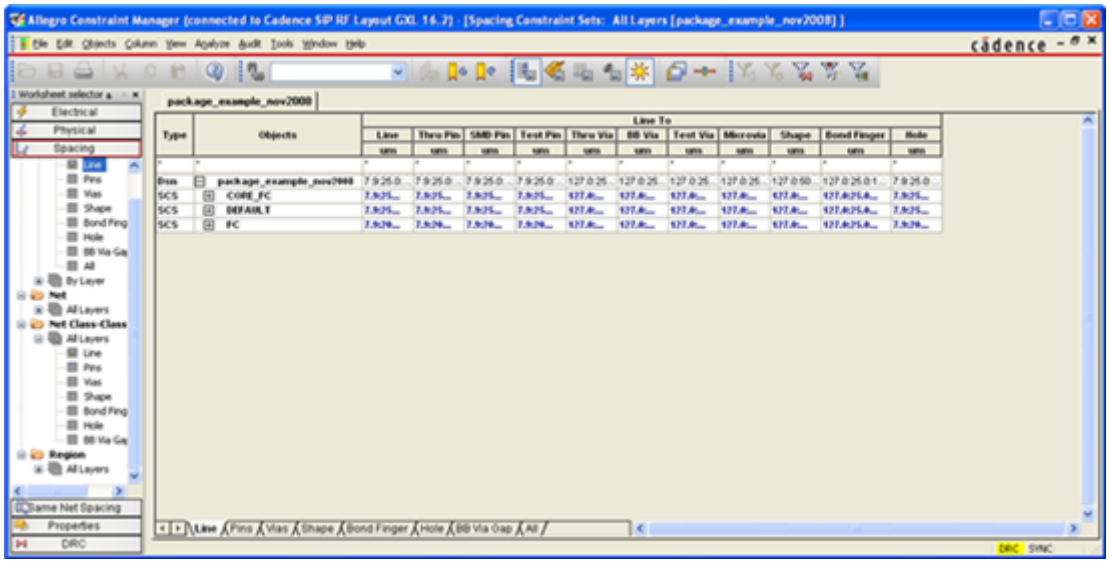
An example of dynamic shape processing is the creation of thermal reliefs and anti-pads on postive shapes.
Operations on big or incomplete designs may disable the dynamic shape update. This can causes incorrect shapes/connectivity and requires manual update before starting an export.

Constraint Manager (version 16.x)

You can define constraints for the following options in a design:

- Spacing, wiring, and delay rules.
- Pad and padstack definitions for blind and/or buried vias and bondfingers.

You do not need an Allegro DFI export for controlling these options. For example, delay constraints for a line can be found in this tool. A tool that checks electrical design rule violations is shown below:



Exporting Design Data From Allegro Using Allegro DFI

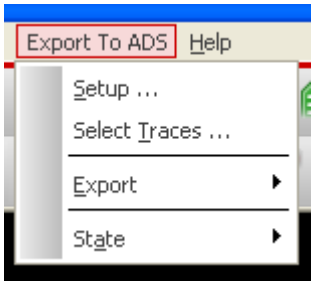
Allegro DFI transfers data targeted at performing an EM simulation. It does not transfer an entire design database. To transfer the entire database you should use other standard mechanisms, like Gerber, stream, IFF, and ODB++ export.

An Allegro DFI export process consists of the following steps:

1. Setup
2. Selection
3. Export

The Allegro DFI export setup and selection is tailored towards gathering the relevant information for setting up and running an EM simulation from ADS. You should be careful while selecting data in Allegro DFI and adjust the export, if necessary. It can save you significant time later when performing the EM simulations in ADS.

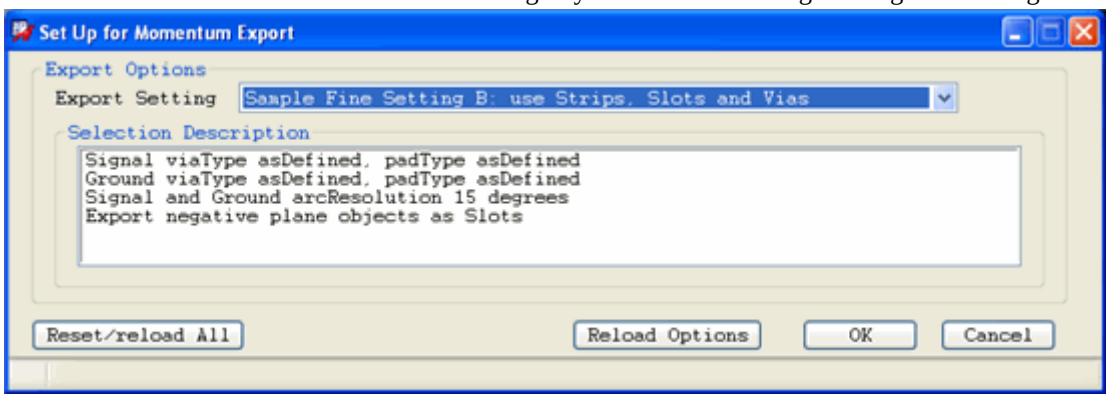
To open Allegro DFI, select **Export to ADS** from the Allegro main menu. This opens the **Export to ADS** drop-down menu, as shown in the following figure:



Selecting **Export > All Artwork** is not the best option as it is an expensive operation, provides no control, and exports flat EGS data and only main substrate. There is little or no other Allegro database information passed, such as no pins, components, or nets.

Set Up and Configuration for Export To ADS

You can configure export by options by selecting **Allegro PCB** > **Export to ADS** > **Setup** to open the **Set Up for Momentum** dialog box. This dialog box is used to select a group of specific configuration options that Allegro DFI needs for exporting files to ADS. You can select an entire group from the drop-down list in this dialog box. The description field is updated according to the chosen selection.



Using the **Set Up for Momentum** dialog box, you can:

- Determine the way layout primitives are processed during export:
  - Behavior with respect to negative objects
  - Vias/Padstacks on Signal Nets, or RF ground nets can be treated independently.
- Define layer mapping information.

- Provide default values for the Allegro DFI selection dialog box, e.g. the default port generation options.

You can also use the Set Up for Momentum dialog box for:

- **Resetting or reloading default configuration options**: You may need to restore the default values after an error from Allegro or Allegro DFI environment results an Allegro command or the Allegro DFI into a faulty state. Click the **Reset/Reload All** button to initialize the default configuration of Allegro DFI data structures in the Allegro environment. For example, if you accidentally try to open Allegro DFI before ending an earlier Allegro edit command, you need to restore the default values.

- **Reloading options**: To update an option file, you need to reload the configuration settings. Click the **Reload Options** button to load new or update option configuration files without reloading Allegro.

Allegro DFI provides some predefined example settings setups to allow a quick start with the export tool. In addition, you can easily define customized setups as per your requirements and automatically add them to the selection list in the drop down list.

Using the Sample Export Settings

The Set Up for Momentum dialog box allows you to choose from predefined example settings. These following sample settings consists of a combination of default values with various levels of accuracy for the export process:

- **Sample Fine Setting A: use Strips and Vias**: This setting consists of the following default values:
  - Signal viaType asDefined, padType asDefined
  - Ground viaType asDefined, padType asDefined
  - Signal and Ground arcResolution 15 degrees

- Converts negative plane objects to strips
- All catch pads are generated

- **Sample Fine Setting B: use Strips, Slots and Vias**: This setting consists of the following default values:
  - Uses Strip/Slot/Via when negative layers present
  - Ground viaType asDefined, padType asDefined
  - Signal and Ground arcResolution 15 degrees
  - Export negative plane objects as Slots
  - All catch pads are generated

- **Sample Board setting using Strips, Slots and Vias**: This setting consists of the the following default values:
  - Uses Strip/Slot/Via when negative layers present
  - Signal viaType asDefined, padType asDefined
  - Ground viaType square, padType asDefined
  - Signal and Ground arcResolution 30.0 degrees
  - Simplify thermal reliefs on negative shapes
  - Remove unconnected catch pads
  - Remove holes in RF Ground shapes not overlapping with RF Signal nets
  - Use the cutter polygon as board boundary

- **Sample Medium Setting**: This setting consists of the following default values:
  - Signal viaType square, padType square
  - Ground viaType square, padType square
  - Signal and Ground arcResolution 30 degrees

- **Sample Course Setting**: This setting consists of the following default values:
  - Signal viaType diamond, padType diamond
  - Ground viaType diamond, padType diamond
  - Signal and Ground arcResolution 45 degrees
  - Remove unconnented catch pads
  - Simplify thermal via connections
  - Remove voids in RF ground shapes that do not overlap signals

Note
Use *Sample Fine Setting B* for a first time transfer. The transferred design will be similar to the original. However, note that the sample settings may not fit your particular analysis needs. Therefore, you should either modify the default settings and/or add additional custom settings to your site or personal configuration files. You can do this by opening, copying, and modifying the _eemom.option_ file(s).

Defining Additional Export to ADS Setups

You can configure the list of settings displayed in the Set Up for Momentum dialog box. This list is created by the concatenation of the contents of *simulation settings* defined in three option files. The title and description fields are selected for display in the setup window and all other values are applied to the internal Allegro DFI configuration fields. The following option files are used:

1. The *$HPEESOF_DIR/ial/config/eemom.option* or the file defined by the environment

variable *EEMOM_GLOBALOPTIONSFILE* if this variable exists. This option file describes all the at the end and it can be replaced by an EDA admin by a site specific default option file by defining the *EEMOM_GLOBALOPTIONSFILE* variable_._

2. *~/pcbenv/eemom.option:* You can copy the default *eemom.option* at *~/pcbenv/eemom.option* and modify it as per your requirements.
3. *< Allegro Design Directory >/< designName > .option*
   - For design specific options, you can also place an option file in the directory of a design, with the base name as the design.
   -
- `.eemom`
  in the directory where

  `.{brd\|mcm\|sip}`

  is placed.

These files are read in the order given above. In case of duplicate settings in the files,(i.e., settings with the same name), the last setting found will be retained for the final list.

Below we show an example fragment from the default *eemom.option* file used to create the *Export Settings* list. Such a file can be copied and modified according to your specifications with a text editor. The latest version of the default file containing a description of all available configuration settings can always be found in the $HPEESOF_DIR/ial/config directory.

Note
The content of the files is actually a independent lisp list, (e.g., it has the format (mnemonic [<name>] <definition>))

- *mnemonic* defines what will be defined
- some mnemonics need a name, others do not
- <definition> is a sequence of values or independent lisp lists

If the lisp mnemonic does not have a name the contents of the sections are concatenated. Every file, of this type, must start with *(setupData* and no blank lines, spaces or comment lines are allowed before these tokens. Aside from this, empty files will also generate errors.

```
(setupData
            ; mnemonic to mark the start of the setup list
  (scratchLayerName "MANUFACTURING/EEM_SCRATCH")
            ; name of a class/subclass in the allegro board that can be
            ; used as a scratch layer for the ADS integration for e.g.
            ; storing the cutter shape ... should not belong to the edge
            ; class
       (egsArcResolution 22.5)
            ; arc faceting angle used when translating Allegro arcs
            ; into the output EGS file
            ; float: (0 < deg < 90) arc resolution used for export
            ;    only for arcs not dealt with in signal or ground
            ;    refacetting step
            ; nil:  use 22.5 degrees
       (egsExportResolution  1000)
            ; int: > 0 preferably a multiple of 10
```

```
      ; if not specified, the allegro native resolution will be used
   (skipGlobalHandling "FALSE")
      ; determines the strategy to use when preparing the selected
      ; objects for export to EGS
      ; skipGlobalHandling does not have any effect for negative
      ; masks when negativeMasksToStrip "TRUE" as this option
      ; requires global handling
      ; flag: true  perform boolean operations locally
      ;       false perform boolean operations globally
...
   (gndOpt
     (viaType "asDefined")
       ; determines how the viashapes will be translated in the output
       ; shape
       ; string enum: "asDefined"
       ;              "square"
       ;              "diamond" (i.e., square rotated by 45 deg)
     (padType "asDefined")
       ; determines how the pin and via pads will be translated in the
       ; output shape
       ; string enum: "asDefined"
       ;              "square"
       ;              "diamond" (i.e., square rotated by 45 deg)
     (arcResolution 15.0)
       ; arc facetting angle used when translating Allegro arcs
       ; into the output EGS file
       ; float: (0 < deg < 90) arc resolution used for export
       ;        (<= 0) do not refacet arc
       ; nil:  use 22.5 degrees
     (minEdgeLength 0.000)
       ; minimum edge length of segments written into the output EGS
       ; file (smaller segments are combined until the threshold is
       ; reached
       ; float: (>= 0) min edge length in the egs file
       ; nil:  use 0.999999 the layout resolution
       ; if arcResolution <= 0, this setting is ignored
     (viaTfAreaRatio 1.0)
       ; multiplication factor for vias when translating the original
       ; shape into the new one. Currently this factor will only
       ; work correctly for circles to square or diamond
       ;    0.0000000000000000   delete shape
       ;    0.7978845608028653   e.g. circle -> inner square
       ;    0.8862269254527580   same perimeter (if circle->square)
       ;    1.0000000000000000   same area
       ;    1.1283791670955125   e.g. circle -> circumvent square
     (padTfAreaRatio 1.0)
       ; same behavior as viaTfAreaRatio but for pads instead of vias
     (simplifyThermal "TRUE")
       ; on negative layers only, for the connections to shape structures
       ; planes
       ;   TRUE   : do not generate thermal relief pad polygons
       ;   FALSE  : generate thermal relief pad polygons
     (sigOpt
       ; same options as in the gnd opt subsection but now for shapes
       ; being defined as signal traces
     (viaType "asDefined")
     (padType "asDefined")
     (arcResolution 15.0)
     (viaTfAreaRatio 1.0)
```

```
          (padTfAreaRatio 1.0)
       )
      )
     ) ; end of setting "Sample Setting A"
  )
)
```

Note
At least one fully specified option file must exist, else Allegro DFI will not work. After making modifications in an eemom.option file, click **Reload Options** to update the settings list.

Using The Option File Sections

Each option file is a lisp “list expression” stored inside a file. It consists of the following features:

- Must start with "(setupData" and end with “)".
- Uses the Skill syntax for a hierarchical list of (name value(s))
- At least one of the option files must start with the general preferences (requires restart of Allegro to update)
  - (scratchLayerName "MANUFACTURING/EEM_SCRATCH"): cutter layer
  - (cutterExpansion 5.0): initial cutter oversize distance
  - (firstEgsLayerNum 1001): first layer number used on ADS side
- The export options have the form (simulationSettings(setting …))

The following options are used in the Option file:

- DrillProps
- Simulation Settings

Configuring DrillProps Section

The DrillProps section allows you to specify the conductivity and other properties of the plating materials for via structures. This is necessary because Allegro is missing this data in the platform UI. In this section:

- Drill -1: Specifies the properties for the DRILL_THROUGH via
- Drill -2: Specifies the properties used by DRILL_<i1i2>, the inter-layer vias and some via structure created in named dielectric layers of a cross-section.

```
(drillProps
  (drillUnits
     (layerThicknessUnits "mil")
     (electricalConductivityUnits "mho/cm")
)
(drill -1
  (layerType DRILL)
  (layerName "default")
  (layerMaterial "COPPER")
  (layerThickness 1.0)
  (layerElectricalConductivity 595900.0)
)
```

```
(drill -2
  (layerType DRILL)
  (layerName "other")
  (layerMaterial "COPPER")
  (layerThickness 1.0)
  (layerElectricalConductivity 595900.0)
)
) ; end of drillProps
```

Configuring Simulation Settings

Each block of simulation settings option contains:

- a title used for the drop down list in the setup dialog
- a multi line description field that provided a textual description of this set of settings
- the actual options section with a global list of settings, followed by gnd and/or signal specific settings.

```
(simulationSettings
            ; start of list of settings available to choose from in the
            ; setup window
    (setting "Sample Setting A"
            ; name that is displayed in setup dialog drop-down combo box
      (description "This Description is viewable in setup dialog"
                   "This will be displayed as the second line"
                   "And this is the third line"
                   "Do not use special escape characters as \n \t")
            ; description displayed in the setup dialog, 1 string/line
      (options ...
```

Frequently used Global Settings

Some frequently enabled or modified global settings in a simulation option section are described below:

- The negativeMasksToStrip setting defines if objects on a negative Allegro layer are exported as strip or slot mask objects for ADS. Convert slot objects into strips objects, if set to "TRUE".

```
(negativeMasksToStrip "FALSE")
            ; determines how to handle negative masks
            ; flag: false - negative masks -> slots
            ;       true - negative masks -> strips
```

- The gndHoleDropDistance and cutterDefinesCrossingGndHoles settings are useful for strip designs with many complex voids. It allows you to drastically simplify plane like structures with many shapes far away from traces of interest before export.

```
(gndHoleDropDistance 0.0)
            ; determines how to handle voids in the RF ground shapes
            ; depending on their distance to Signal net features they can be
            ; skipped from export in the egs file
```

```
            ; float: (>=0.0) voids further away are dropped form the egs export
            ; nil or (< 0.0): keep all holes
(cutterDefinesCrossingGndHoles "TRUE")
            ; determines how holes/voids in RF ground shapes are threated
            ; either as voids with respect to the shape of the board or as
            ; voids
            ; with respect to the cutter shape.
            ; flag: false: only real holes in the original shape are considered
            ;       for hole removal
            ;       true: the cutter shape is used to define potential holes in
            ;       a shape for removal
            ; Note: when true the cutter outline is followed outside the board
            ;       outline because this area is essentially also a void region
            ;       with respect to the cutter.
            ;       If board shape must be maintained use either the false
            ;       setting which is less aggressive or modify the cutter
            ;       shape to stay within the board boundary.
```

- The noContourSlotOnNegative settings controls the layer behavior on split ground plane structures on negative planes in Allegro. In this setting:
-
  - A contour slot is created by default on negative layers exported as **SLOT** to separate split ground structures. This introduces an non-existent split/discontinuity in the ground plane structure that can disturb the EM simulations.
  - When this option is set to **TRUE**, it asks the exporter to skip the generation of this contour slot during the export.
  - Avoids editing the slot structure on the ADS side, but shorts split plane structures together on a perfect conducting plane.
  - If the split plane structure must be used, specify the following value: (negativeMasksToStrip TRUE)

```
(noContourSlotOnNegative "FALSE")
   ;determine how to handle shapes on negative masks exported as slot
   ; flag: false - negative masks -> slots with contour slot
   ;       true -negative masks -> slots no contour slot
   ;       negative masks exported as strip donot have contour around the shapes anyhow
```

- You should carefully use the egsMinEdgeLength, autoPortForAllNetPads, and gndHoleDropDistance options.

> **Note**
> Never change the settings for these options, unless given as a workaround by EEsof support.

```
(egsMinEdgeLength 0.000)
            ; minimum edge length of segments written into the output EGS
            ; file (smaller segments are combined until the threshold is
            ; reached
            ; float: (>= 0) min edge length in the egs file
            ; nil:  use 0.999999 times the layout resolution


(autoPortForAllNetPads "FALSE")
    ; determine if the automatic port insertion feature will generate individual
```

```
; ports for RF Signal Pads only or on both RF Signal and RF Ground Pads


(gndHoleDropDistance <positive float>)
            ; determines how to handle voids in the RF ground shapes
            ; depending on their distance to Signal net features they can be
            ; skipped from export in the egs file
            ; float: (>=0.0) voids further away are dropped form the egs export
            ; nil or (< 0.0): keep all holes
```

Frequently used gnd/signal specific settings

The following options are used frequently:

- padType: It is recommended to specify "asDefined", which is handled by Layout simplification.
- viaType: This setting is useful if exact via model is not needed (e.g. for gndvias).
- pathType: It is recommended to specify "rounded", which is handled by Layout simplification.

  Tip
  Use "mitered" when you have problems with missing segments or flipped arcs on traces. (clines)

- If you get **Allow suppression of unconnected catch pads** error due to incorrect pad stack definition, You can specify the following option:

```
(exportPads "connected")
            ; string property describing which "catch" pads
            ; will be generated in the egs output for pins and vias
            ;   "all"       : (default) generate all regular pads in the output
            ;   "none"      : no pads are generate except the ones on outer
            ;                 layers
            ;   "connected" : only if a regular connection other than a via exists
            ;   "startend-connect" : "connected setting" + the top and bottom  pads
        )
```

Selecting Net Based Geometry for Export and Placing EM Ports

Choose **Export to ADS** > **Select Traces** to open the *Select Geometry and Place Ports* dialog box. This dialog box enables you to choose specific nets and add them to *Signal* and *Ground* output lists for export into ADS, by using the *Trace Select* and *Layer Select* tabs.

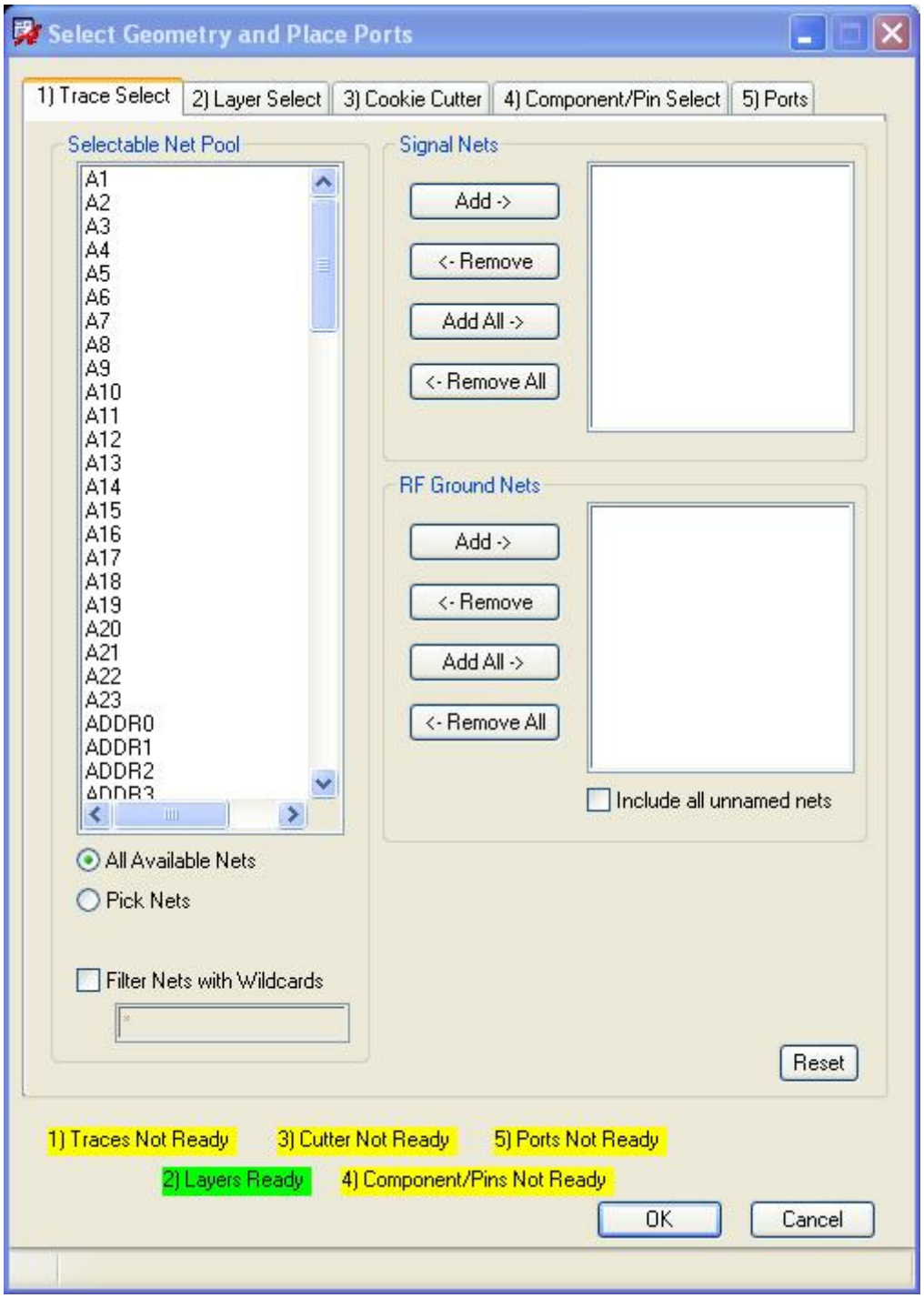
It also enables you to reduce the geometry of the ground nets by using the functions available in the *Cookie Cutter* tab to define a "cutout" shape. Using the cookie cutter, you can reduce the stack of the board, select which layers will be exported and create port definitions for export into ADS.

Note
By default the cookie cutter feature only acts on ground nets and does not alter the shape of the signal nets.

With the *Component/Pin Select* you can choose the component pins that are of interest for

the simulation port generation. The *Port* tab enables you to automatically create ports from the selected trace pads, or clear and edit already existing port definitions.

A visual status check is also provided to alert you that a step is complete and ready for export (green), or that more information is required (yellow).



If you want to experiment now it is time to have a look at an example. Allegro ships with an simple example board cds_routed.brd in the <Allegro install directory>/share/pcb/examples/board_design that you can easily experiment with. Copy this cds_routed.brd file to a local directory and perform the Allegro uprev command if needed. Starting with this board you can follow the operations in the following sections in

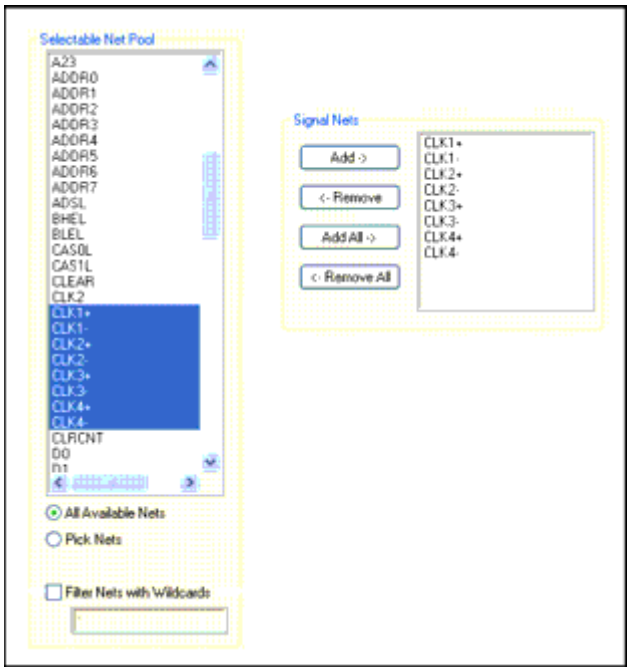
the actual tool.

- Open cds_routed.brd.
- Choose *Sample Fine Setting B* as export setup.
- Experiment with "CLK<i><+|->" signal nets, and "VCC"/"GND" RF ground nets.
- Upcoming sections provide tips when going through the various steps.
- Try to add a large net (PCLK) keeping the cookie cutter.

Using the Trace Select Tab

The **Trace Select** tab enables you to add specific nets to the *Signal Nets* and *RF Ground Nets* output lists.

You can select nets for export can by choosing specific nets from the *Selectable Net Pool* and clicking the **Add** button in the *Signal Nets* section of the dialog, as shown below. The Nets selected in the list will appear as highlighted traces in the design.



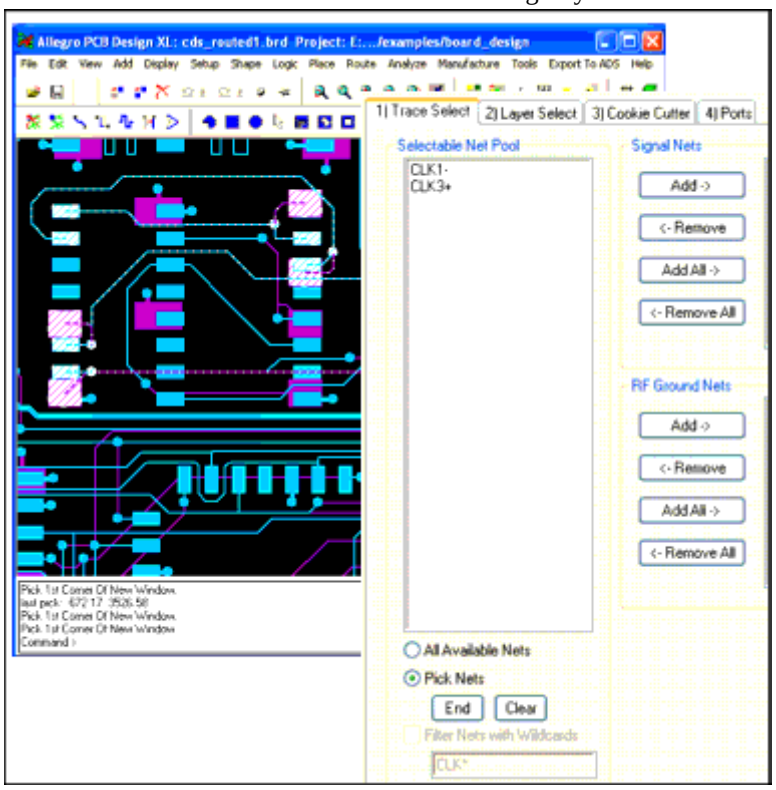
You can populate **Selectable Net Pool** in the following ways:

1. By selecting **All Available Nets** located below the **Selectable Net Pool** window.
2. By picking specific nets from the Layout window.

To pick specific nets from the Layout window:

1. Select **Pick Nets**.
2. Increase zoom on the required area.
3. Click the specific net or nets you wish to add to the **Signal Nets** list.

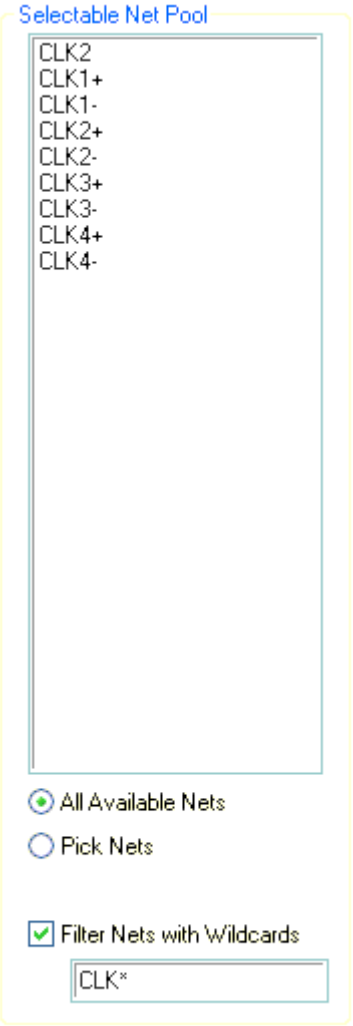
The selected traces are highlighted in the design and appear in the *Selectable Net Pool*, as shown in the *Nets selected from the Layout window* (allegrolink).



Nets selected from the Layout window

Optionally, you can filter the selectable nets displayed in the selectable nets list by clicking the *Filter Nets with Wildcards* check-box and providing a filter pattern, as shown in *Selecting nets using a wildcard* (allegrolink). Pressing **Enter** on your *keyboard* implements your selection and populates the *Selectable Net Pool*.

Note
This feature is available only when the *All Available Nets* radial button is active.

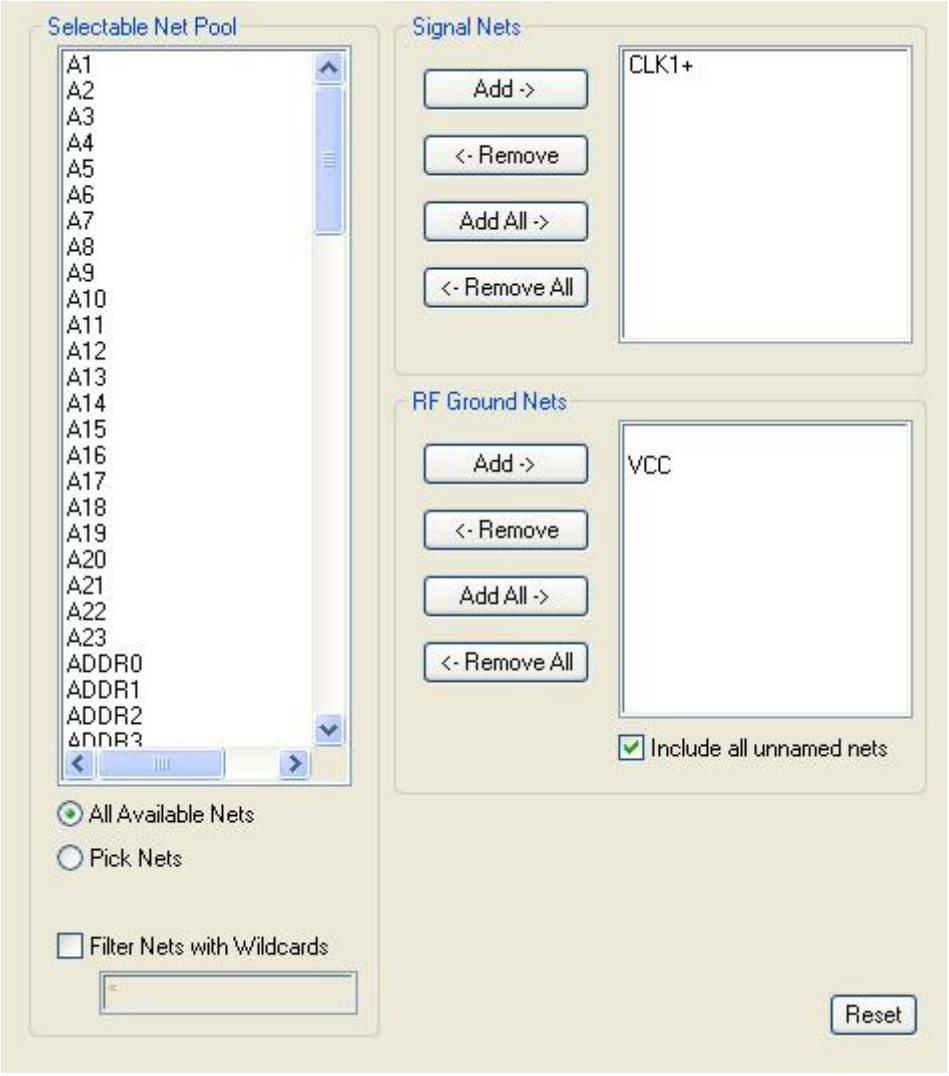


Selecting nets using a wildcard

Selecting RF Ground Nets

RF Ground Nets can be selected using the same steps required to select signal nets; selecting the nets from the list of all available nets, or by picking specific nets from the Layout window.

The **Include all unnamed nets** option allows you to select objects without an Allegro net name as RF Ground Net objects for exporting. When enabled the list of the RF Ground Nets in the select box will have an empty extra line at the top as indication of the extended net name filter used for all later processing. Individual selection of net objects without a name in Allegro is currently not possible because the ADFI tool has no reliable way to keep track of their selection state. If you want to use such objects in your Signal Nets, attach a name to the objects inside Allegro with **Logic > Net Logic**.



Selecting RF Ground Nets

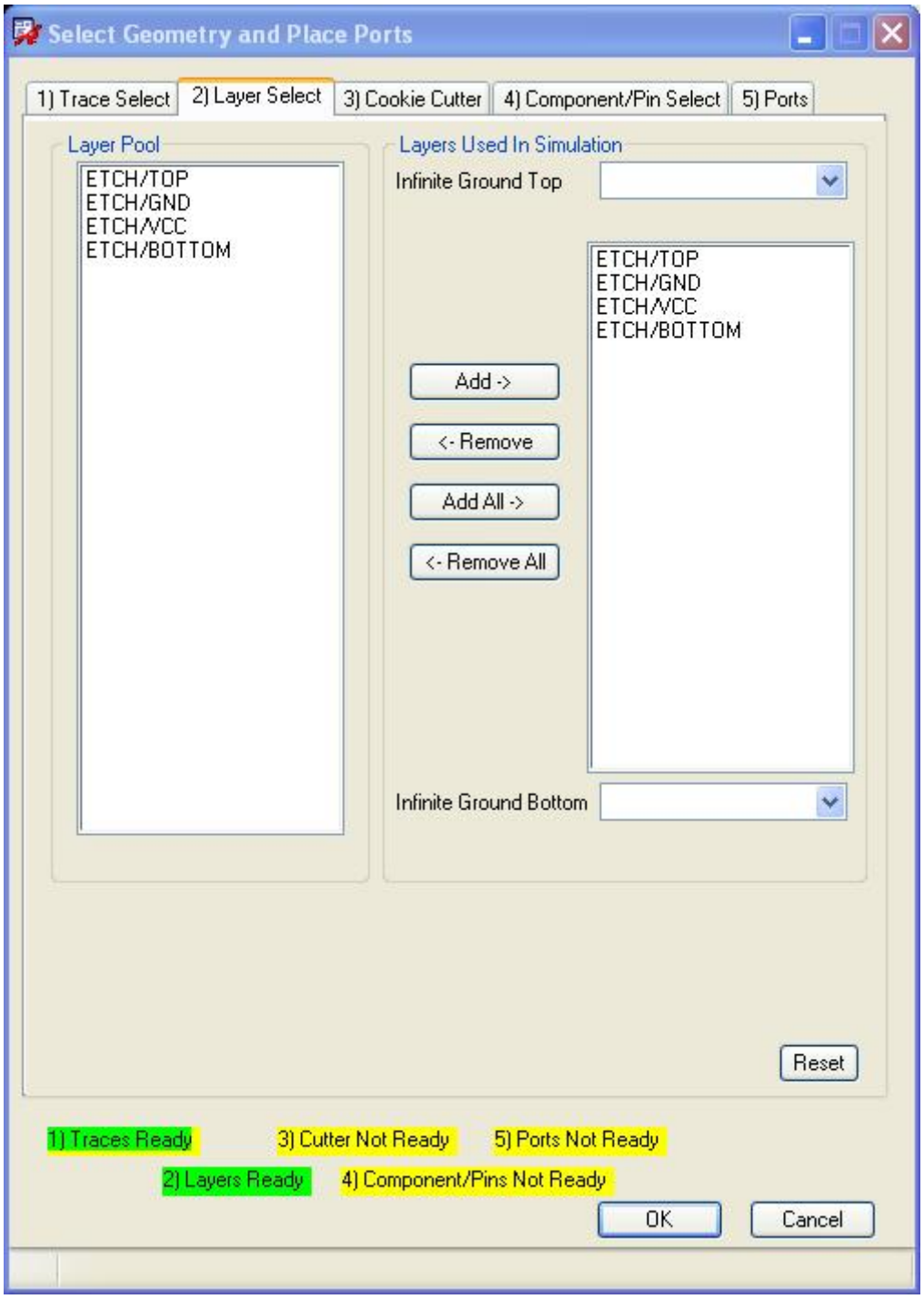
Once the **Signal Nets** and **RF Ground Nets** are selected, the yellow **Traces Not Ready** status bar is automatically updated by the green **Traces Ready** status bar. You are now ready to select layers using the **Layer Select** tab.

- In the list area, type a character to go to first match.
- Press the Tab key in net filter with wildcards to update selectable net pool.
- When using Pick Nets:
  - Watch out for the window / dialog focus!!!
  - Start/ Endstate button indicates if canvas is still in selection mode (?End indicates selection mode is active).
  - When in selection mode, watch the Design Object Find Filtersettings!!!
  - Turn off visibility of planes if needed.

Electrical objects like vias, traces, shapes not attached to a named net can all be selected as RF Ground objects
Cookie Cutter will cut through them

Using the Layer Select Tab

This tab enables you to choose the layers that will be exported to ADS. By choosing all the layers found, as shown in *Layer Select Ready* (allegrolink), the object will be exported from *all* layers and no additional infinite ground planes will be defined.



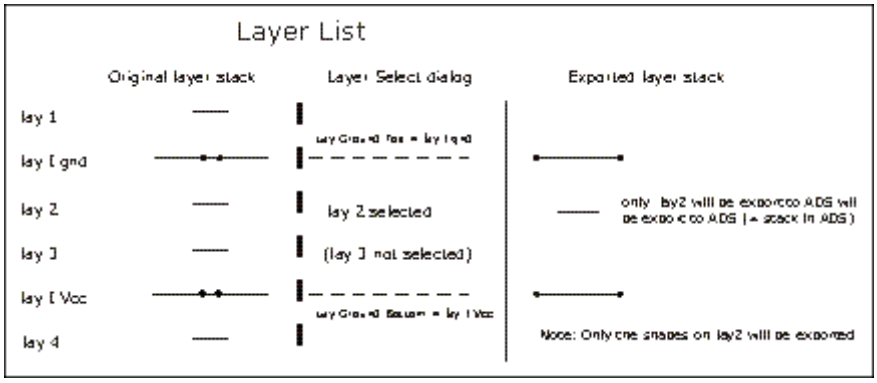
Layer Select Ready

To limit the board stack by adding infinite ground planes, choose the layer from the list in the *infinite ground* top or bottom drop-down boxes. The metalization patterns on those

layers will be replaced by a solid metal layer.

Note
The layers *above* Top or *below* the Bottom infinite ground selections will not be exported to ADS.

The graphic in *Example Layer Export List* (allegrolink) illustrates what will be exported when layers are selected in the *Layers List*.
As seen in this illustration, only the layers between the ground planes are available for export. *Lay 1* (which is above the top ground plane) and *lay 4* (which is below the bottom ground plane) will not be available for export. From the available list, (lay 2, lay 3) only *lay 2* was selected. Therefore, only the shapes located on *lay 2* will be exported.

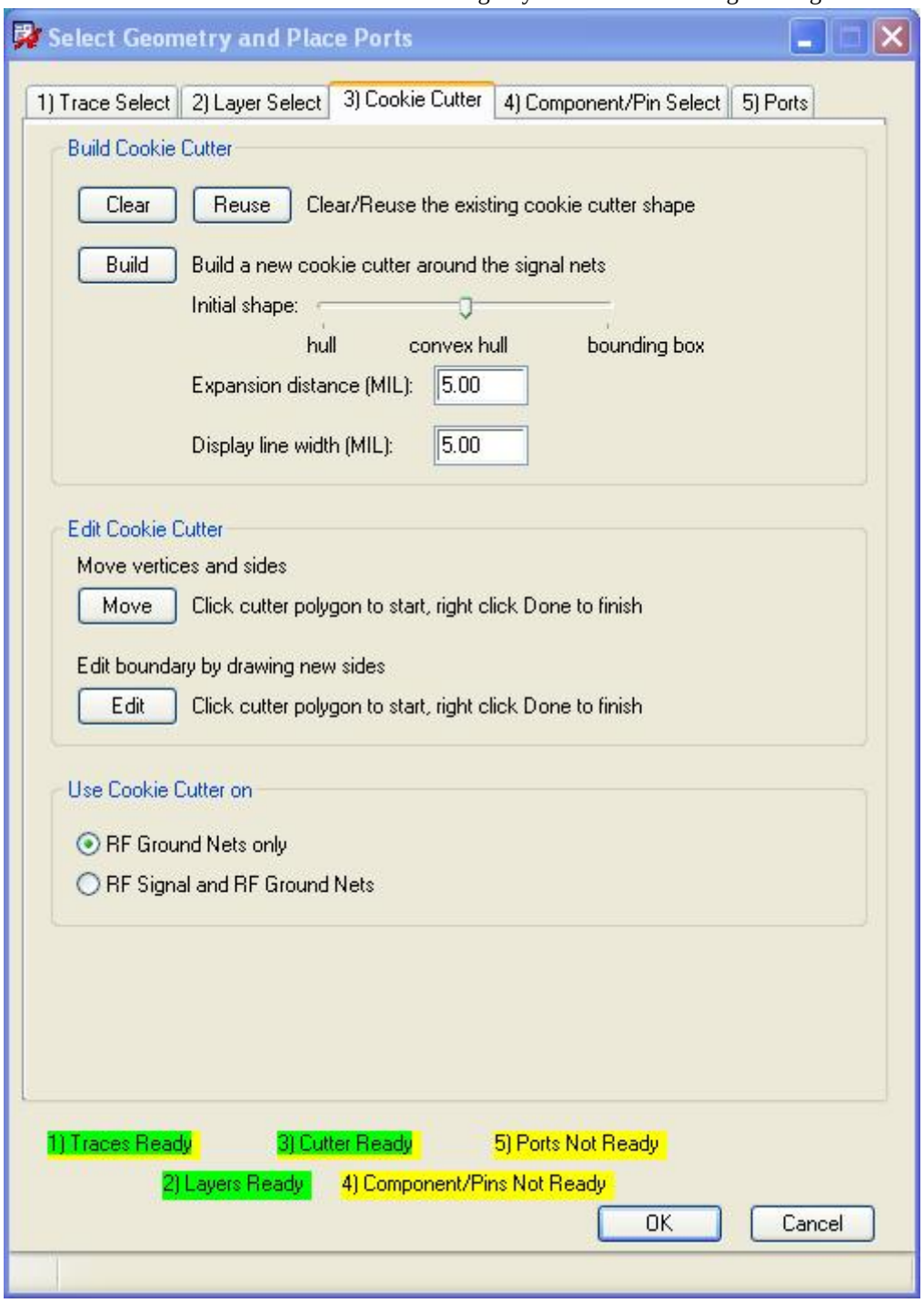


Example Layer Export List

- Export all layers unless there is a clear split of nets in the board.
- Making a sub selection of layers here will cut away pieces of the board layout and may invalidate the connectivity. So, use with care or stay away from it…

Using the Cookie Cutter Tab

The Cookie Cutter tab enables you to limit the exported area to an area that only contains the signal traces of interest.



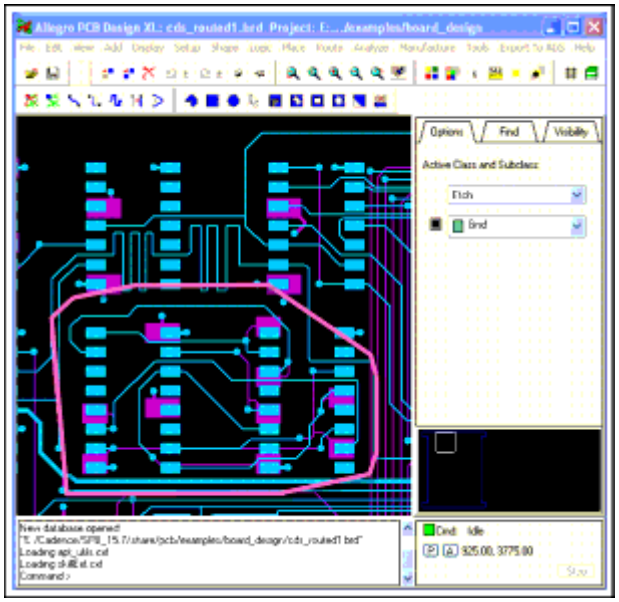
Building a Cookie Cutter

Using the *Build Cookie Cutter* section you can:

- Clear an existing cookie cutter polygon
- Reuse the cutter shape already present in the layout view
- Build a new cookie cutter polygon around the selected signal traces
- Modify the initial shape through with slider that allows you to gradually change the shape
- Set the expansion distance
- Set the display width of the cutter for visibility.

- Enable if the cutter will act on RF Ground only or on both RF Signal and Ground metal

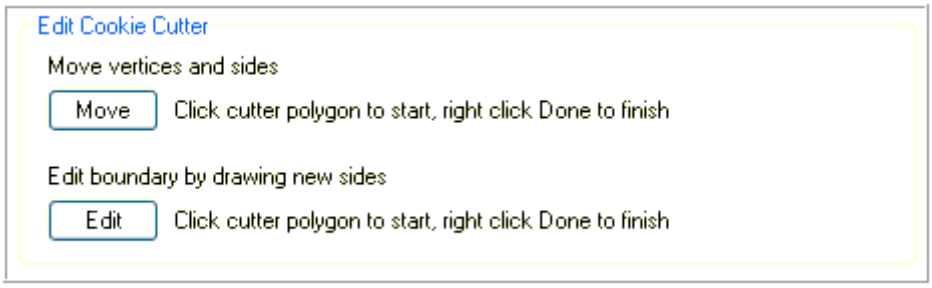
Setting the *Expansion distance* and clicking **Build** auto-generates a cookie cutter polygon around the selected signal nets. You can modify the shape of the polygon by using the *Initial shape* slider. This enables you to gradually change the cutter from the hull of the Signal Nets, over a convex hull up to the bounding box of the selected Signal Nets. The cutter polygon will be visible on a scratch layer of the design, and the first status indicators turn green. The display width of cutter can be modified to have optimal visibility of the cutter.
If the autoshape is correct, click **OK** , to dismiss the dialog.



Auto-generated cookie cutter.

Editing the Cookie Cutter

The *Edit Cookie Cutter* section enables you to alter the shape of the cookie cutter polygon:



In this section you can perform the following changes:

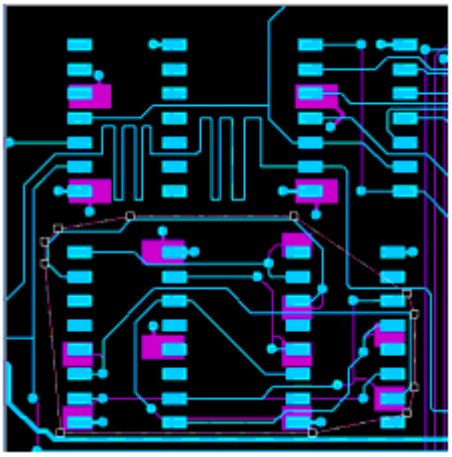
- Move vertices and sides: To select the cutter shape:
  - Verify the Filter settings.
  - If wrong shape is selected, right-click **Reject** to display a list of possible shapes.

Choose the scratch layer object.
- Once selected, you can move segment, vertices, delete vertices,…
- End your edit command by clicking **Done** or **Cancel**. This is visible by a change in the display line width of the cutter polygon

Avoid using the Edit the cookie cutter as polygon editing is complicated.

Example: Include a large signal net without enlarging the cookie cut (e.g. add PCLK net)

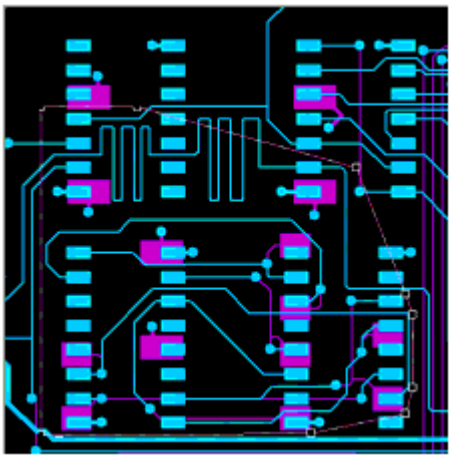
1. First make your selection without the large net.
2. Build the cookie cut.
3. Go back to net selection, add the large net to the signal nets. This will invalidate the cookie cut.
4. Go back to cookie cutter, and click **Reuse**.
5. You can now choose what to do with the large net: keep it or cut it. Click **Move** and select the cookie cutter polygon in the Layout window using a left click of the mouse.



This highlights the polygon and the nodes located at the vertices.

Note
If the wrong shape is selected (highlighted area), click the right mouse button and select **Reject** from the pop-up menu.
In the *Reject* item selection window, select the item on the scratch layer (by default *Manufacturing* / *Eem_Scratch*).

1. Select a point on the cutter shape, either a side (line segment) or vertex (node), click and hold the left mouse button: To move the selected object, scroll the mouse to the desired location and release the button.

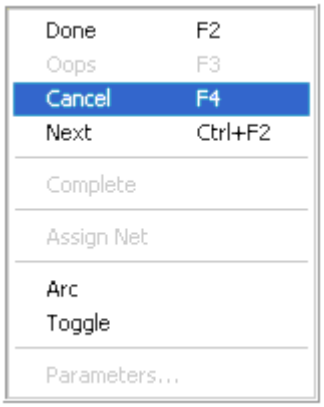


2. When finished, use a right mouse click inside the Layout window and click **Done**.

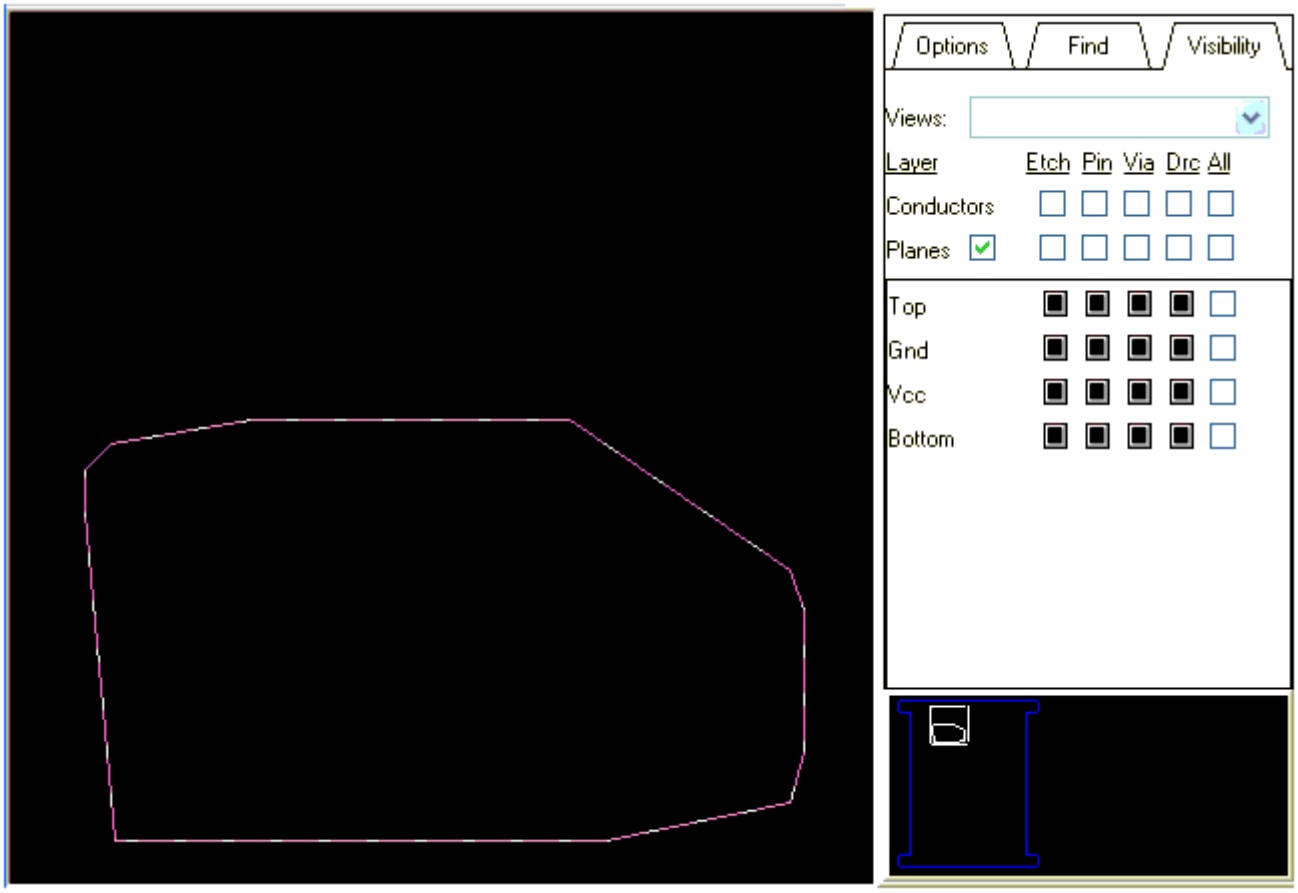
Editing the Boundary by Drawing New Sides

The *Edit boundary by drawing new sides* option enables you to edit the cookie cutter polygon by drawing new lines to reshape it.

1. Click **Edit** and *select* the cookie cutter polygon in the Layout window using a left mouse click.
   If the wrong shape is selected, right click the mouse and select **Cancel**.

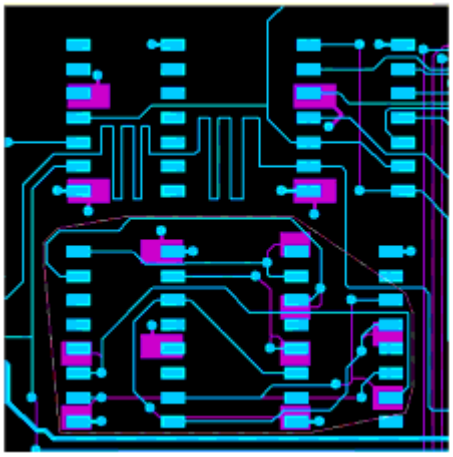


If you have difficulty selecting the cutter shape (located on the scratch layer: by default *Manufacturing / Eem_Scratch*) you can use the *Visibility* tab in the *Allegro Layout* window to turn off layers overlapping the shape, as seen in *Overlapping Layers "Turned Off" to Highlight Cutter Shape* (allegrolink).

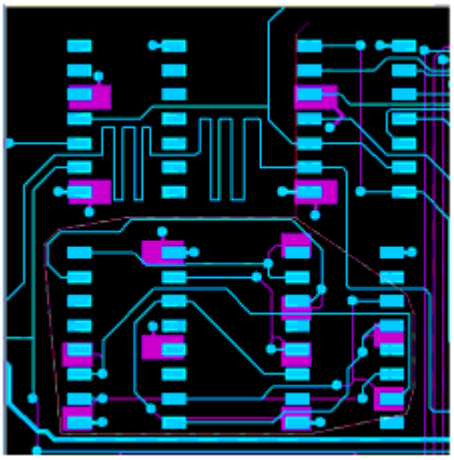


Overlapping Layers "Turned Off" to Highlight Cutter Shape.

1. Select a point on the polygon using a left mouse click. It can be either a line segment (side) or vertices.

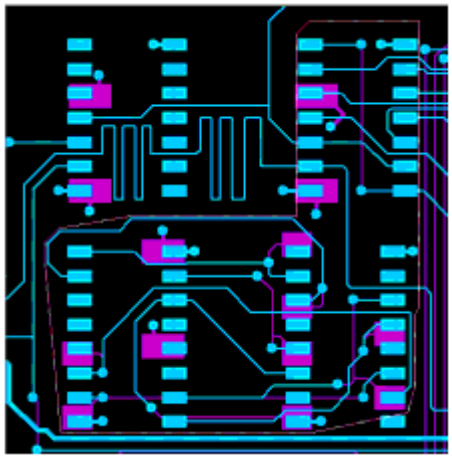


2. Move the mouse to a new location and click again to draw the new line/boundary.

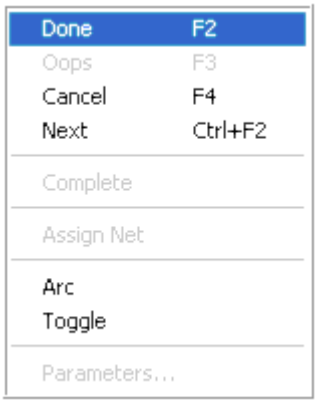


Continue doing this, as necessary, to create the new boundary.

3. To finish drawing the line, reconnect the line to the original polygon cutter shape. The boundary of the cookie cutter should now include the new area.



4. Right click the mouse inside the Layout window and click **Done** to complete the edit.



Note
You must use Reuse, Move or Edit cutter immediately after a *Load State* operation to get the Cutter Ready indicator green if you want to reuse the loaded cutter. Just push Move or Edit and click **Done** to complete the edit immediately does the same as Reuse. The cutter is verified and the Cutter Ready indicator will go to green automatically.

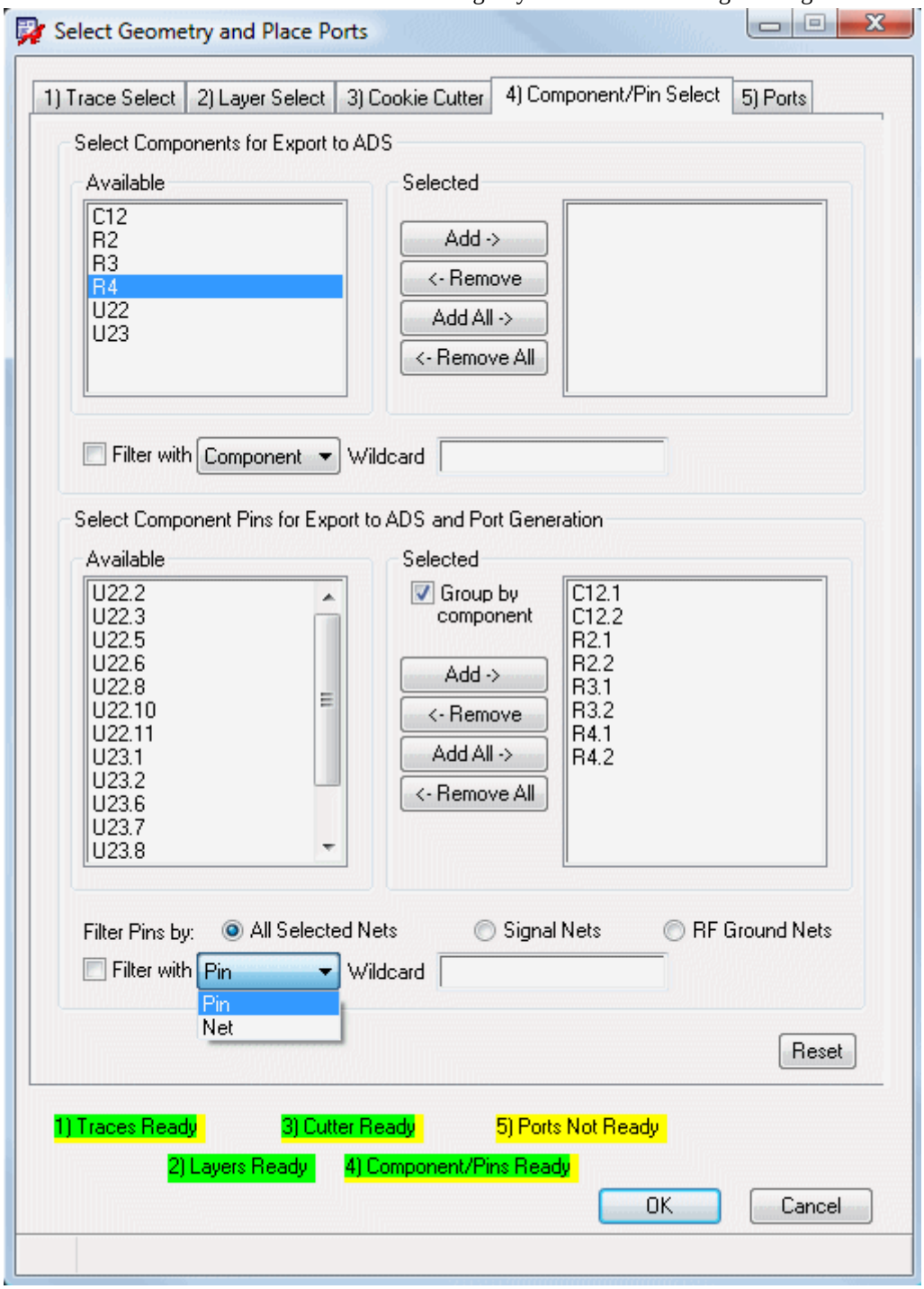
Note
If you have a large net selection it can be interesting to create a cutter on a small selection first, use edit to modify the cutter to its correct size and modify the signal selection afterwards. The cutter will be invalidated in that case but Reuse allows you to immediately go forward without the time consuming outline lookup for large and complex net selections.

Selecting Objects for Cookie Cutter

The **Use Cookie Cutter on** radio buttons allow you to modify the cutter behavior. By default, the cutter is created around all RF Signal Nets and limit only RF Ground Nets to the cutter shape. It will not cut the RF Signal Nets with the first option even if you modify the cutter to cross the RF Signal Nets. In the second mode the cutter will sill be build around the RF Signal Nets but if you modify the cutter that it crosses the RF Signal Nets all objects will be limited to the cutter shape.

Using the Component/Pin Select Tab

The **Component/Pin Select** tab enables you to make a filtered selection of the component pins needed for simulation port generation in the *Ports* tab.



The **Select Components for Export to ADS** group is organized around the components, while the *Select Component Pins for Export to ADS and Port Generation* group is organized around the component pins. Similar to the *Trace Select* tab, the left side of this tab provides a list of available components and pins, while the right side shows the lists of selected components and pins. When a pin or component is selected, it disappears from the **Available** lists.

The Add and Remove buttons enable you to move one or more entries between the **Available** and **Selected** lists in both the upper and lower group boxes.

Filtering is available by selecting **Filter with <type> Wildcard** below the *Available* and

Selected lists in both the upper and lower group boxes. These filter settings act on the items visible in the *Available* and *Selected* lists and don't change the selection. For items in the *Select Components for Export to ADS* group, wildcard filtering is available that allows you to filter on either instance names or on the connected nets to the components (from the selected nets lists). For items in the *Select Component Pins for Export to ADS and Port Generation* group, you can also make a distinction between the pins connected with nets of all select nets, RF Signal or RF Ground nets. The wildcards filter works either on the pin names directly or the connected nets to the pins.

Initially, when this tab is opened, the available components contain a list of all components with at least one component pin connected to a RF Signal Net, or RF Ground Net inside the cutter region. No components are selected and filtering is disabled.

The *Available* list in the *Select Component Pins for Export to ADS and Port Generation* group contains all the component pins of the nets that are selected inside the cutter region. To begin with, the *Selected* list will only contain the RF Signal Pins. These pins are used by default for port generation in the *Ports* tab.

When a component is added to the *Selected* list, all pins associated with the component are added by default to the *Selected* list in the *Select Component Pins for Export to ADS and Port Generation* group. The selected pins are shown grouped together as *Ref. Designator <n1,n2,...>* when the *Group by component* checkbox is selected. When a component is removed, a dialog will appear asking if you want to remove all the component's pins from the *Selected* list in the *Select Component Pins for Export to ADS and Port Generation* group.

Note
If you keep the pins, they will lose their visual grouping in the *Selected* list in the *Select Component Pins for Export to ADS and Port Generation* group.

Visually grouped pins cannot be manipulated in the *Selected* list in the *Select Component Pins for Export to ADS and Port Generation* group. To do this, you must first disable the *Group by component* flag. Once this is done you will be able to change the list of pins to only the pins you are interested in during port generation.

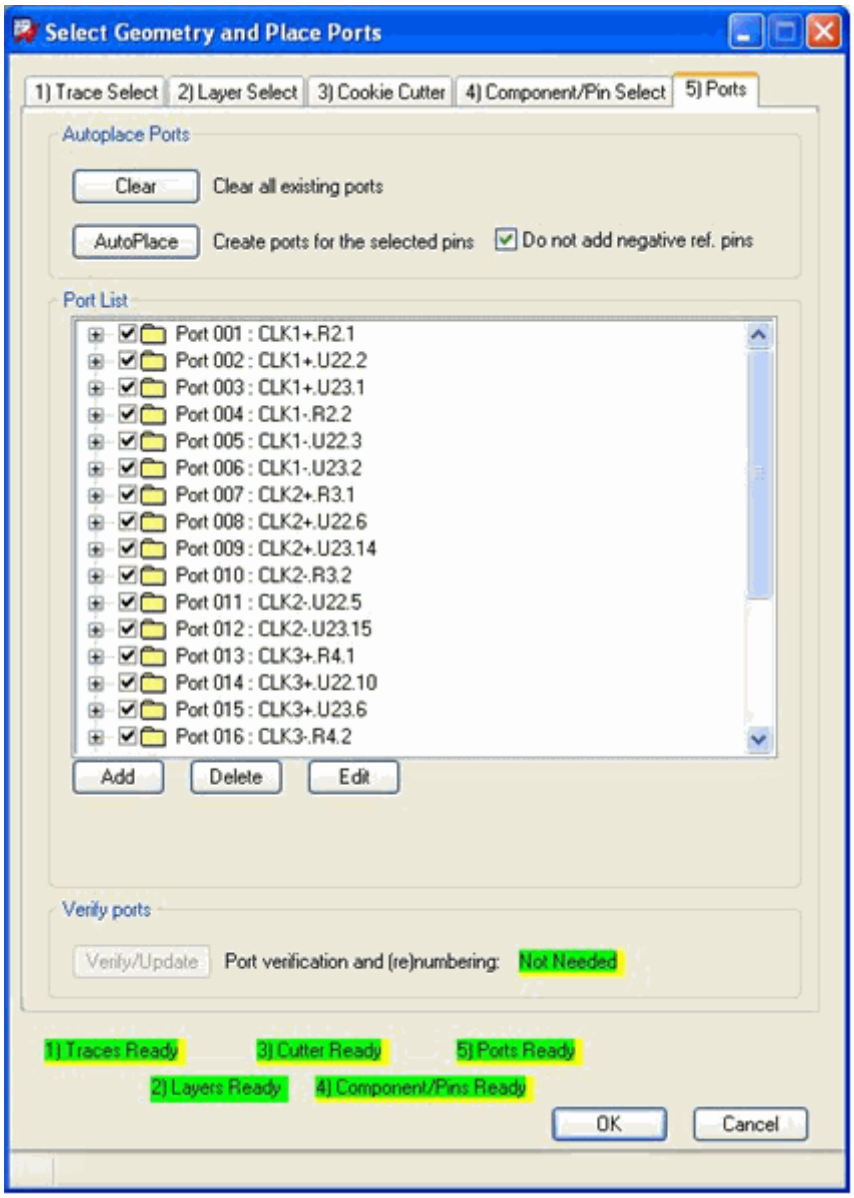
Select the components to be transferred in a hierarchical export:

- Available components have at least 1 pin on the selected nets (both signal or RF ground) and are within the cookie cutter.
- Wildcard filtering can be done based on component name.
- Select the pins that will be used for EM port generation.
- Available pins are on the selected nets (both signal or RF ground) and within the cookie cutter.
- The Selected Pins can be grouped by component.
- Wildcard filtering can be done based on pin or net names.
- By default, all available pins on signal nets will be inserted in the selected set.
- When adding a component to the selection, e.g. R5, all available pins of that component (on signal or RF ground nets and within the cookie cut) will be added to the selection (and can be shown as grouped).
- If you want to remove pins of selected components, disable Group by component.

- Reset will clear the entire selected pin list. Go back to cookie cutter and return to re-initialize the default selection.

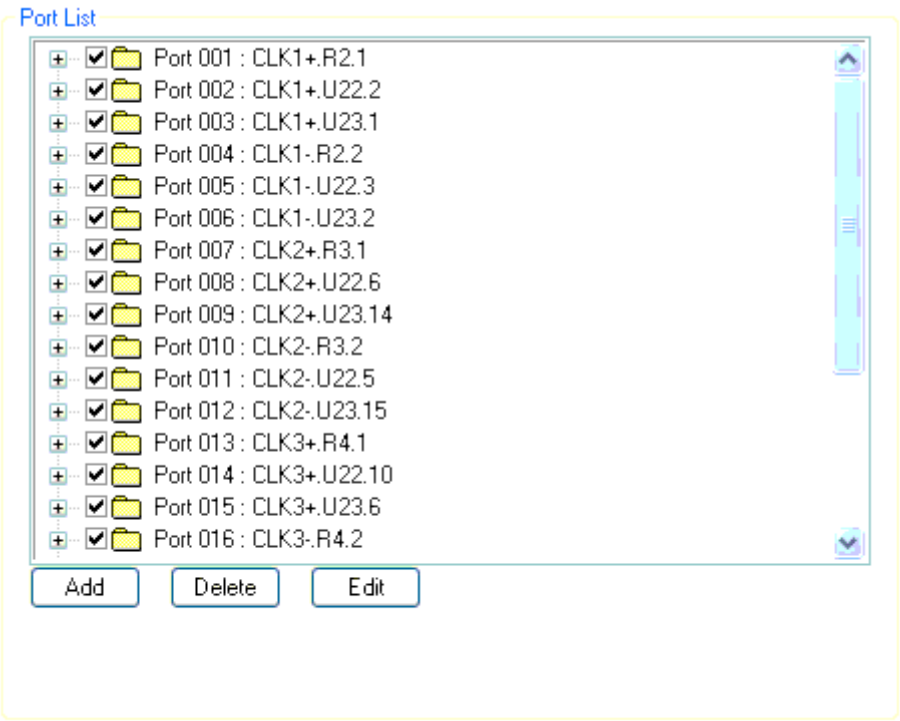
Using the Ports Tab

The **Ports** tab enables you define the simulation ports based on your selected nets.



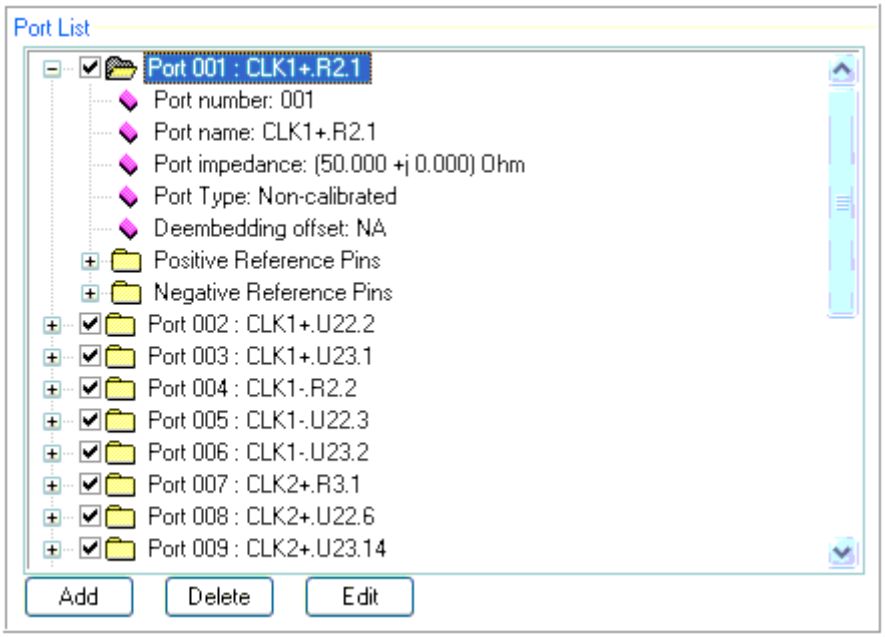
Auto Creating Ports

Selecting **AutoPlace** enables you to automatically generate a port for every selected component pin (pad) in the **Component/Pin Select** tab. The result is then visible in the *Port List* window, as seen in *Port List Window* (allegrolink).



Port List Window

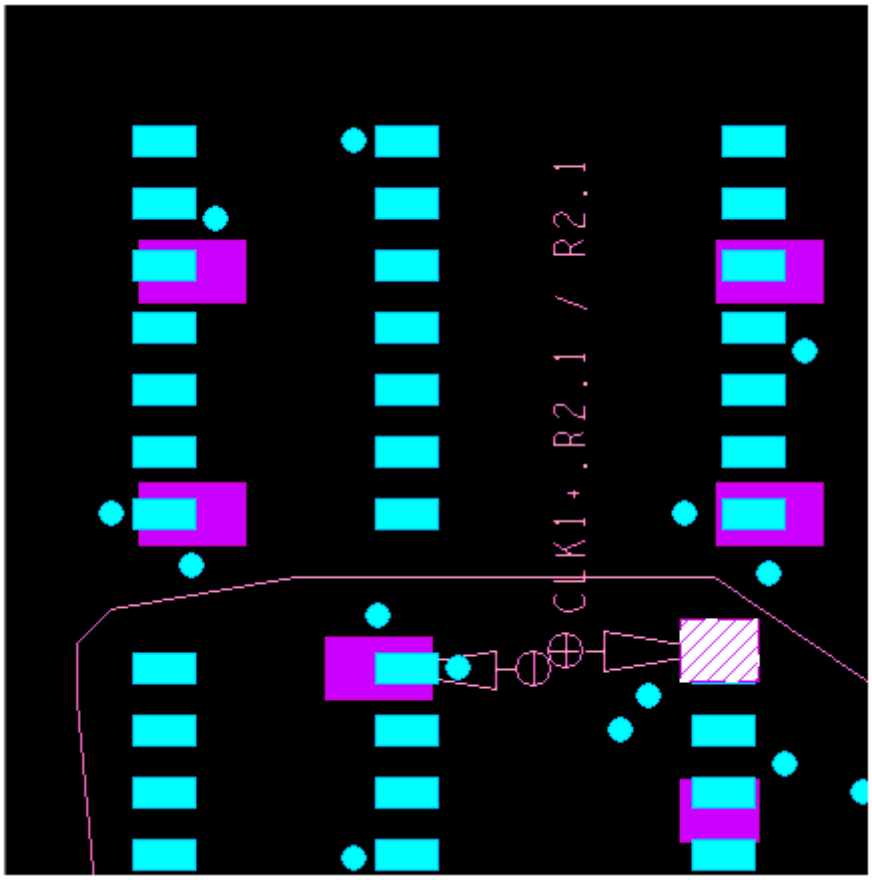
You can expand each port definition in the port list to make the properties of a port visible, as shown in *Expanded Port Definition* (allegrolink).



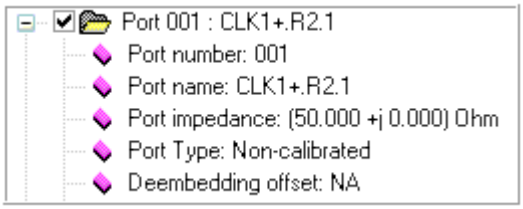
Expanded Port Definition

When a port is selected, pointer symbols are automatically placed in the layout with the

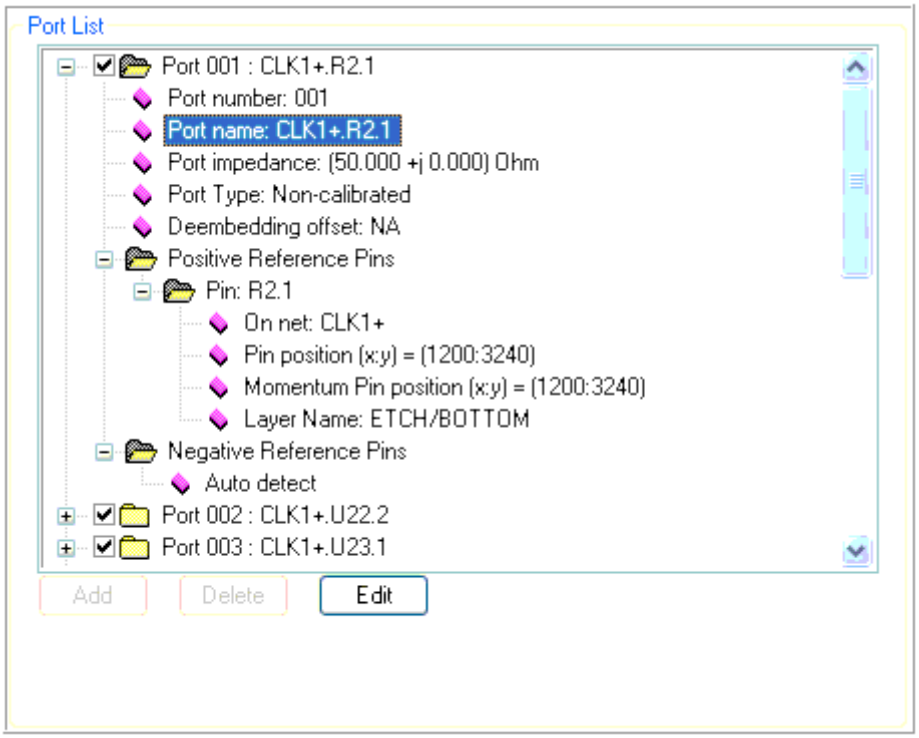
exact positions of the positive and negative reference pins of that port.



By using the check box in front of each port you can *enable* or *disable* a port for later export. This way you can store the port definition, if it was heavily modified for later use.



The modifiable properties for a particular pin or port are indicated by whether or not they have active *Add*, *Delete*, or *Edit* buttons. Unselectable, or inactive buttons will be *grayed out*.



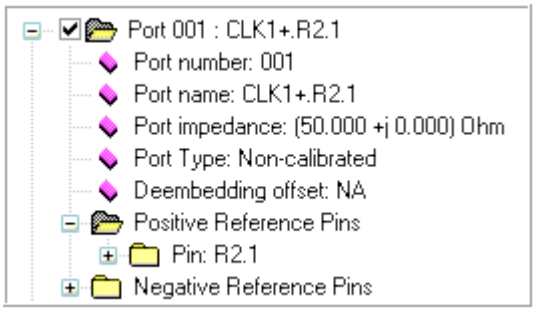
Port Properties

As shown in *Port Properties* (allegrolink), it is possible to change the *Port name* because the the *Edit* button is active/enabled when that property is selected. However, if *Port number* is selected, none of the change options are available and all are grayed out or inactive since the *Port number* cannot be changed.

Port Properties

Port properties include:

- Port numbers: Port numbers are automatically assigned by the tool. They are consecutive starting from 1 and must be regenerated in order for a port set definition to be valid.
- Port name: The port name is auto-generated as < *net name* >.< *instance* >.< *pin nr* >. However, this name is modifiable and can be used to sort automatically generated ports.
- Port impedance: The default port impedance is 50 Ohms, but this can be modified to meet your specifications.
- Port Type: Only non-calibrated ports are available in this version.
- Deembedding offset: Deembedding distances can not be used in this version.
- Positive and Negative reference pins: Both positive and negative reference pins are available.



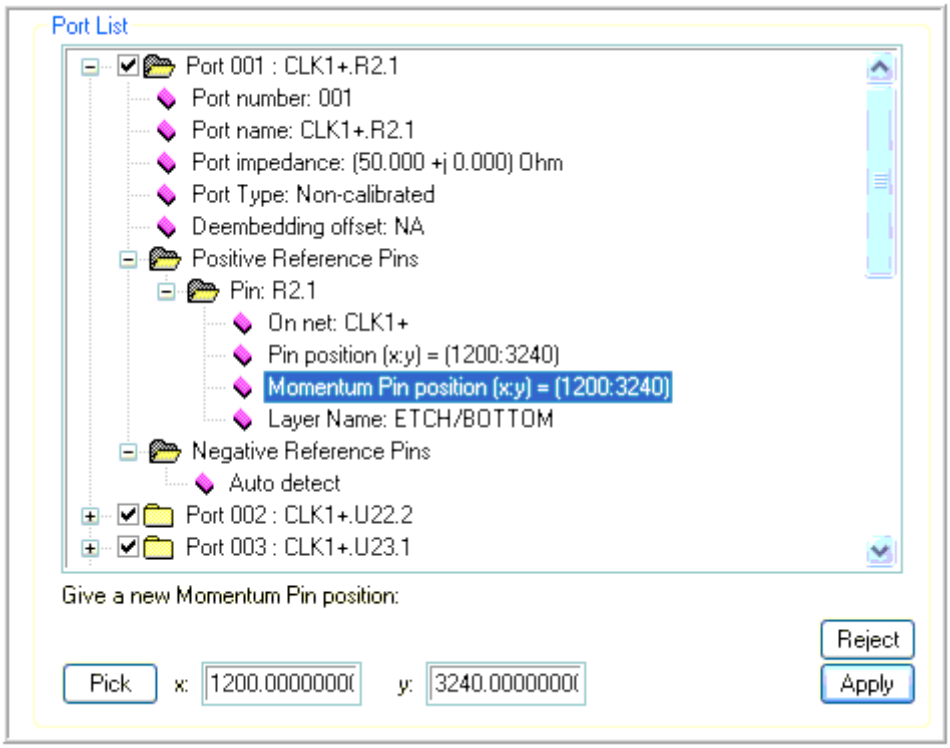
Pins are associated to ports as either positive or negative reference pins. These pins have both a standard pin position (normally the device's pad position) and a momentum pin position that can be modified for better simulation setups. Pins are also defined to be on a specific metalization layer. However, on multi layer device pins, the layer setting can be modified to valid simulation layers in the setup.

It is important to note that a port is *always* defined between a positive and negative reference pin and that both reference pin types must be *manually* defined for each port, under normal circumstances. If not, the simulation results will be wrong. By exception, if some form of infinite ground layer definition exists in the structure, the negative pin will be omitted. In this case, the Momentum simulator uses an implicitly defined port on the ground layer. Infinite ground layers can be derived from existing negative plane definitions in Allegro, or by defining them in the *Layer Select* window.

Positive reference pins will only appear on the component pins of selected pins and the negative pins are normally component pins of an *RF Ground Signal*. However, the auto generation feature can also use a nearby *via* or *point* pin on a shape that is part of an *RF Ground Signal,* if it runs out of an RF Ground Signal devices pins that are close enough. These last point pins have the following naming convention: *NonDevRefPin*<type><signal><initial location>.

If an infinite ground is present in the structure the automatic port generate mechanism will rely on the implicit ground reference pin feature of Momentum and not generate a negative reference pin. Enabling the *Do not add negative ref. pins* flag disables the automatic generation of the negative reference pins. In this case, setting up the correct reference becomes your responsibility.

The Momentum pin position can be changed within the area of the shape for which it was defined. To do this, click the *Momentum Pin position* : in port properties and select the *Edit* button. This provides you a coordinate *Pick that* enables you to select a new pin position as shown in *Momentum Pin Position* (allegrolink).



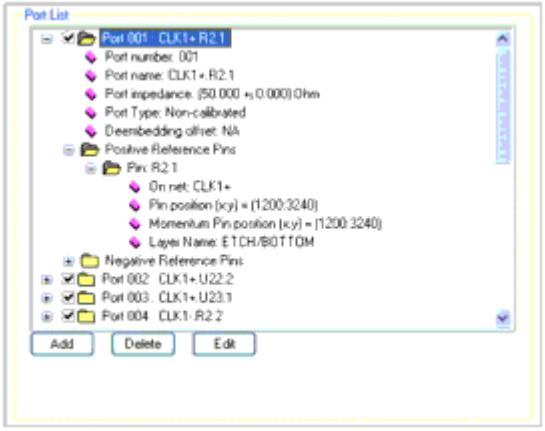
Momentum Pin Position

Using the *Apply* or *Reject* buttons enable you to accept the change to the port definition or cancel it. This also returns the interface back to the default configuration.

Delete

This button, shown in *Deleting a Port* (allegrolink), enables you to remove specific pins or port definitions from the list.

Note
When ports or pins are deleted the component pins become available for manual port and pin placement.



Deleting a Port

Clear

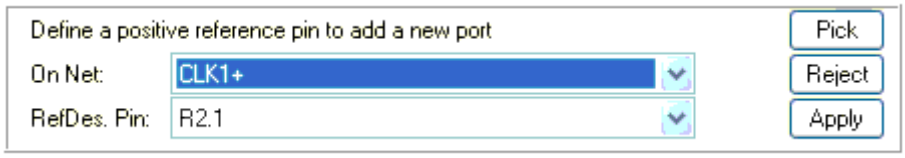
Selecting *Clear* removes all existing ports.

You should note that whenever you modify the port properties, or change layer definitions you must perform a *Verify/Update* to ensure your port definitions are still consistent with the available layer and cookie cutter settings. This step also assigns consecutive port numbers for export and simulation.

Note
Port numbering *must* be consecutive for the port state to be valid.

Manual Port Creation

When one or more *Signal Net* device pins are not associated with any port, it is possible to manually create a port. Selecting the *Add* button enables you to create a new port by selecting one of the free *Signal Net* device pins from the pin selection interface. The pin can either be selected from the list boxes, or by using the *Pick* button to select the pin in the Layout window.



Selecting *Apply* adds the port into the port list. If a port was already selected in the port list, the new port will be placed immediately behind that port. If no port was selected, the new port is created at the end of the list.

When possible, the port creation code tries to add an appropriate *Negative Reference Pin* to the port definition.

Manual Pin Association

It is possible to use manual pin association if you wish to add more *Negative Reference* pins, or when more complex ports must be created with multiple grouped pins (either Positive or Negative Reference pins).

In order to do this, select in the port's definition in the *Port List* , the Positive or Negative Reference Pins tag and select the *Add* button, this opens the *pin select interface*. Next, select the *Net* in the *Select* window and find the pin name that you wish to add.

A *Pick* capability is activated enabling you to select *Signal Net* component pins for *Positive Reference* pins, *RF Ground Net* and *Signal Net* component pins for *Negative Reference* pins.

For *Negative Reference* pins you can also create non device point pins by selecting the *Non Device Reference Pin* option and using the *Pick* button. You can then place a pin anywhere on the selected net.

Note
No validity checks are performed when you place the pin.

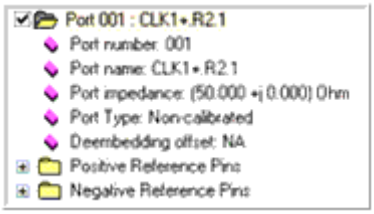
A *NonDevRefPin...* will be created when a valid position was picked.

Use *Apply* or *Reject* to make or drop the changes to the port definition.

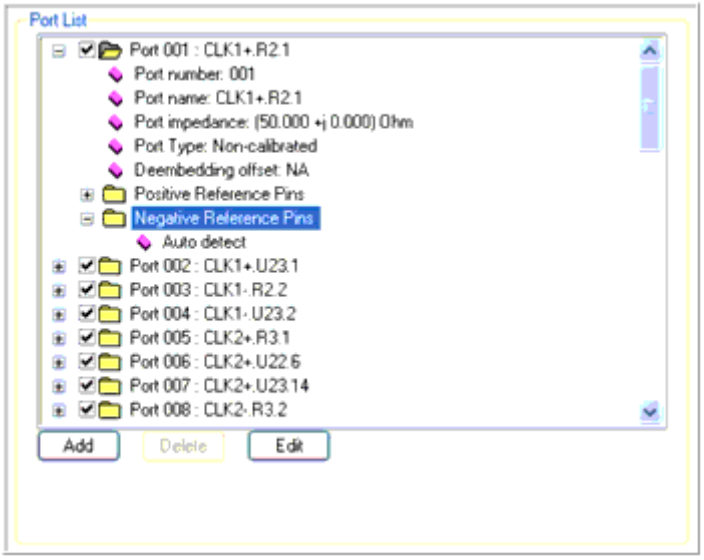
Creating a Negative Reference Pin

In certain situations (e.g., where only finite positive shapes are available), it may be necessary to manually add a negative reference pin to a port.
To do this:

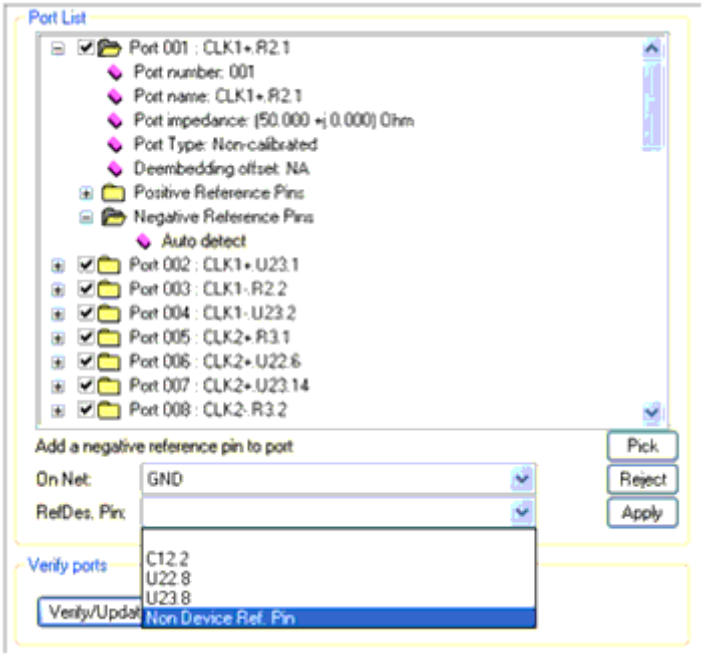
1. Select a port and open the tree view to see the list of properties and pins associated with it.



2. Select *Negative Reference Pins* , open the tree view and choose **Add** , as shown in *Allegro Design Flow Integration* (allegrolink).

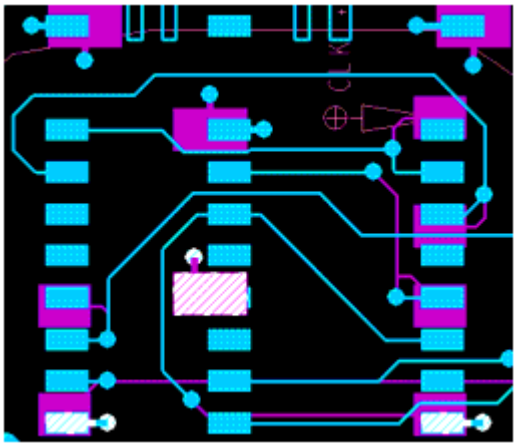


3. In the *Add a negative reference pin to port* section, select **GND** from the dropdown list of available nets and **Non Device Ref. Pin** from the *RefDes Pin:* drop-down list.



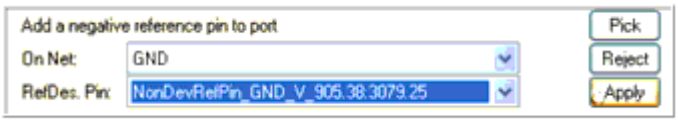
Adding a Negative Reference Pin to a Port.

1. Next, choose **Pick** and select a "highlighted" shape on the Allegro board to create the pin.



Available Shape Selections on the Allegro Board.

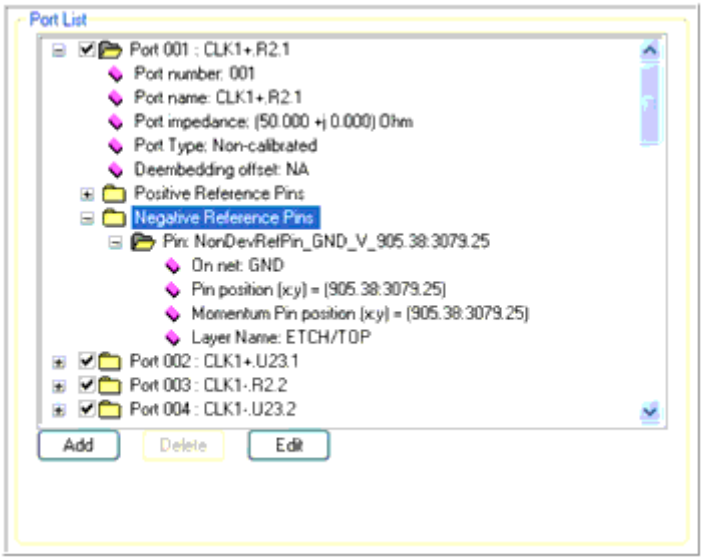
Once selected the pin name will appear in the *RefDes. Pin:* section as shown in *RefDes Pin Name* (allegrolink).



RefDes Pin Name

1. Select **Apply** to add the new pin to the port.

The new pin and its properties now appear in the *Port List* as shown in. *New Pin Added to Port List* (allegrolink).



New Pin Added to Port List

Note
It is import to remember to *update/verify* the port after adding the pin.

Ports Generation Hints

An S-parameter port has a positive and a negative terminal (or reference). Each terminal connects to one or more pins. In the absence of a pin connection, the terminal is assumed to be connected to the reference pin of the S-parameter model.
AutoPlace will generate S-parameter ports from the selected pin list automatically, which is sufficient for standard cases.
Three flags control the port generation:

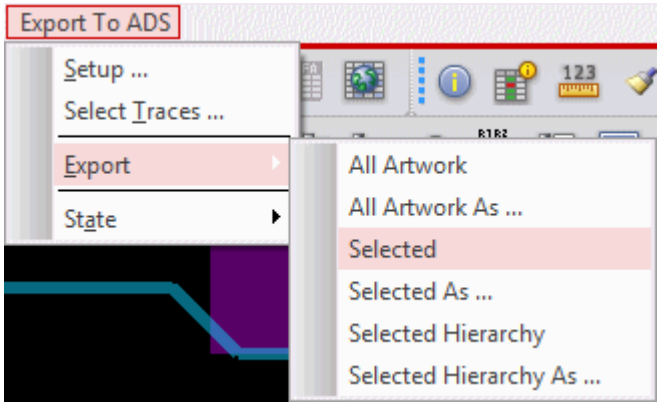
- Do not add negative reference pins: In case there is no infinite ground (no negative layer), the standard and automatic creation of negative reference pins on the RF ground nets is suppressed by enabling this flag. Enable this flag when:
  - Proper S-parameter port references will be defined once in ADS. Make sure that RF ground pins were added to the Selectedpins so that can serve as reference pins once in ADS.
  - You know that an infinite ground will be added in the ADS substrate. In such case, that infinite ground plane will give a physical meaning to the reference pin of the S-parameter model.
- Combine on discrete: when the two pins of a discrete component are added to Selectedpins, a single S-parameter port will generated from these two pins.
- Start port name with ref. des.: will start the automatic port generation with instance name at start and sort accordingly

You can refine the setup, e.g. combine 2 pins to define a port with a positive and negative reference pin.

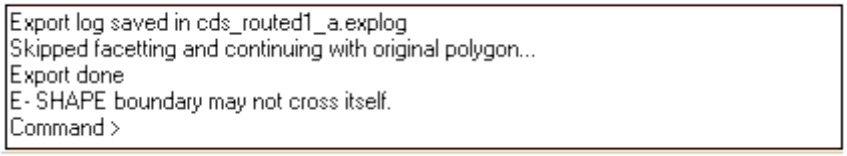
- Delete a port to free up the pin(s)
- Go to the port you want to add the pin to,select Positiveor Negative Reference Pins,click Add, select the free pin you want to add,click Apply.
- After editing the ports, click Verify/Update.
- Notes:
  - Multiple pins can be grouped as positive or negative reference pins of a port. However, Momentum doesn?t support this (yet)…
  - Manually create special ports first. This keeps their position at the start of the port list.
  - A single pin cannot be used to define multiple S-parameter Ports. E.g. two S-parameters ports sharing a common reference pin. Momentum doesn?t support this (yet).

Export to ADS

The final step is to save the selected design information for import later in ADS. To save the design, choose **Export to ADS** > **Export** and select one of the six export modes.



Once the export complete, an entry will appear at the bottom of the Allegro log window indicating the export was successful or failed.



Note
Errors appearing below the export success information indicates that the Allegro merge operation failed. However, code has been added to work around those failures, so they can be ignored.

The export selections provide you with three export modes which we describe in the following sections.

Geometry only export

Note
Only recommended to use this for small designs and testing purposes. It provides no control and drops almost all relevant information needed to create an EM setup for the design.

Choose **Export To ADS > Export > Export All Artwork** to:
Automatically saves the mask layout of entire board or package. This selection creates the output files and automatically puts them in the directory where the board file resides. The exported EGS filename will be the same as the board name with the extension *_a*, and no suffix. For example, if the board name is *cds_routed1* then the exported filename will be *cds routed1_a*.

The **Export To ADS > Export > Export All Artwork As** also saves the entire board layout as above, but it allows you to choose the name and location you wish to save the design to.

Flat export without components

Choose **Export To ADS > Export > Selected** to:

- Automatically save selected features of the design. This export generates an flat EGS

artwork file, a port file and single cross-section definition. As with *Export All Artwork,* the output files and automatically placed in the directory where the board file resides with a basename identical as the Allegro design name. The exported EGS filename will be the same as the board name with the extension *_a* , and no suffix. For example, if the board name is *cds_routed1* then the exported filename will be *cds_routed1_a*. The *.ads* file describes the list of files needed for the import into ADS.

- During the export, Allegro can generate warnings/errors about shape creation and boolean operations. These can be ignored. Errors like "unbound variable" or "fprintfwrong type of parameter nil_ „E-**Error** ilRplacd: Illegal operation: first argument in static space -(nil)" should not be ignored. These are most frequent export errors. When this occurs, try a slightly different selection or setup setting i.e. (pathType"mitered" for traces is a common problem)

Choose **Export To ADS > Export > Selected As** to:

- Enables you to choose the name and location you wish to save the selected area of a design to.
- However, realize that in some versions of the Allegro DFI the design name inside the files remains unaffected. Import in ADS will redefine that design.

The export generates multiple files: four are used during the import of a flat export/import which are as follows:

- <name>.ads : import control file
- <name>.slm: substrate stack for Momentum/EMDS
- <name>_a : EGS file with geometries
- <name>_ports.ael : ports definitions

The <name>.ltd file is the substrate stack in Momentum technology format view/edit with the $HPEESOF_DIR/bin/eesofsubed(.exe) subtrate editor and export slm from here.

Note
Since the ADS 2009 Update1 version:

- Like other layers, drill hole layers get unique color
- Layers are ordered according to the cross section.

Hierarchical export with selected components

Choose **Export To ADS > Export > Export Selected Hierarchy** to save two export variants of the selected features in the design:

- A flattened version as in *Export Selected*
- Hierarchical version of the design with components which is stored in hierarchical (assembly) design with the name <name>_adfi.

Choose **Export To ADS > Export > Export Selected Hierarchy As** if you want to change the name and location that you want to save the selected area of a design to.

- A flattened version as in *Export Selected As* is stored

- The hierarchical version is combined together in a top level design file with the name <exported designname>_adfi.

The top-level <design>_adfi design contains:

- An instance of the man board, module or package design (similar to the flat export version)
- Instances of all selected components, including their selected pins, and parameters attached to them.
  In hierarchical mode all types of selected components, bondwires, bonding balls, diestacks, interposers, spacers, BGA bonding balls (if defined) will be exported as subdesigns.
- Layout Ports with properties that define the S-parameter Port setup needed for an EM simulation.

Cross-section information in the hierarchical designs is used in the following way:

- The hierarchical mode exports individual cross-section definitions for the main substrate, the substrate stack above and substrate stack below the main substrate.
- These substacks are attached to the respective sub designs.
- For the top level design the various stacks are merged into a single substrate containing dielectric blocks for DIE, interposer, spacer objects.
- In this mode metal layers are exported as thick conductors unless they are defined as shield (plane layers) in the Allegro cross-section definition.

In APD and Cadence SiP bondwires and BGA solderball models can be attached:

- Bondwires attached to a single component are exported as 3D-EM SBOND instances inside a WB_<instance name> design.
- When used in an ADS schematic these bondwire components can netlist as Philips Bondwire components combining all the bondwires together.
- You can also create an EM model using the Momentum or FEM engines for them if you want.
- The profile and bondwire properties defined in Allegro are automatically picked up and transfered into ADS. The SBOND components can have up to seven segments and Allegro wire profiles are limited to that number. More complex profiles will loose their last segments.
- SBOND definitions are automatically translated into Philips Bondwire arrays (which must have exactly five segments). If less than 5 segments are available the last segment is cut up until five segments exist. When more are available some points will be dropped until we have exactly five segments.
- BGA Solder balls will show up as pin components with Momentum vias but they contain enough information to replace the balls by 3D EM solder ball components in ADS. This process has not been automated so far but is planned for a later release. If placed in and ADS schematic the solderball components will by default netlist as ADS tapered VIA components.

When you select instances in the Component/Pin tab in the ADFI for export they will export as individual components designs for ADS:

- R,L,C, Diode components are recognized (when possible) and replaced with their equivalent symbol for use in ADS schematics.
- These components will netlist as the ADS builtin components, but all parameters known on the Allegro side for a component are transfered to the ADS side automatically.
- Allegro parameters will translate in component parameters or as instance properties depending on the situation.
- If you only select the component pins and not the compoment the pins will only be used for port creation in ADS but the component will not be included as subdesign.

The Allegro Import Design Kit in ADS must be version 2.6 or later to load this hierarchical export of a design.

Difference between Cadence SiP and Allegro Package Designer:

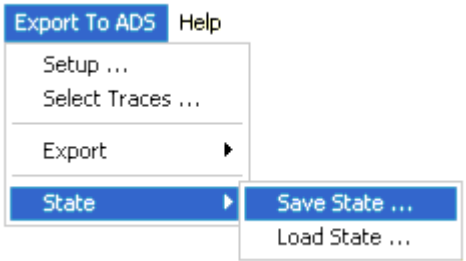
- In Cadence SiP Allegro DFI will export:
  - stacked IC dies, interposer as dielectric bricks
  - it cannot export the molding covering stacked designs as this information is not stored in the SiP database.
- In Allegro APD the Allegro export:
  - has no diestack definition available for the stacked devices and interposer or spacers. The export for IC dies will be limited to the pin layers only. No dielectric brick will be created for an IC die or spacer in APD.
  - APD non-substrate layers have physical properties and allow to define a mold material around the chip and will be exported as background material.

The hierarchical export generates multiple files: five are used during the import of a hierarchical export

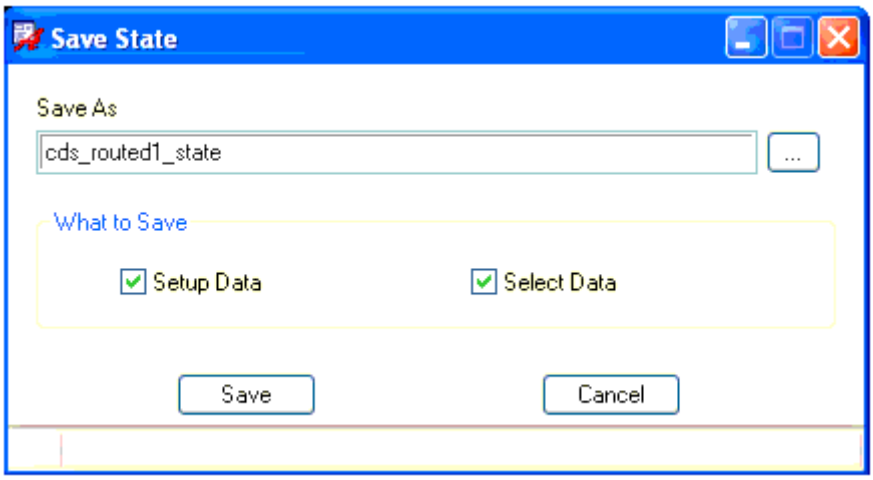
- <name>.ads : import control file
- <name>.adfi: xml descriptions with hierarchical export
- <name>.slm: substrate stack for Momentum/EMDS
- <name>_a : EGS file with geometries
- <name>_ports.ael : ports definitions of flat export part

Saving and Loading States

Once you have selected the Nets of interest and completed configuring their port properties, you can save this state or configuration to a file for later use or further testing. To do this, from the Allegro main menu select **Export To ADS > Setup > Save State...**

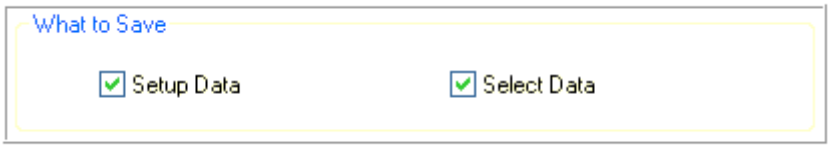


This opens the *Save State* dialog, shown in *Save State Dialog* (allegrolink).

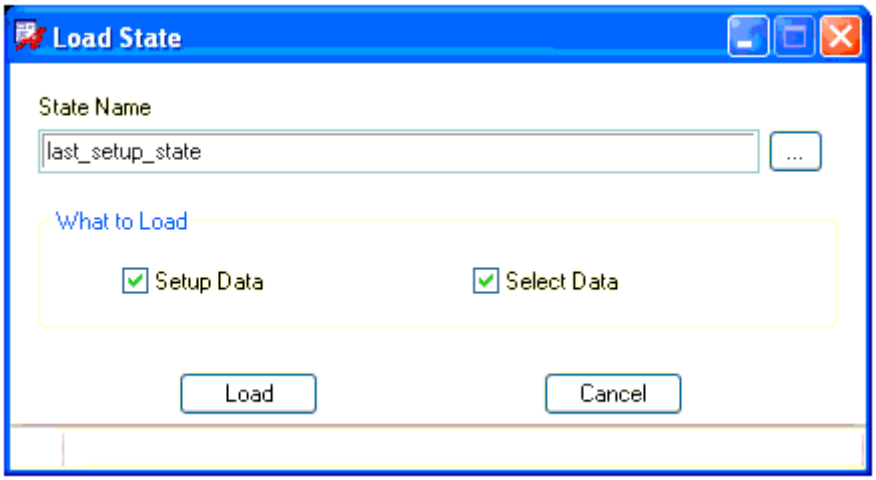


Save State Dialog

Using this dialog you can choose to save either the setup data, selected data of your designs internal state, or both by picking the appropriate check box.



This enables you to change the export settings and return to a prior state without losing setup or configuration information.
In the same way you can reload previous setup information by choosing *Export To ADS* > *Setup* > *Load State...* from the Allegro PCB Editor main menu.
This opens the Load State dialog, shown in *Load State Dialog* (allegrolink).



Load State Dialog

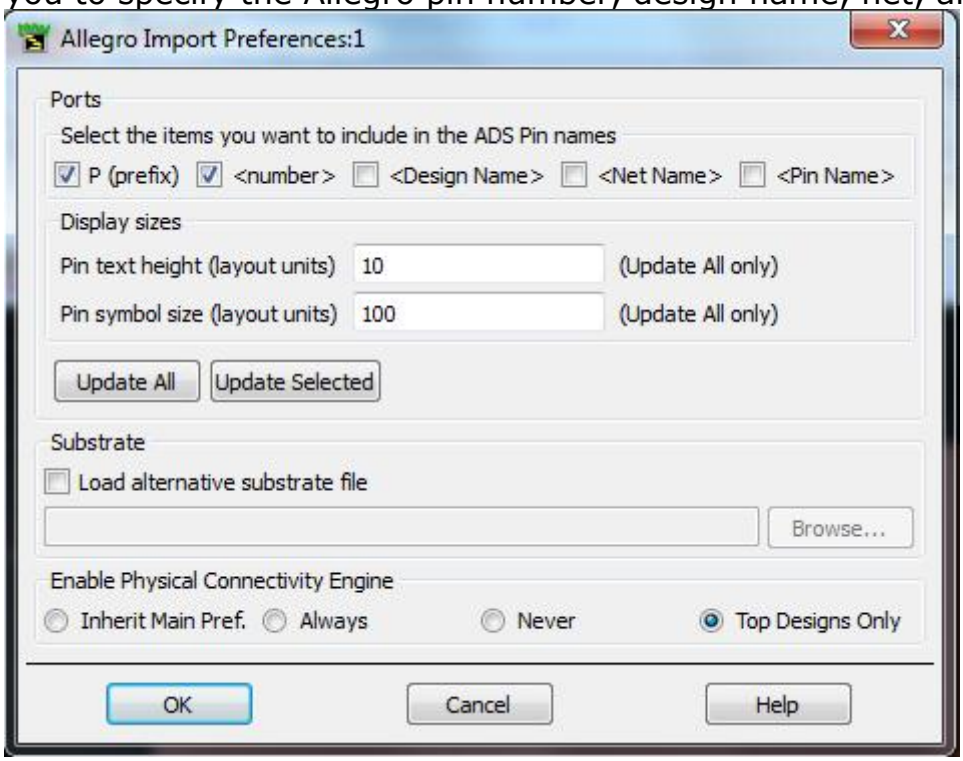
As with the *Save State* dialog, you can choose to load either the setup data, selected data of your designs internal state, or both by picking the appropriate check box.
Both the Save and Load State dialogs enable to choose the name and location of the file using a standard Windows file browser mechanism

Importing Allegro DFI Files in ADS

Before importing Allegro files, you need to import the *Import Allegro Design Kit* in your ADS workspace. This enables you to display the **Allegro Tools** menu in your ADS layout windows. For more information, refer to *Installing Allegro DFI* (allegrolink).

To import the exported Allegro files by using *Import Allegro Design Kit*:

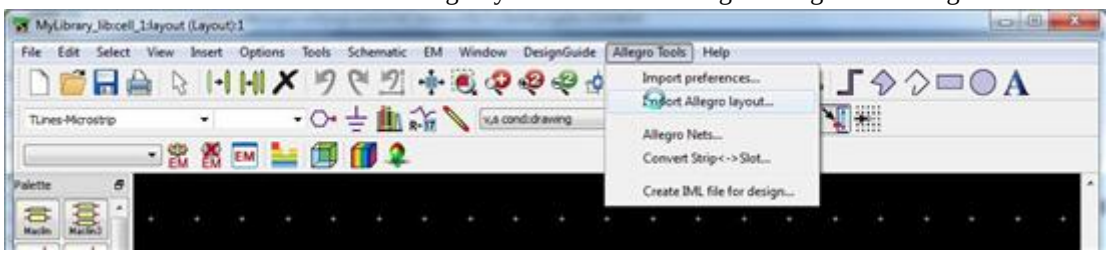
1. Select **Allegro Tools** > **Import Preferences** in the ADS layout window to configure ADS pin names. This displays the Allegro Import Preferences dialog box that enables you to specify the Allegro pin number, design name, net, and pin names.



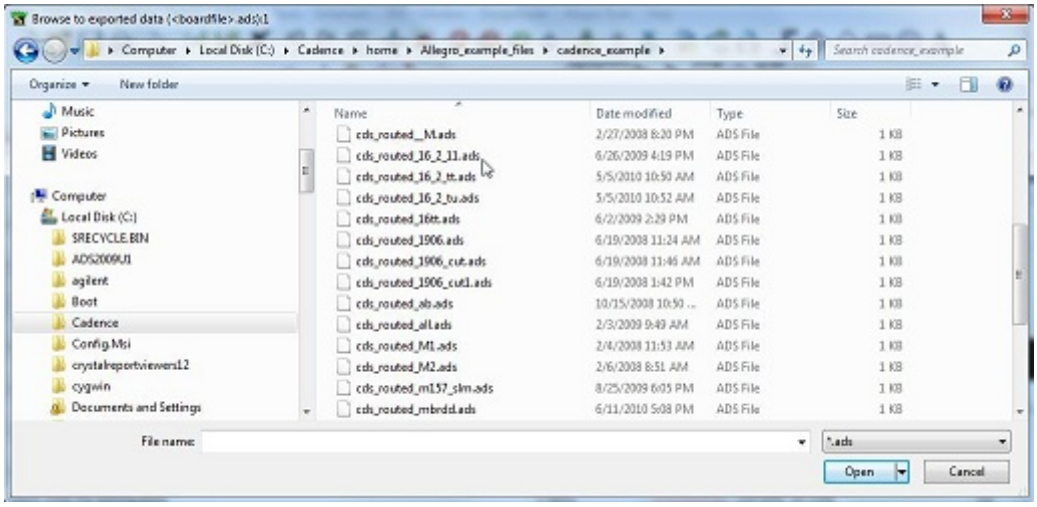
2. In the Allegro Import Preferences dialog box:
    1. Select the options that you want to include in the ADS pin name such as prefix, number, or design name.
    2. Specify the display sizes, if there is a unit mismatch between the ADS workspace and the Allegro design that will import.
    3. Override the substrate (slm) files that gets loaded during the import by specifying an alternate file in the **Substrate** section.
    4. Select if the Physical Connectivity Engine (PCE) needs to be enabled on the created designs during the import process. Enabling the PCE on large designs can be very time consuming.

    > **Note**
    > After importing file, you can use the Import Preference dialog box to update the pin names, pin text, and arrow sizes for either all pins or a selected subset.

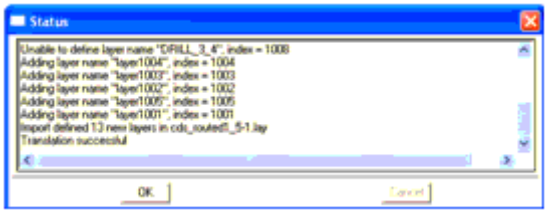
3. Open a new ADS layout window by selecting **Allegro Tools** > **Import Allegro Layout**.



4. Browse the export directory and select *<boardfile>.ads file that you want to import*.



5. Click **OK** to initiate the import process.
6. While running the import process, the following layout designs are displayed again:
   - The original layout cell used to start the import process.
   - A layout cell view containing the finished <design library name>_lib:<cell with flat design name>:layout.
   - When a hierarchical *.adfi* representation is available, a top level layout design <design library name>_lib:<cell with design name _adfi>:layout.
   - An ADS import status window, as shown below might also appear. Click **OK**.

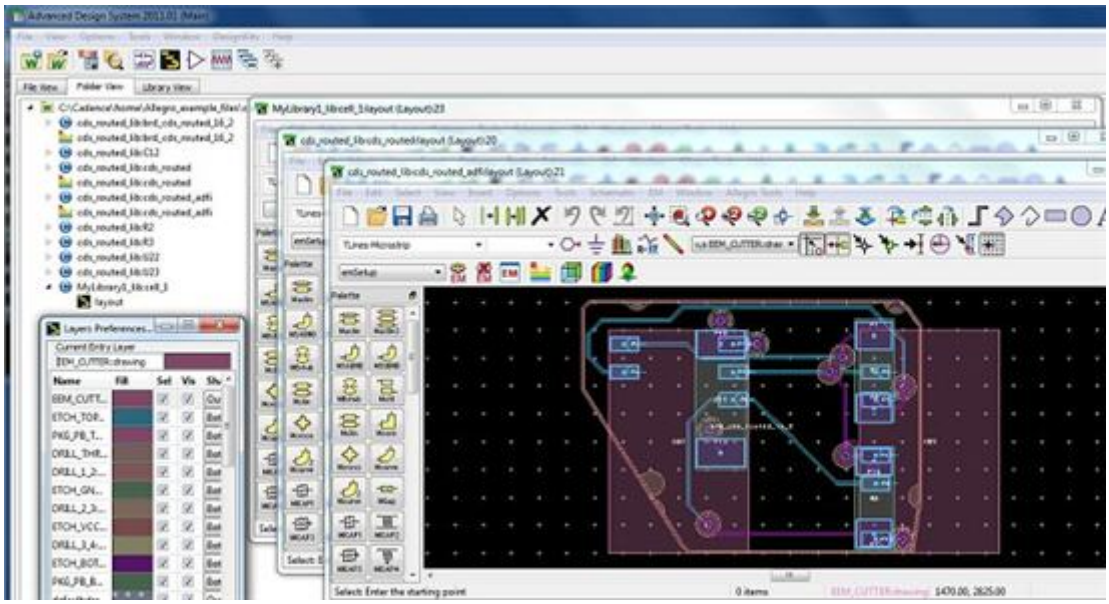


   - If some problem occurs a warning or info message might be shown to indicate the translation problem.

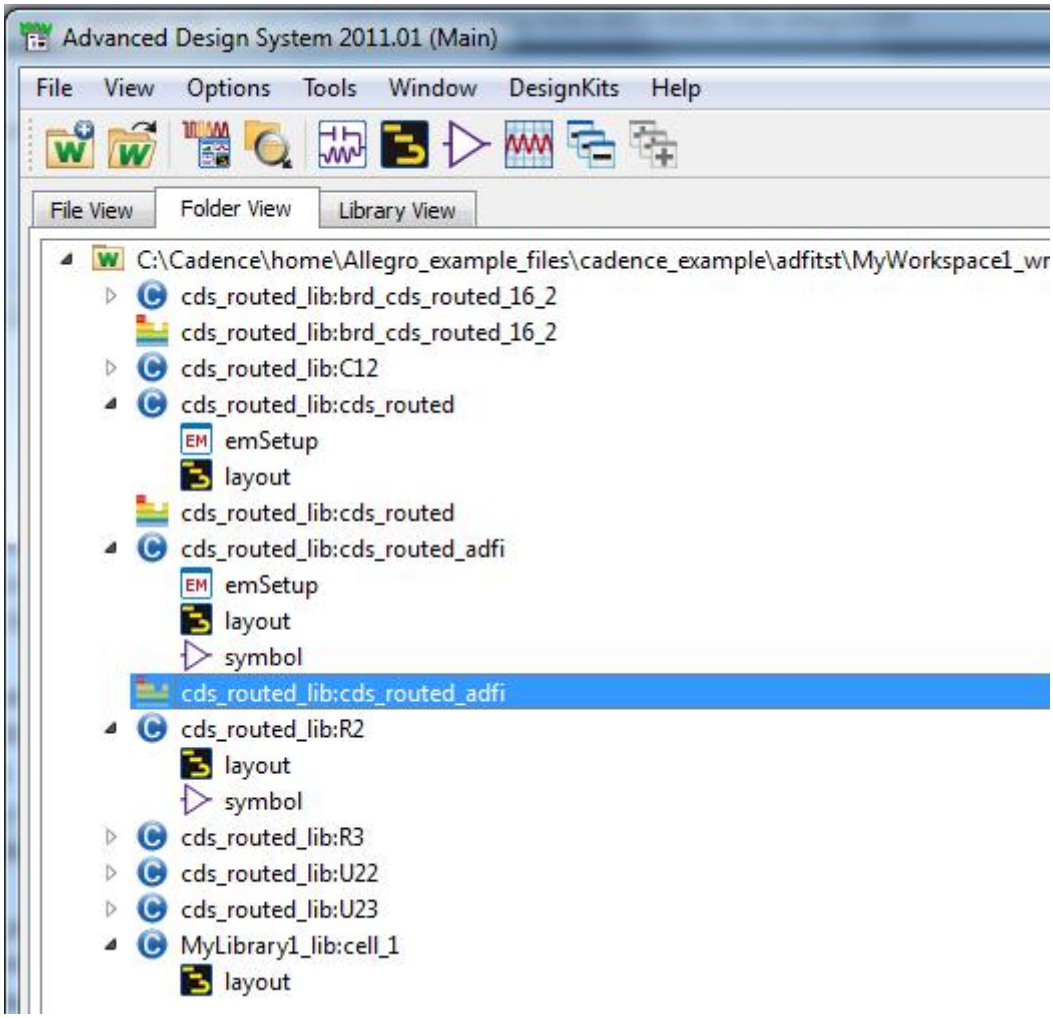
The import process creates a new library in your ADS workspace with the name of the selected export design name. If the library name already exists the name followed by a sequence number is used to avoid overwrite conflicts with existing designs in earlier libraries. The new library created to support the import process will have the same resolution and unit setup for layout as specified by the Allegro DFI export files produced by Allegro. This avoids conversion issues during the import of the geometry files.

The text and pin arrow sizes can still be wrong if the workspace initial setup was done using other layout resolution and unit settings. The actual import process loads the 4 or 5 import Allegro DFI files produced by the export: the EGS layout, hierarchy description, the

Momentum substrate definitions, the exported ports and associated pins. During the import it will also update the layer binding for the Layout Connectivity engine based on the given Momentum substrate definition in this library.



Besides a layout view, the imported cells for the design also contain a EM Setup view with an updated port and substrate description to allow EM simulations.



Inside an EM setup, the pins are combined into ports. These ports use a combination of plus and minus pins as created during the export process. The port numbering is maintained from the Allegro side. The pin numbers above the highest exported port number are the *Negative Reference* pins in the EM setup. The ports have no calibration by default.

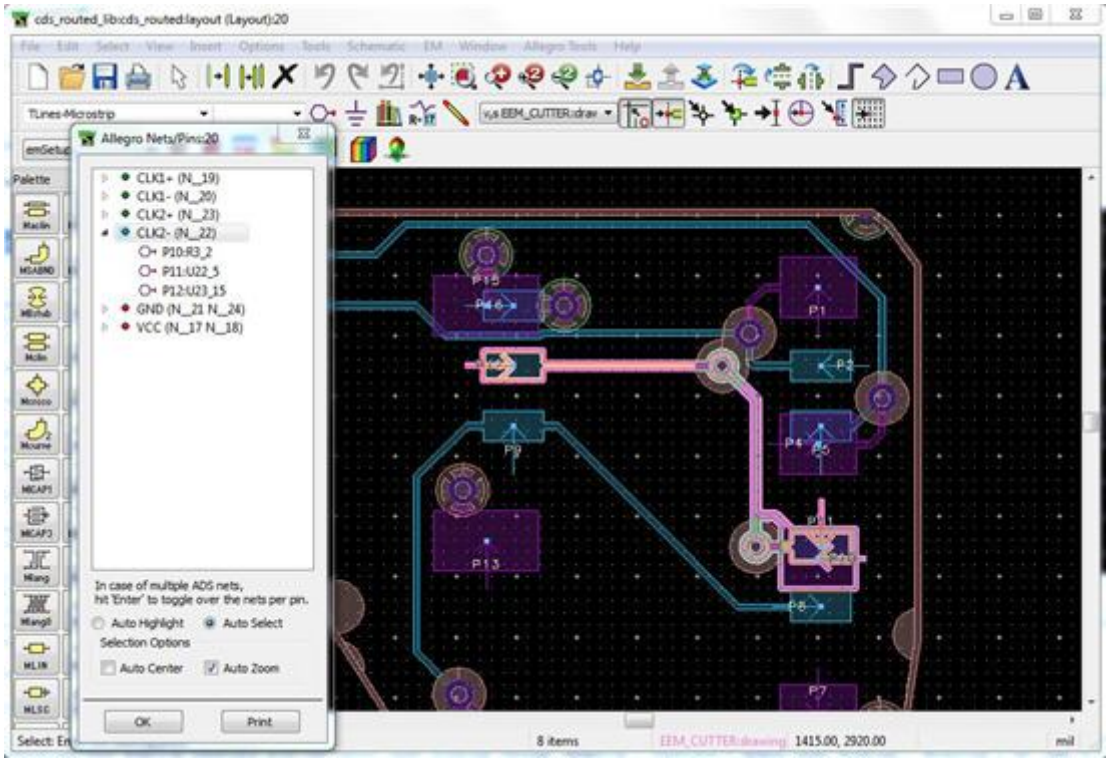
A minimal update, such as the addition a frequency plan to the EM setup, should be sufficient to start a Momentum or 3D EM simulations for these designs with setup.

Inside the created layout views, several properties are attached to each imported ADS pin:

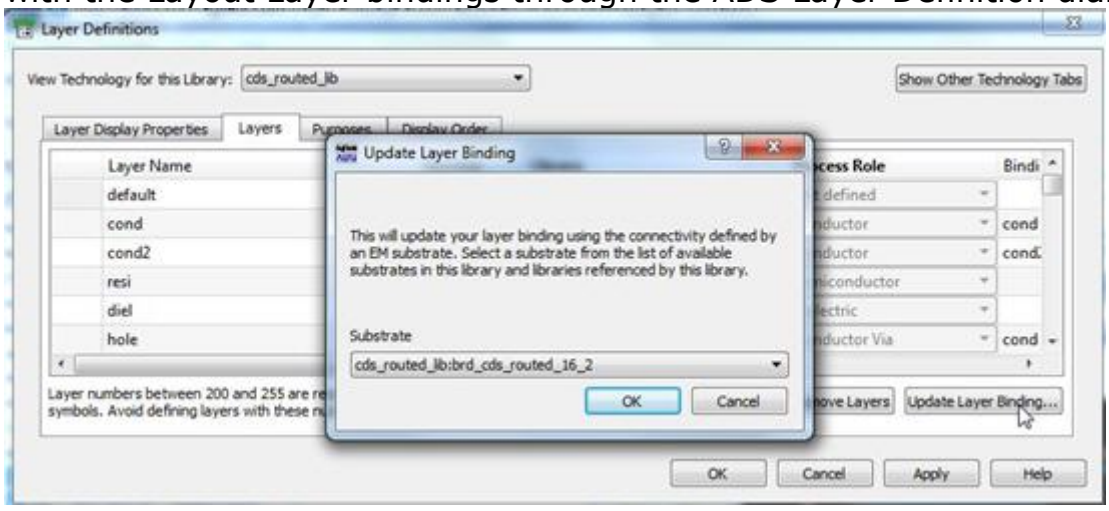
- **From Tool**: This property specifies the tool from which the imported data in ADS was exported. The value of this property is allegro when the data was produced by Allegro and this is the <tool> prefix for the next properties.
- **<tool>_pin**: This property specifies the <component instance>.<package pin number>, e.g. allegro_pin = U16_28.
- **<tool>_net**: This property specifies the name of the net, e.g allegro_net = clk2+.
- **<tool>_design**: This property specifies the name of the design in the tools, e.g. allegro_design = cds_routed.
- **<tool>_portNumber**: This property specifies the Momentum port number assigned by the ADFI export tool, for example, allegro_portNumber = 4.
- **<tool>_portName**: This property specifies the Momentum port name assigned by

the ADFI export tool, for example, allegro_portName = clk2+_U16_28.

This information is used by the Allegro Nets tools to show the imported nets, pins, and port configurations. You can access this information by **Allegro Tools > Allegro Nets**.



The Allegro Nets/Ports utility allows you to highlight and select specific net and pin/port configurations in the imported ADS design. It uses the ADS Layout Connectivity engine to show and verify net connectivity once you have synchronized your Momentum substrate with the Layout Layer bindings through the ADS Layer Definition dialog box.



Selecting a net name will immediately highlight the net through the board structure (at least if there is no complex layout hierarchy). If you select one or more pins in a the Allegro Net/Pins utility it will immediately select either the pin(s). The enter button allows you go through the different parts of the unconnected net parts.

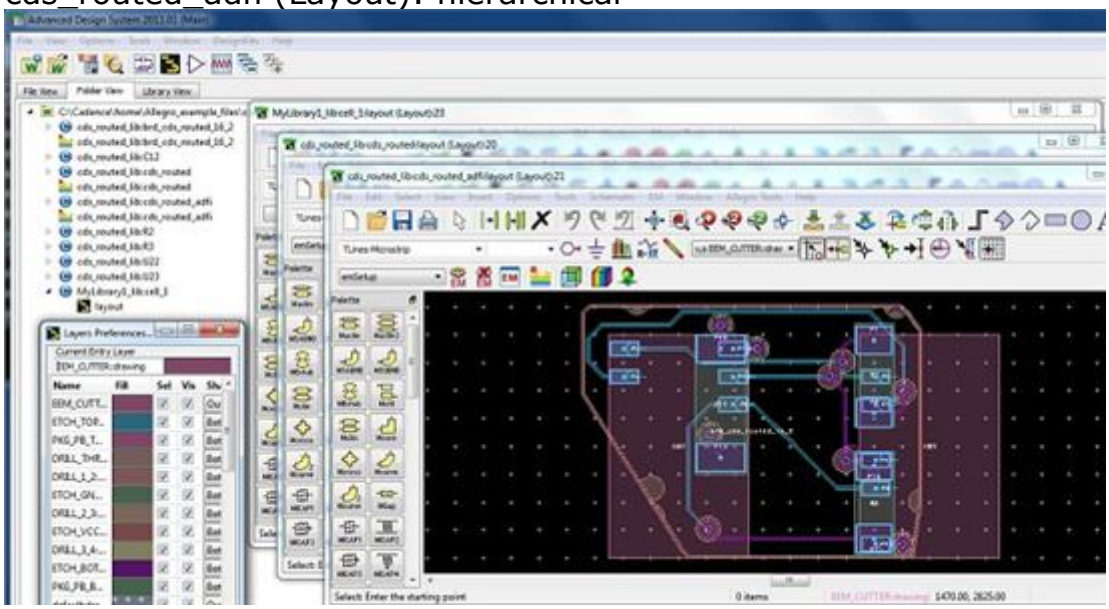
Importing with Hierarchy

An import kit automatically imports a *<exportname>.adfi* file when it exists. This file consists of the following:

- Main top level design <exportname>_adfi
- Board, Package, SiP design(brd_|sip_|apd_)<allegro design>
- Instances of exported components are sub designs using the reference designator from Allegro platform
  - No reuse with instantiating.
  - Pins lists can be incomplete.
- Area pins on all component pins.
- Parameters/properties transferred where available.
- Black box symbol or R/L/C/D symbol attached when recognized.
- Sync from Layout to Schematic possible.

In this case, two versions of the design are imported. If you try this with the cds_routed design, you will see a library cds_routed_lib with the design cells:

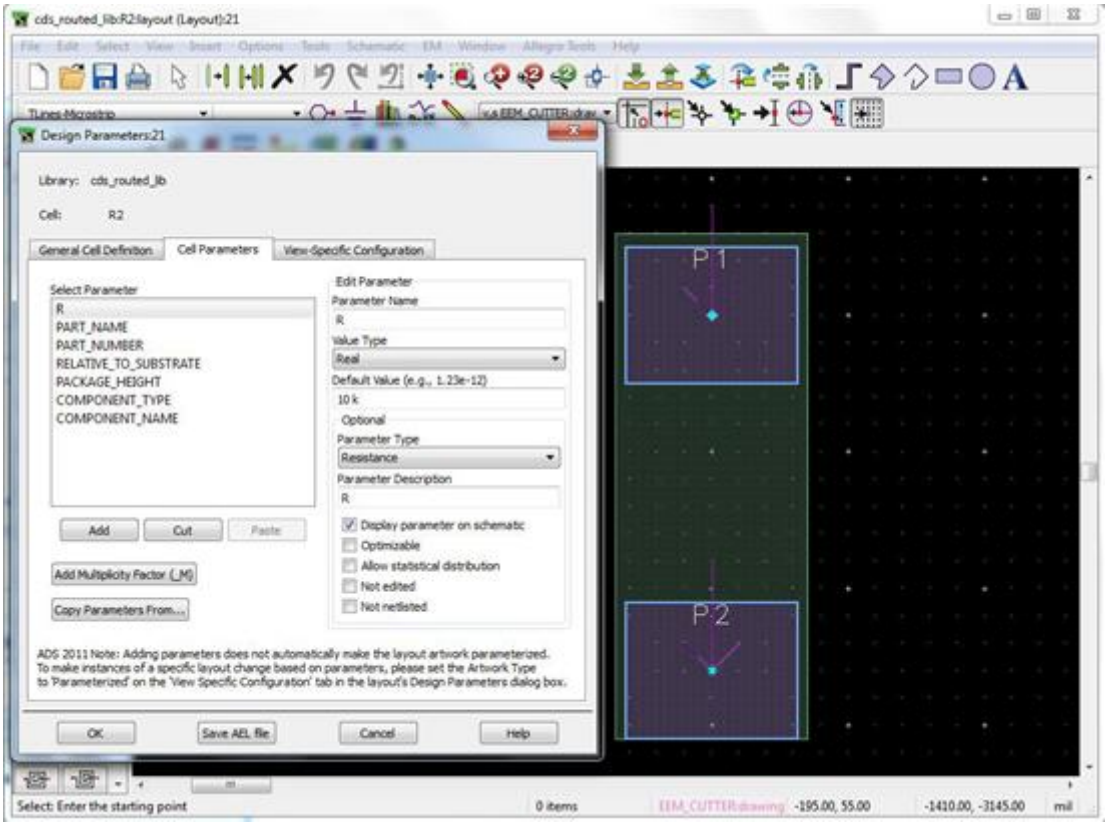
- cds_routed (Layout): flat
- cds_routed_adfi (Layout): hierarchical



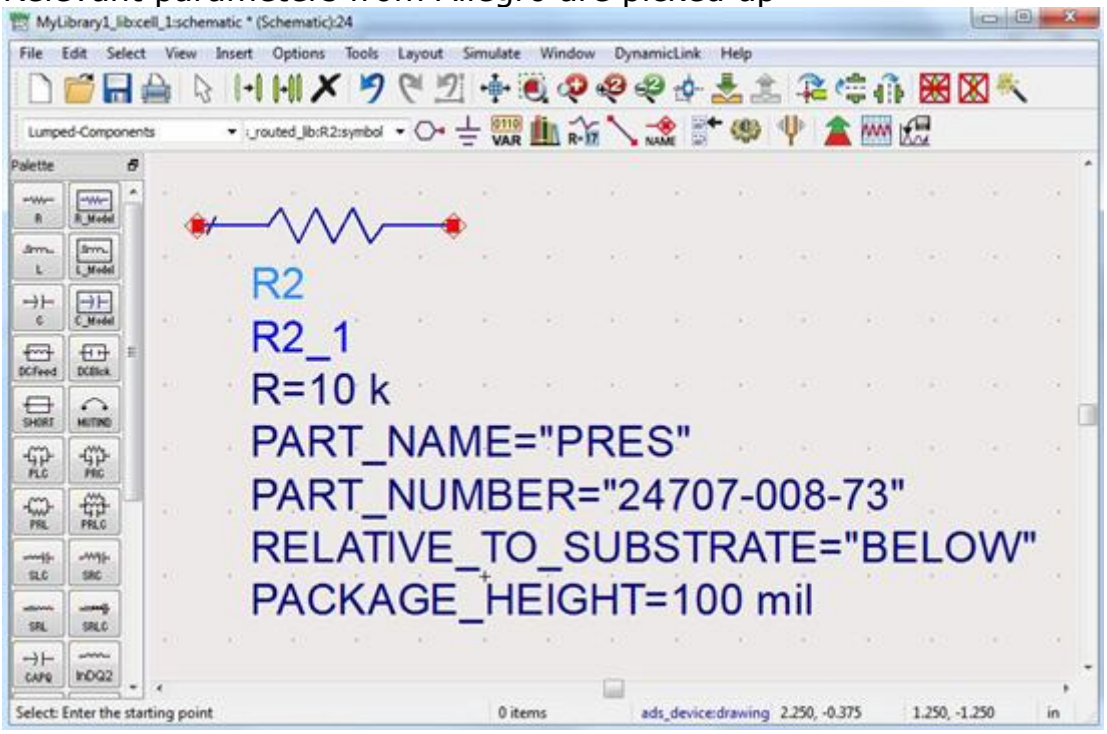
Examples Using Hierarchical Import

Example 1: Discrete SMD Capacitor

- Component designs have fixed layout as extracted from Allegro with the exported pins as area pins



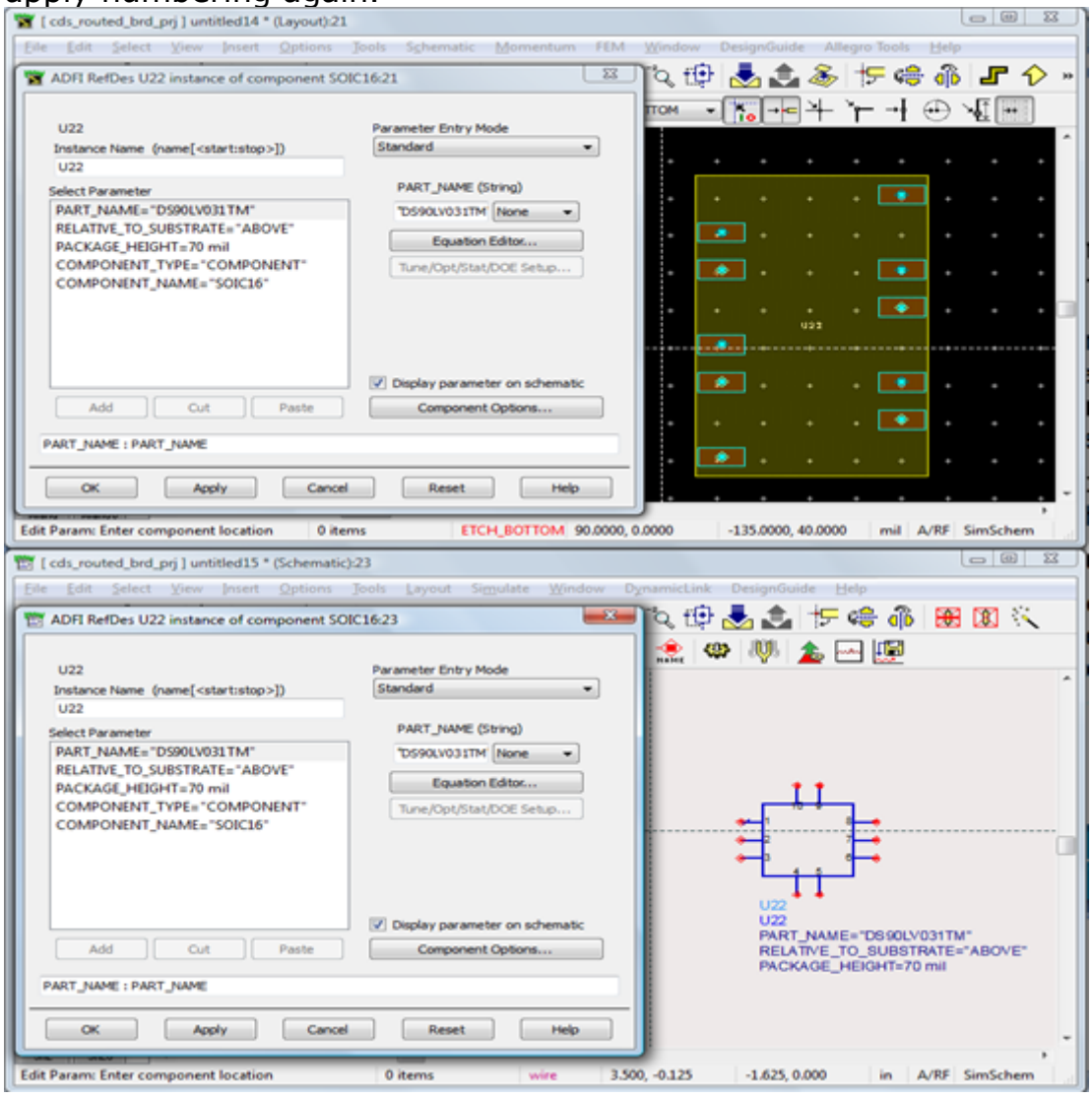
- Default symbol attached if know
- Relevant parameters from Allegro are picked up



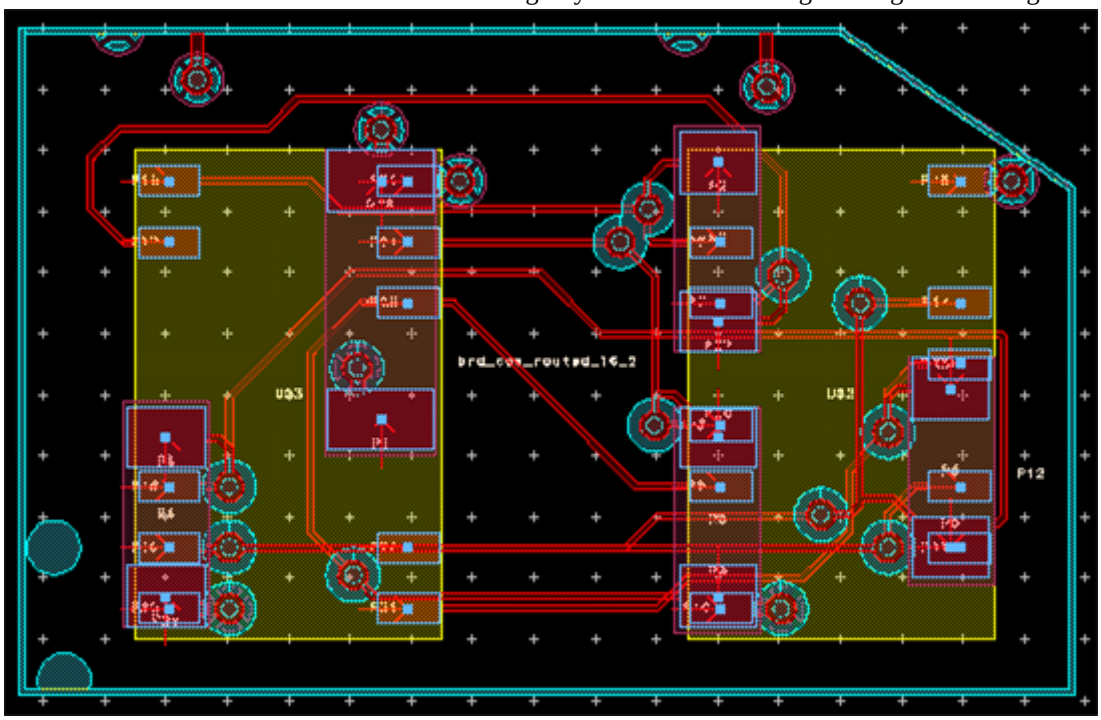
Example 2: Multi-pin IC
Pins limited to selected subset (name/number issues possible)

- Allegro uses names and the number is not relevant. For ADS it is generated from name in export/import using heuristic but not always correct.
- Footprint is correct pin name property is matches with Allegro.

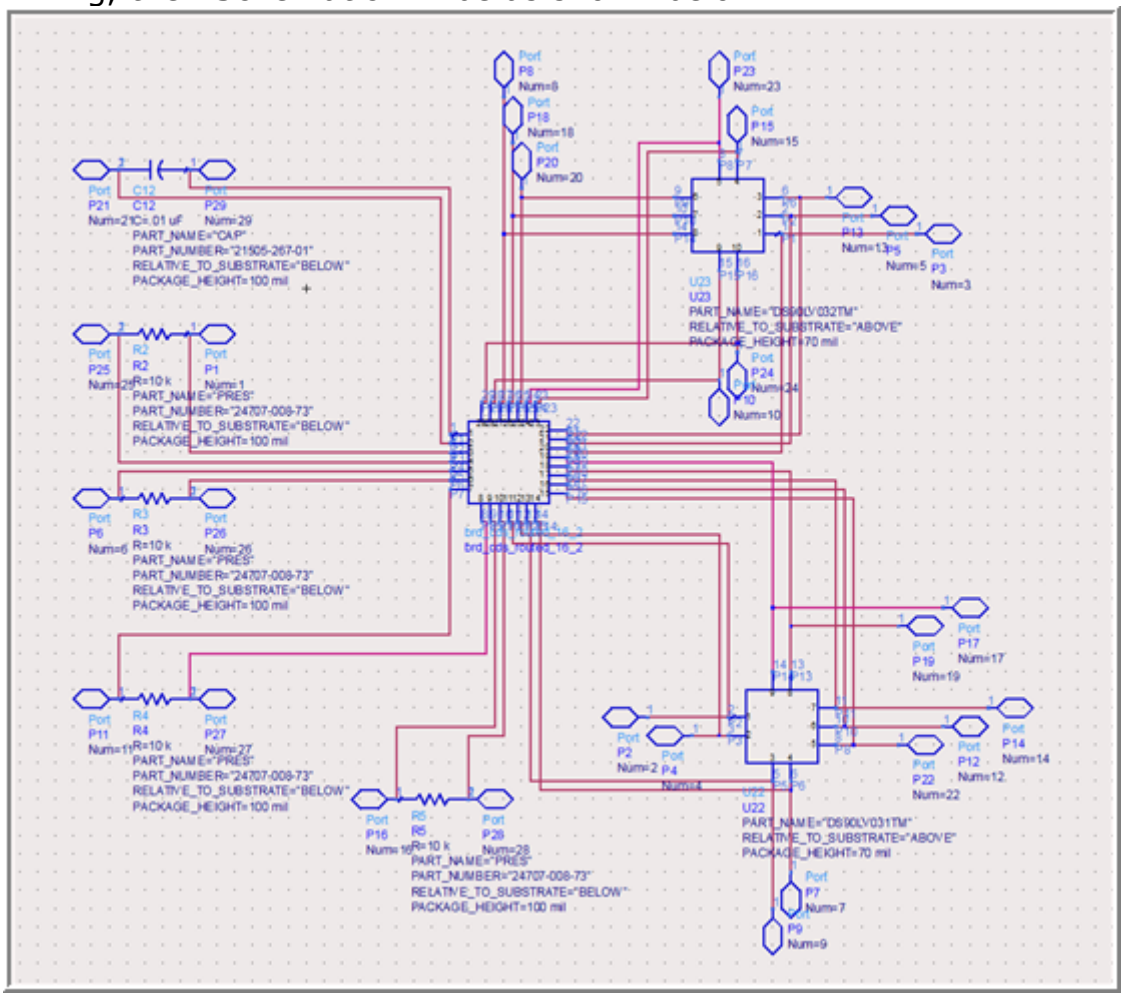
- Check the pin numbering.
- For look-alike symbols generating Momentum or FEM layouts, verify if you need to apply numbering again.



Example design sync with manual wiring for cds_routed



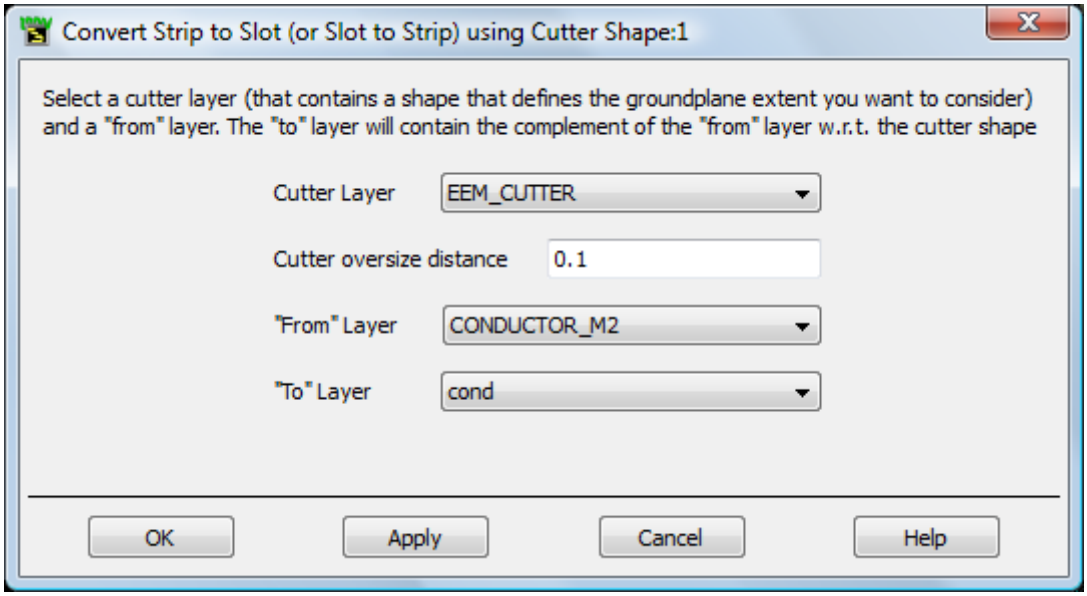
ADS Design Synchronization:Place Components from Layout to Schematic with Manual Wiring, then Schematic will be as shown below



Note
You should check the symbol pin numbering, if a wiring error occurs. Symbol generation can get confused if components are partially exported.

Converting Layers between Strip and Slot Representation

You can convert between Strip and Slot Layer representations by using the **Allegro Tools > Convert Strip <-> Slot** utility. This opens the following dialog box that enables you to convert strip and slot layer representation:



You can select the following types of layers:

- Cutter layer is used as boundary for Boolean operations on a layer having *cutter* in the name. The utility searches layers that have been exported as cutter layers by the Allegro Export tool (layer name ending with *cutter*). The polygonal shape on this EEM_CUTTER:drawing layer is used as area of interest for the conversion.
- The oversize field allows the user to extend the cutter shape with a specific distance so that complicated shapes as result of the required Boolean operations are avoided. You can use this setting to remove or avoid slivery objects from the conversion.
- The *From Layers* drop-down list provides the existing strip or slot layers in the design.
- The *To Layer* drop-down list provides the existing layer from the accessible layer list from this library.
- The substrate settings are automatically adjusted between Strip and Slot setting. Make sure the substrate editing is not open on this substrate during the execution.

Note
Slot Layers are ideal infinite layers and the conversion looses material properties.

Creating a Cadence IML file from Simulation Results

To create a Cadence Interconnect Model file in IML format from a simulation result of the current design design go to **Allegro Tools > Create IML file for design**.
This utility transfers simulation results back to the Cadence analysis tools through an IML file.

- This is a *Touchstone* file format with special header.
- Needs a dataset file of S-parameter simulation with matching pin setup of the current design. Group port constructions are not supported.
- <datasetname>.iml file created in data directory of a workspace.
- Can be used by Cadence PI/SI option in Allegro 16.3.

```
! NETSLIST B_DVSS , B_DVCCQ
! Port 1 = D1_Vddq B_DVCCQ
! Port 2 = D1_Vddq1 B_DVCCQ
! Port 3 = D1_Vddq2 B_DVCCQ
! Port 4 = D1_Vddq3 B_DVCCQ
! Port 5 = D1_Vddq4 B_DVCCQ
! Port 6 = D1_Vddq5 B_DVCCQ
! Port 7 = D1_Vddq6 B_DVCCQ
! Port 8 = D1_Vddq7 B_DVCCQ
! Port 9 = D1_Vddq8 B_DVCCQ
! Port 10 = D1_Vddq9 B_DVCCQ
! Port 11 = D1_Vss B_DVSS
! Port 12 = D1_Vss1 B_DVSS
! Port 13 = D1_Vss2 B_DVSS
! Port 14 = D1_Vss3 B_DVSS
! Port 15 = D1_Vss4 B_DVSS
! Port 16 = D1_Vss5 B_DVSS
! Port 17 = D1_Vss6 B_DVSS
! Port 18 = D1_Vss7 B_DVSS
! Port 19 = D1_Vss8 B_DVSS
! Port 20 = D1_Vssq B_DVSS
! Port 21 = D1_Vssq1 B_DVSS
! Port 22 = D1_Vssq2 B_DVSS
! Port 23 = D1_Vssq3 B_DVSS
! Port 24 = D1_Vssq4 B_DVSS
! Port 25 = D1_Vssq5 B_DVSS
! Port 26 = D1_Vssq6 B_DVSS
! Port 27 = D1_Vssq7 B_DVSS
! Port 28 = D1_Vssq8 B_DVSS
! Port 29 = D1_Vssq9 B_DVSS
! Port 30 = Package_M4 B_DVCCQ
! Port 31 = Package_M6 B_DVCCQ
! Port 32 = Package_M7 B_DVCCQ
! Port 33 = Package_M9 B_DVCCQ
! Port 34 = Package_N3 B_DVSS
! Port 35 = Package_N5 B_DVSS
! Port 36 = Package_N7 B_DVSS
! Port 37 = Package_N9 B_DVSS
!
! Created Tue Nov 17 14:58:22 2009
# hz S ri R 50
! 37 Port Network Data from data block
! freq  reS1_1  imS1_1  reS1_2  imS1_2  reS1_3  imS1_3  reS1_4  imS1_4
!       reS1_5  imS1_5  reS1_6  imS1_6  reS1_7  imS1_7  reS1_8  imS1_8
!       reS1_9  imS1_9  reS1_10  imS1_10  reS1_11  imS1_11  reS1_12  imS1_12
!       reS1_13  imS1_13  reS1_14  imS1_14  reS1_15  imS1_15  reS1_16  imS1_16
!       reS1_17  imS1_17  reS1_18  imS1_18  reS1_19  imS1_19  reS1_20  imS1_20
!       reS1_21  imS1_21  reS1_22  imS1_22  reS1_23  imS1_23  reS1_24  imS1_24
!       reS1_25  imS1_25  reS1_26  imS1_26  reS1_27  imS1_27  reS1_28  imS1_28
```

Allegro Tools Hints

Importing Files in ADS Hints

When creating the ADS workspace, make sure that you choose workspace technology files with a length unit that corresponds with the design units in Allegro. ADS Import Design Kit can be found under *$HPEESOF_DIR/ial/import_kit*. Always use the most recent version. When loaded, the Allegro Tools menu in a Layout Window will be available.

- ADS 2011.01 ships with version 3.0.
- Check www.agilent.com/find/eesof-knowledgecenter and search for "ADFI" to find out about updates. Here mentioned features are based on design kit version 2.6 and later.

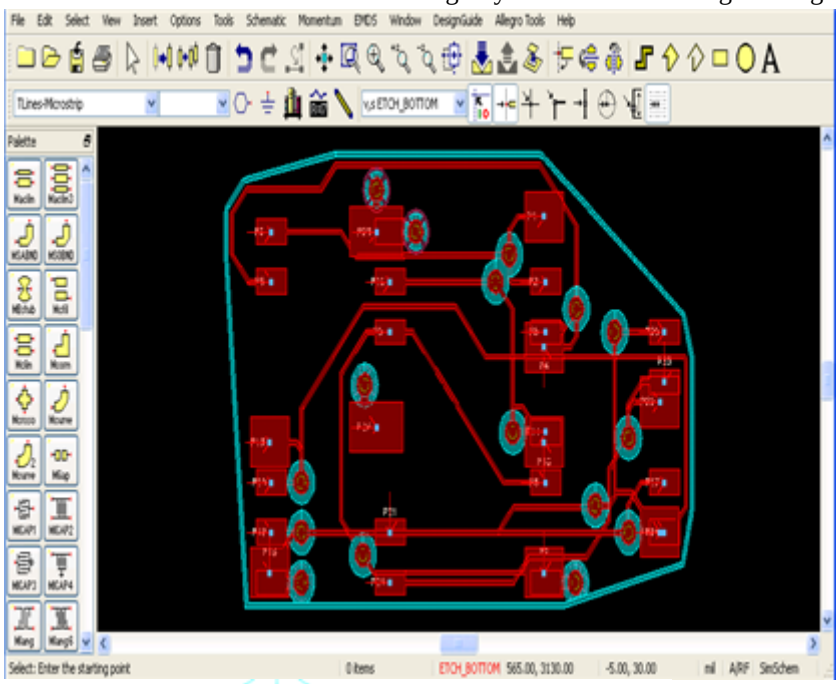
Using the Layer View Utilities

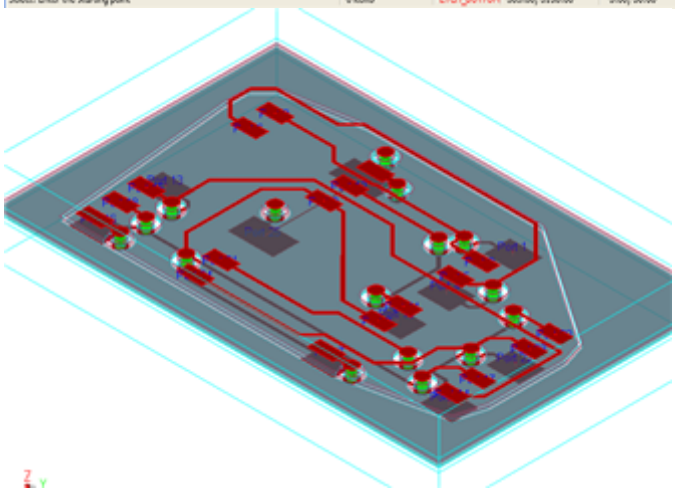
You can access various options related to Layers as **View > Layer View > <Type>**. The layer views are available:

- By Name
- Show Substrate Layers
- Hide Non-Substrate Layers
- Top
- Up
- Down
- Bottom
- Show Current and Previous Layers
  These menus toggle layer visibility based on their location in the substrate.You can assign hotkeys too for this.

Importing Allegro Layout and Slots

- Layers and substrate get loaded. Layer binding is updated, cutter shape is available on EEM_CUTTER layer with purpose drawing.
- In case of slot layers, auxiliary slot shapes are passed that will support the conversion of split ground planes from slot to strip.





Splitting Ground Planes on Negative Layers

Allegro negative shield layers keep their cutouts in the plane as slots, as shown in the following figure:

	Subclass Name	Type	Material	Thickness (MIL)	Conductivity (mho/cm)	Dielectric Constant	Loss Tangent	Negative Artwork	Shield	Width (MIL)
1		SURFACE	AIR			1	0			
2	TOP	CONDUCTOR	COPPER	1.2	595900	1	0	☐		5.00
3		DIELECTRIC	FR-4	8	0	4.5	0.035			
4	GND	PLANE	COPPER	1.2	595900	4.5	0.035	☒	☒	
5		DIELECTRIC	FR-4	8	0	4.5	0.035			
6	VCC	PLANE	COPPER	1.2	595900	4.5	0.035	☒	☒	
7		DIELECTRIC	FR-4	8	0	4.5	0.035			
8	BOTTOM	CONDUCTOR	COPPER	1.2	595900	1	0	☐		5.00
9		SURFACE	AIR			1	0			

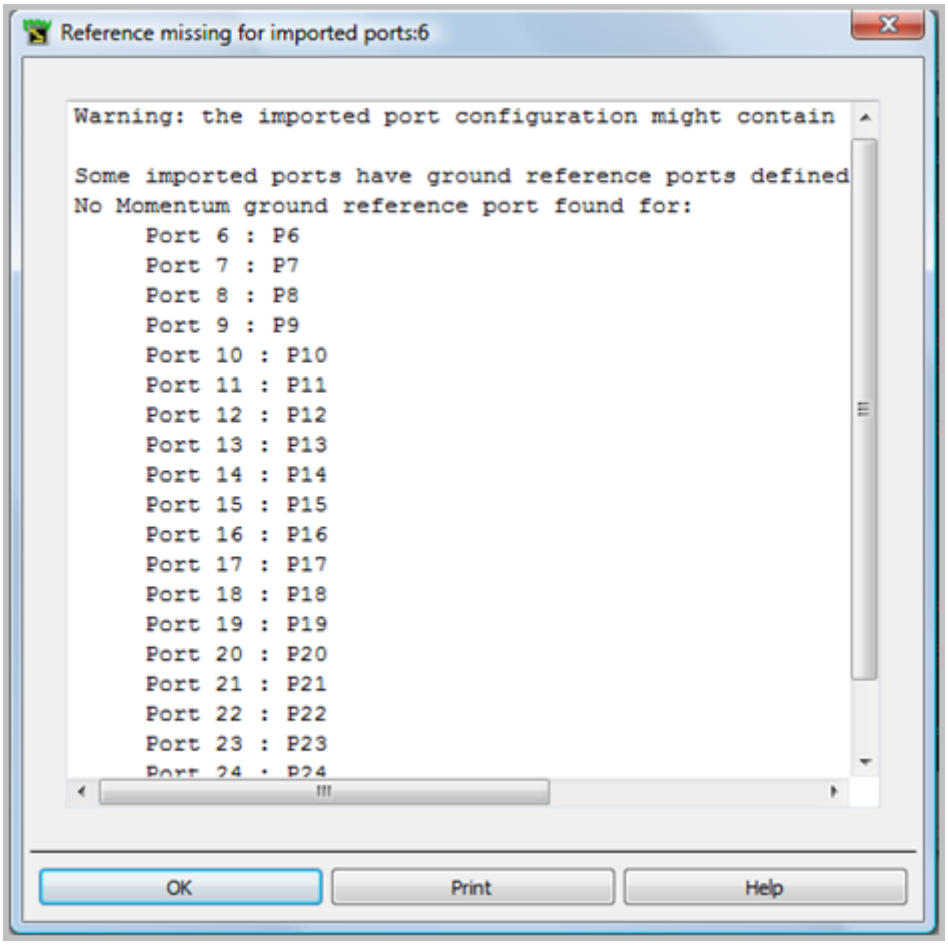
can export as slot layer

A slot contour is added to create isolated areas.

Note
An isolated gnd shape cannot be electrically large for an accurate Momentum simulation. The whole shape will be considered as one cell.

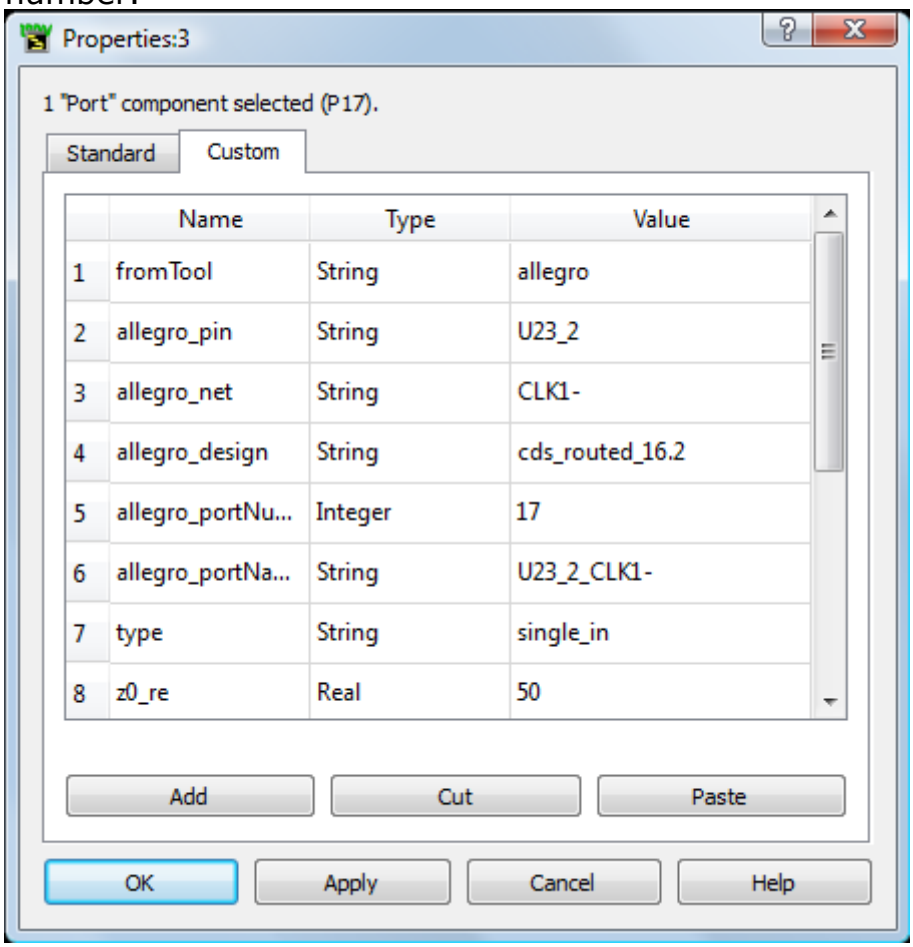
Ports Hints

In some cases, if S-parameter ports have a negative reference pin, a warning is displayed, as shown below:



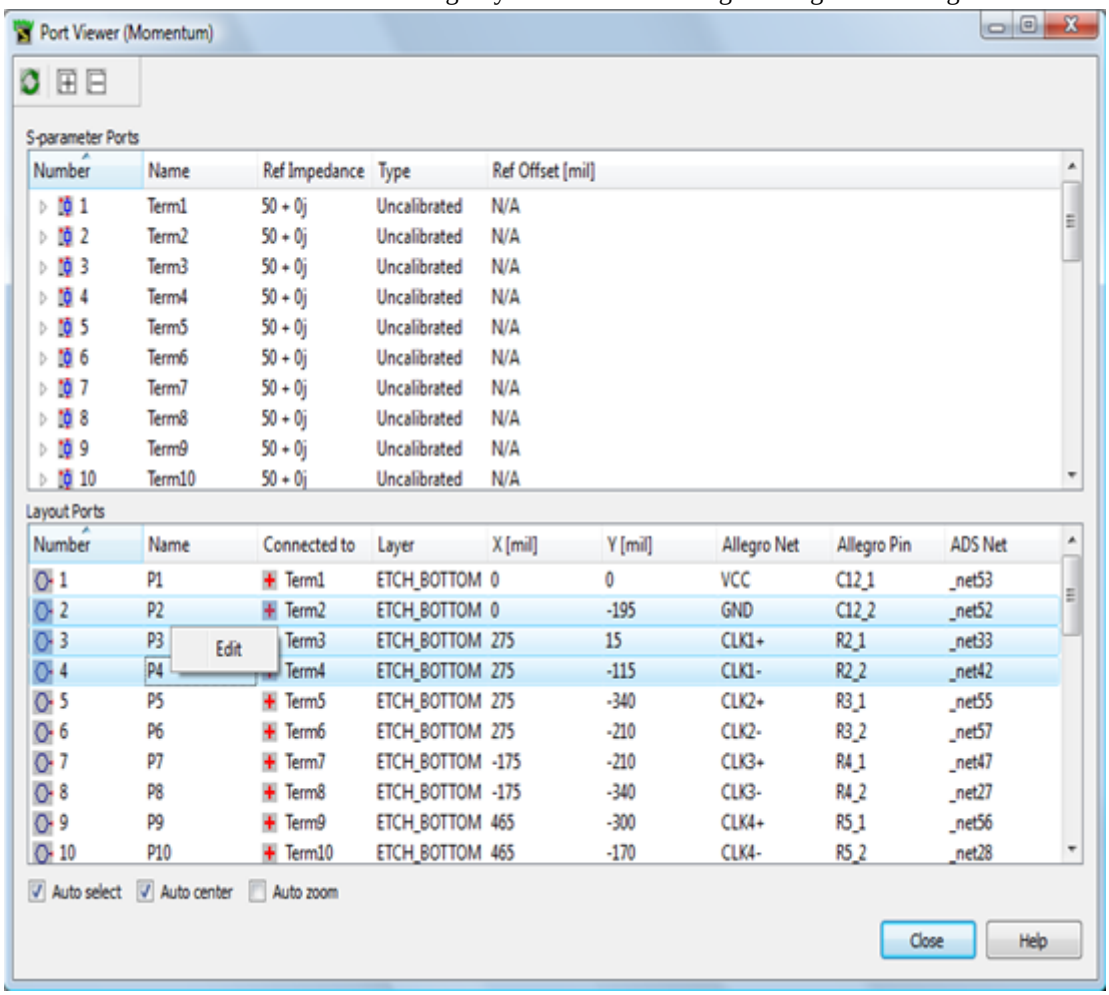
Relevant Allegro information is attached as properties to a port and you can access this

information from this Properties dialog box. For example, net name, pin name, and number.



The EM Port Editor helps you to manipulate pins and ports:

- It provides an overview of S-parameter ports and Layout ports.
- Easy to sort and filter operations on the columns.
- Provides select and multi-select in the layout.
- Shows Allegro/SiP/APD nets/pin properties, if available.



You can move layout ports from one layer to another. After selecting ports from (sorted) list in the port editor. perform the following steps:

- In ADS 2009 Update 1:

1. Select **Edit** > **Component** >**Group Edit Parameter Value** from the layout window. In the **Group Edit Parameter** dialog box, specify a value for the following parameters:
    1.
        - Parameter Name is set to **layer**.
        - Parameter Value includes a layer name in double quotes.
        - Click **Apply**.

- In ADS 2011.01, select **Edit > Move > Move to Layer** from the layout window.

Allegro Net View Hints

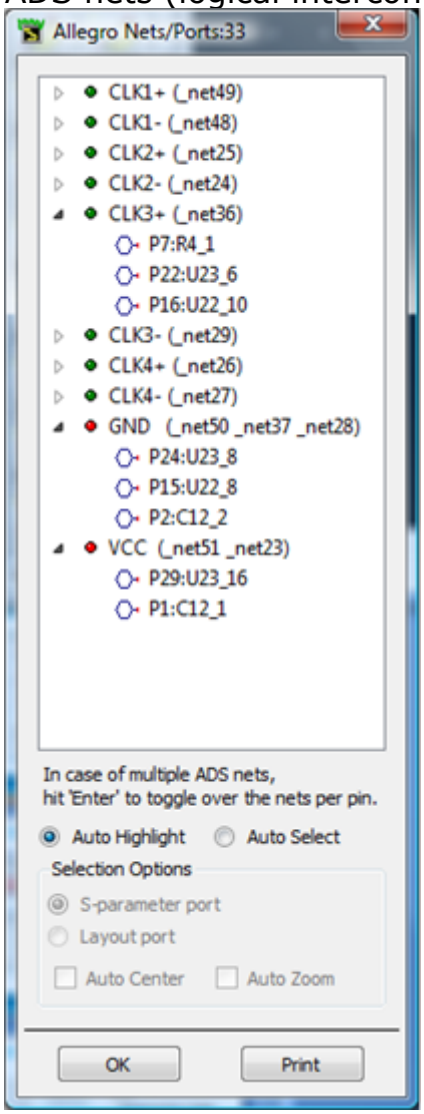
Allegro Nets/Ports dialog box lists the Allegro nets found from port properties.

Note
PCE must be enabled. Select the *Using PCE for Drawing* option in Main Preference dialog box (Main Window > Preferences).

ADS net name(s) is between „(„ ?)?. When there is more than one, bullet is displayed in color red. Possible causes:

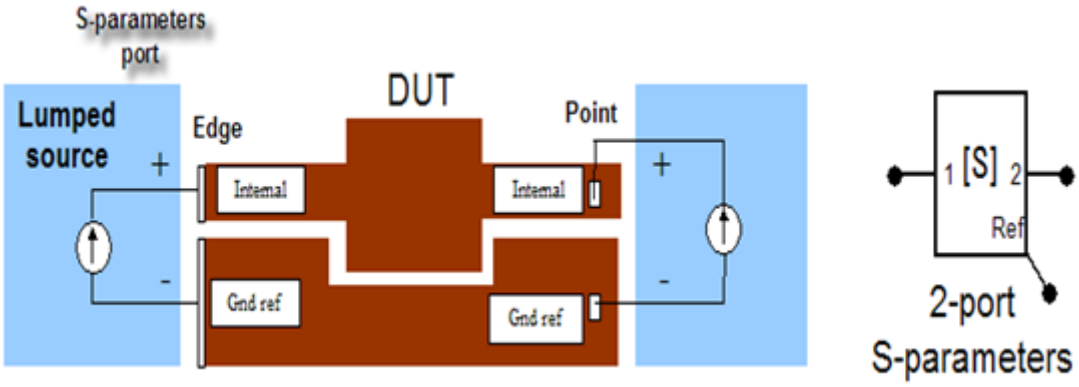
- ADS design is hierarchical(*_adfi cell). An ADS net (aka logical interconnect) does not go into hierarchy. Hierarchy splits a net.
- PCE does not recognize connectivity:
  - Between shapes that are in touch with an infinite ground plane (slot layer).
  - Through wire bonds
- Allegro net is having cut.

In such case, Press **Enter** to toggle over the Allegro pins. The ADS net connected of the pin will be highlighted/selected.When Auto Highlight is enabled, the physical interconnect which goes through hierarchy will be highlighted.When Auto Select is enabled, selected ADS nets (logical interconnect) with the connected component instances are selected.



Selection Option - You have two port defined here ,one is S-parameter port and other is Layout port.

- S-parameter port
  - All Allegro pins (= ADS Layout Ports) connected to the S-parameter port are selected.
- Layout port
  - Allegro pin (= ADS Layout Port) is selected.


S-parameters
port
Lumped
source
+
-
Edge
Internal
Gnd ref
DUT
Point
Internal
Gnd ref
+
-
1 [S] 2
Ref
2-port
S-parameters

Installing Allegro DFI

Allegro DFI consists of an add-on utility integrated into the Cadence Allegro Layout editor as the *Export To ADS* menu and the design kit *IMPORT_ALLEGRO* which must be enabled in ADS to provide the *Allegro Tools* menu.

The files needed to provide the *Export To ADS* menu in Allegro and the *Allegro Tools* menu in ADS can be found in your local ADS installation directory under either:

%HPEESOF_DIR%/ial or *$HPEESOF_DIR/ial*

Installing Allegro DFI inside Allegro

Allegro DFI ships with ADS, but the Cadence Allegro part is standalone. For standalone use inside Allegro, you can place the *ial* directory anywhere in your file system.

The configuration script *$HPEESOF_DIR/ial/scripts/eemLocalConfig.scr* requires AXL-SKILL command line capabilities that are only available in tiers of Allegro with the Performance option L and higher. Design L versions of Allegro 16.x load the ADFI tool but you need to use a manual configuration as described below.

Quick Configuration for a Single User

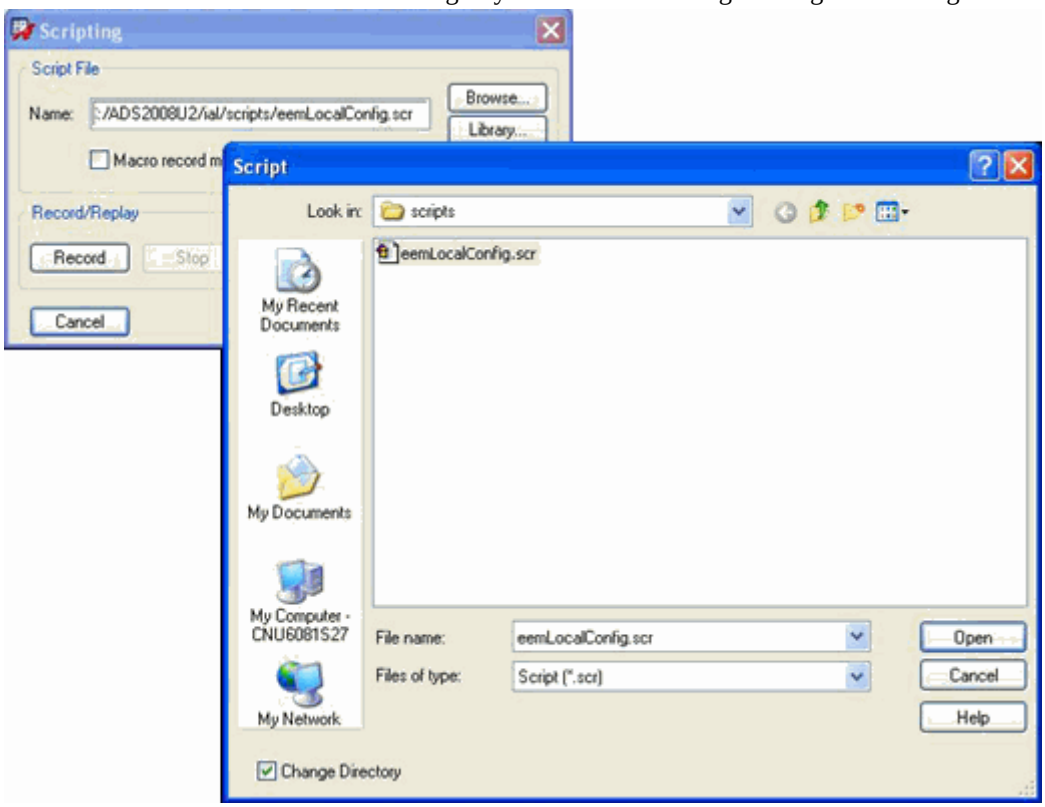
Do not try to to update from Allegro DFI versions earlier than ADS 2008 Update 2. The configuration scripts cannot deal with the old manual setups. You need to manually cleanup the Allegro Platform ilinit files or remove the entire *$HOME/pcbenv* directory if you start from such an old environment.

The *Quick Configuration For A Single User* can be the starting point for a site wide configuration. It will create a configuration file in the home environment that can be edited and moved to site wide locations.

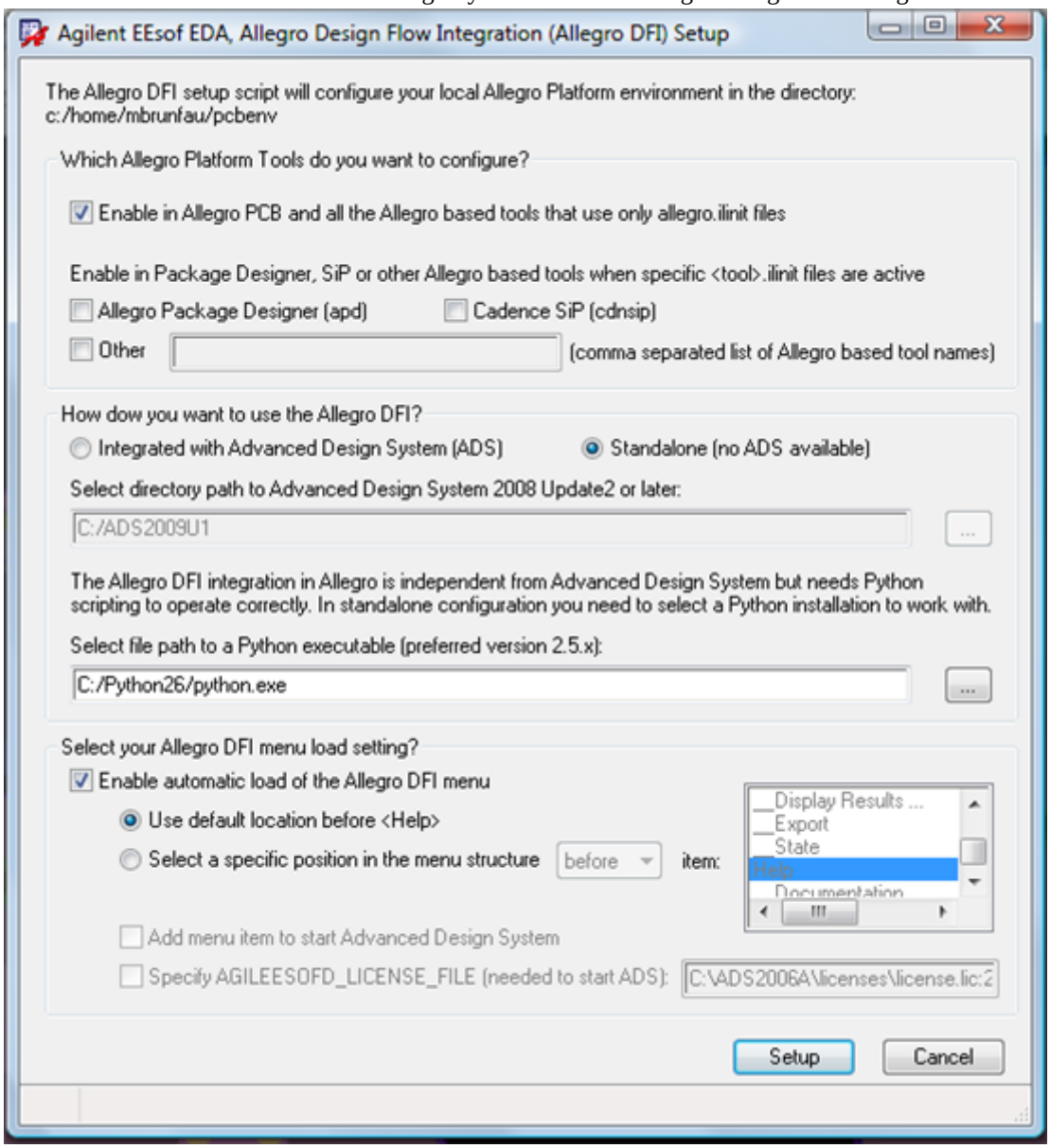
Note
"~/" translates to *%HOME%* or *%HOMEDRIVE%%HOMEPATH%* in Windows and *$HOME* in Linux

For installing Allegro DFI, perform the following steps:

1. From the Allegro menu, select **File > Script** to open the *Scripting* tool.
2. Browse to *$HPEESOF_DIR/ial/scripts* and select **eemLocalConfig.scr**.
3. Enable the **Change Directory** flag at the bottom of the **Script** file browser. **VERY IMPORTANT STEP**



4. Click **Open**.
5. Click **Replay**. This opens the single page Allegro DFI Setup dialog, as shown in the following figure:



The Allegro DFI Setup dialog box provides all the information you need to configure the Allegro DFI inside your **local** Allegro configuration in a single page. The Allegro pcbenv enviroment directory that will be modified is shown below the *The Allegro DFI setup script will configure your local Allegro Platform environment.* This is your *$HOME/pcbenv* on Linux and *%HOME%\pcbenv* or *%HOMEDRIVE%%HOMEPATH%\pcbenv* on Windows machines. You can find this information inside Allegro under **Tools > Utilities > Env Variables**.

Next, you need to select a few installation preferences in three sections:

1. *Which Allegro Platform Tools do you want to configure?*
   This section allows you to specify the Allegro Platform based tools that you want to configure.
   - *Enable in Allegro PCB and all the Allegro based tools that use only allegro.ilinit files*
     The default allegro.ilinit startup file used by Allegro PCB and all other tools if no tool specific ilinit file exist should normally always been enabled.

> **Note**
> allegro.ilinit is always used by Allegro PCB but it only remains the general skill initialisation file as long as no tool specific ilinit exist. APD, Cadence SiP, etc will ignore allegro.ilinit in favor for the tool specific ilinit file. Allegro 16.01 and later read all ilinit files found in the configuration path not just the first one found. The other ilinit files can be picked up from site specific locations and interfere with initialisation.

- *Enable in Package Designer, SiP or other Allegro based tools when specific <tool>.ilinit files are active*
  - *Allegro Package Designer (apd)* and *Cadence SiP (cdnsip)* flags for the two most common tools.
    When you have Allegro Package Designer apd.ilinit or Cadence SiP cdnsip.ilinit specific ilinit files; you can selectively enable the Allegro DFI configuration here. These options are enabled by default if one of these two files exists in your Allegro environment path. If they don't just keep the allegro.ilinit flag enabled.
  - *Other* provides a field for tool specific ilinit files for less common Allegro Platform based tools.
    You can specify a comma separated list of Allegro program specific tool names in this field. You find the program names using the skill axlVersion('programName) on the tool's command line.

2. *How dow you want to use the Allegro DFI?*
   This sections deals with how you want to use the Allegro DFI inside Allegro and allows to pick up the necessary support tools based on that selection.
   - The selection between *Integrated with Advanced Design System (ADS)* or *Standalone (no ADS available)* provides options to select the mode in which you want to use Allegro DFI.
     Based on that initial choice you need to specify one of the next two choises and only the relevant choice will available to you.
   - *Select directory path to Advanced Design System 2008 Update2 or later* selects directory path to ADS.
     You can browse for an existing HPEESOF_DIR and when a value for HPEESOF_DIR is known this will be picked up by default. This is the preferred option and you best select the latest ADS available.
   - *Select file path to a Python executable* allows you to specify the path for Python installation by selecting the appropriate executable. It is only enabled in standalone mode as Allegro DFI needs Python to run properly. Version preferred for Python installation is Python 2.5 or 2.6.
3. *Select your Allegro DFI menu load setting?*
   Enables you how the Allegro DFI menu load will happen in Allegro. The Allegro DFI has default behavior for an automatic load of the *Export to ADS* menu fragment in the Allegro menus. The default behavior for this automatic load of the menu fragment is just in front of Allegro **Help** button, but the Allegro DFI setup allows you to select any existing menu item in Allegro menu hierarchy.
   - *Enable automatic load of the Allegro DFI menu* flag
     Switches this automatic load mechanism on (default) or off. If you select the off setting you will need to manually configure you menu files (see below).
   - *Use default location before <Help>* flag just switches to the default behavior and disables the other choices.
   - *Select a specific position in the menu structure before or after the item*
     Provides a selection list of all available menu items in the current Allegro menu

hierarchy. By clickin on an item you specify the exact location of the Allegro DFI menu fragment in the entire Allegro menu structure. By default **Help** is selected in this list.

> **Note**
> Do not use foreign menu items (including the Allegro DFI list) when these are present in this menu list. Always take a basic item from the Cadence Allegro tools to avoid load problems if the tool configuration changes.

- *Before or After* dropdown field select precise control with respect to a known menu item.
- *Add menu item to start Advanced Design System* flag
  This enables or disables that you can start ADS from an additional menu item in the Allegro DFI menu if the integration with and ADS installation is enabled.
- *Specify ADS_LICENSE_FILE (needed to start ADS)*
  If the ADS menu item is enabled you need ADS licensing to be setup correctly. The script will try to find this setting if present in the Allegro environment but it needs to be specified here to make sure the start environment for ADS from Allegro is correct.

Setup and *Cancel* buttons allow to run or cancel the Allegro DFI configuration process inside Allegro.

Pressing the **Setup** button the runs the actual configuration script and this performs following tasks:

- Setup first makes up to four backup files with <name>.<ext>,(1,2,3,4) of all relevant configuration files that it might change.
- Removes any local ADFI configuration information from the selected ilinit files that it recognizes.

> **Note**
> Non local configuration loaded through other ilinit files in the Allegro environment path are not modified. These can interfere with this installation. If this happens please contact your EDA administrator to modify the load order.

- Creates an eemAdfiSetup.il file with actual configuration settings in ~/pcbenv.
- Add the (load ~/pcbenv/eemAdfiSetup.il) command to the selected <tool>.ilinit files.
- At the end of the confguration you are asked to restart your Allegro tool.

Advanced Manual Installation and Site Level Installations

This installation mechanism is for advanced usage only; when you want to share common setups between multiple users or if you use a low tier Design L version of Allegro which misses the scripting functionality to perform the automated configuration.

A text version of the installation instructions can also be found under the name "INSTALL" in the $HPEESOF_DIR/ial directory. This file can be opened with any text editor.

Note

- *HPEESOF_DIR* is the environment variable pointing to the head of the ADS installation tree.
- You must use "/"as a path delimiter inside the Allegro configuration files!

To install the Allegro DFI functionality in Allegro manually and/or on the site level use the following steps:

1. Prepare an Allegro DFI setup file called eemAdfiSetup.il with a text editor starting from:
    1. The template content (see inside the INSTALL file) shown by the following skill code fragment and store it in a text file called eemAdfiSetup.il.
    This is a skill script which defines the Allegro and environment variables used by the skill command `(load "../eemom.ini")` at the end. This load statement is responsible for activating the Allegro DFI skill context. Update the various settings according your local environment and make sure you remove the ; comment characters at the start of each line to activate that particular setting.
    Allegro DFI setup file eemAdfiSetup.il template

```
;; ------------------------------------------------------
;; --- BEGIN ADFI Agilent EEsof ADFI config
;; --- Modified:
;; Place or remove or the comment character (;) in front
;; and modify values if you want to change the ADS Allegro DFI load behavior.
;; enable/disable automatic menu load OFF or ON (default)
;(setShellEnvVar "EEMOM_MENU_AUTOLOAD=ON")
;; if automatic menu load is OFF you manually have to add menu fragment
;; $HPEESOF_DIR/ial/config/eemMenuFrag.men to all the
;; menu files you want in < allegro_install_path >/share/local/pcb/menus
;; and store the updated .men files in the appropriate menu location
;; menu string to search for when adding menu fragment
;(axlSetVariable "Eem_Menu_ItemPosition" "&Help")
;; position before or after the named item nil|t
;(axlSetVariable "Eem_Menu_AfterItem" nil)
;; add the Start ADS menu item
;(axlSetVariable "Eem_Menu_HasStartAds" 1)
;; IMPORTANT
;; !!! Independent of the platform Windows and Unix always !!!
;; !!! use / as the directory delimiter in this file       !!!
;; !!! And avoid directories with spaces in the name       !!!
;; specify the ADS installation directory if you want integration
(setShellEnvVar "HPEESOF_DIR=C:/ADS2009U1")
;; Probably best to always enable the license settings to avoid not found issues from
within Allegro
;; Note that additional preference settings for licensing can be required.
;; ADS 2009 Update 1 and earlier
;(setShellEnvVar "AGILEESOFD_LICENSE_FILE=")
;; ADS 2011.01 and later
;(setShellEnvVar "ADS_LICENSE_FILE=")
;; If no ADS installation is available a Python installation 2.5.x or 2.6.x must be
specified
;(setShellEnvVar "EEMOM_PYTHON=C:/Python26/python.exe")
;; Define the EEMOM_SKILL_DIR only in a user's local configurations not at site level
;; to keep the overloading capabilities for personal setups available.
;(setShellEnvVar "EEMOM_SKILL_DIR=/ial/skill/15.7")
;; Specify a site specific eemom.option file instead of the default under
```

```
;; /ial/config/eemom.option
;; note : the contents of the optional files $HOME/pcbenv/eemom.option and
;;        /.option will be added to the list
;(setShellEnvVar "EEMOM_GLOBALOPTIONSFILE=/eemom.option")}}
(load "/ial/skill/15.7/eemom.ini")
;; --- END ADFI Agilent EEsof config
;; -----------------------------------------------------
```

1.
   - When Allegro scripting works you can also derive a site level configuration from the ~/pcbenv/eemAdfiSetup.il file created by the *Quick Configuration For A Single User* procedure described above. It is the perfect starting point to create a setup for a multi user enviroment with minimal effort. Just update a few of the necessary settings and move it to the correct location.

Lookup all your site's skill configuration directories which you can find in the following locations and copy the updated Allegro DFI eemAdfiSetup.il file into the skill configuration directory that you want to use for loading the Allegro DFI.

- <cdsroot>/share/pcb/etc/skill (or a user-defined location specified by the CDS_SITE environment variable)
- $ALLEGRO_SITE/skill
- ~/pcbenv
- . (the program's start directory)

The system, site and user level allegro.ilinit (or other tool specific ilinit like apd.ilinit, cdnsip.ilinit, etc) file(s) can reside in any of these skill configuration directories. Allegro loads all of ilinit files in the order as defined in the previous item. Update the appropriate ilinit files to load the eemAdfiSetup.il only on time for each tool and from the correct directory.

1. Open inside the selected skill configuration directory the *allegro.ilinit* in a text editor.

   Note
   Create this file if it does not exist.

   Add the following line with the path correctly specified (remember the "/" delimiter)
   `(load "<path to the your site specific eemAdfiSetup.il>") ; Agilent EEsof EDA, Allegro DFI load`
   and save the file.
2. Repeat this procedure for apd.ilinit, cdnsip.ilinit and any other tool specific <tool>.ilinit file that exists in the skill configuration directories because these tools will ignore the allegro.ilinit files.
3. Check the skill configuration directories of all users for other <tool>.ilinit files that contain skill load statement for *eemAdfiSetup.il* or *eemom.ini*. These can cause multiple load conflicts and are better removed.

Add the *Export to ADS* menu section to each of the Allegro menu files that you want to have with the Allegro DFI functionality.

- This is an optional step with EEMOM_MENU_AUTOLOAD=ON in the eemAdfiSetup.il file.
  The automatic load mechanism will in this case check for the existence of required menu fragments. If it finds a manual configuration in the active menu it will skips the

load
otherwise the menu section is automatically added about 1 second after the Help item becomes available in the Allegro menu.

- This is required if you have set EEMOM_MENU_AUTOLOAD=OFF in the eemAdfiSetup.il file.
  This setting disables the automatic menu load mechanism and doesn't check for the availability of the Allegro DFI menu items anymore. It is now the responsibility of the user to setup the menu files statically.
  Editing of the menu file must be done for each menu file of an Allegro Platform based tool individually. We illustrate the process here for the Allegro PCB editor which uses the allegro.men file to initialise the menu structure.
- Example: install the *Export To ADS* menu components in the Allegro menu file.
  1. Copy **<allegro_install_path>/share/pcb/text/cuimenus/allgero.men** or the menufile you are currently using to:
     <allegro_install_path>/share/local/pcb/menus
  2. Open *allegro.men* in a text editor.
  3. Add the contents of:
     $HPEESOF_DIR/ial/config/eemMenuFrag.men to this file just before the entry *POPUP "&Help"*.

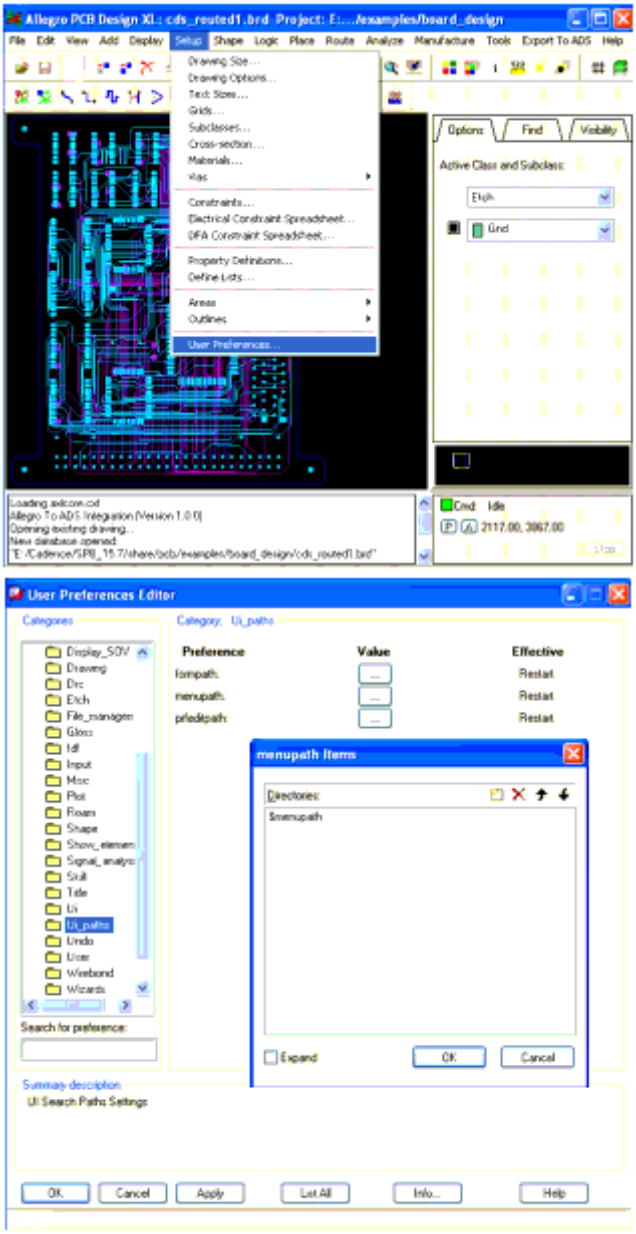
```
POPUP "Export To ADS"
BEGIN
   MENUITEM "&Setup ...", "MnxSetup"
   MENUITEM "Select &Traces ...", "MnxSelect"
   MENUITEM SEPARATOR
   POPUP "&Export"
     BEGIN
       MENUITEM "Export All", "MnxExportAll"
       MENUITEM "Export All As ...", "MnxExportAllAs"
       MENUITEM "Export Selected", "MnxExportSelected"
       MENUITEM "Export Selected As ...", "MnxExportSelectedAs"
     END
   MENUITEM SEPARATOR
   POPUP "State"
    BEGIN
      MENUITEM "Save State ...", "MnxSaveState"
      MENUITEM "&Load State ...", "MnxLoadState"
    END
END

POPUP "&Help"
```

  4. Save the file.

> **Note**
> An example of the popup menu to be added to the Allegro main menu can be found at: *$HPEESOF_DIR/ial/config/eemMenuFrag.men*

  5. If you need to change the menu location from < *allegro_install_path>/share/local/pcb/menus* to another directory you have to modify the *MENUPATH* Allegro variable to include the new directory.
     In this case, if for instance *menufile* resides in the working directory make sure that the working directory is first directory in Allegro's *MENUPATH*, then that file will be used.
     1. Selecting **Setup** > **User Preferences...** from the *Allegro PCB Design* main menu and adding the directory to where the *menufile* resides to the *UI_paths* menupath.



2. Adding the line:
   `(setShellEnvVar "EEMOM_MENUPATH=<directoryWhereMenufileResides>")`
   to your *eemAdfiSetup.il* file is an alternative solution.

Copy and update the *eemom.option* to the location of the EEMOM_GLOBALOPTIONSFILE setting. Often application and technology specific settings need to be added to the default eemom.option file in $HPEESOF_DIR/ial/config. It is a good idea in that case to keep the original shipped file and point the global option file setting to another location.
The configuration is now ready for use.

Check Allegro for Allegro DFI functionality

After a restart of the Allegro Platform tool verify that the menu fragment *Export to ADS* is available in the expected menu location and that one and only one Momentum Allegro Integration (Version n.n.n) is loaded. The load of the ADS 2011.01 version (Version 4.1.0) is shown in the following Allegro command window. One and only one such message

should be plotted to the Allegro log system.

- If you see multiple instances, check allo your allegro.ilinit (and <tool>.ilinit) files in the skill startup path and remove old references to eemom.ini or eemAdfiSetup.il.
- If the command window does not show this (Version n.n.n) line the load operation has probably been blocked:
  - For Cadence SiP and Package Designer specific setups, ADFI configuration is not picked up by apd and cdnsip after the restart when tool specific versions are around
  - Site configuration might block the load of local user setup.

Enabling the Allegro Import Design Kit in ADS

Enable the Allegro import interface into ADS by loading the import design kit *lib.defs* into the ADS 2011.01 workspace that you want to use for importing exported Allegro designs. Perform the following steps:

1. Unzip the design kit zip file *$HPEESOF_DIR/ial/design_kit/import_allegro*<VERSION>*dk.zip* by selecting ADS **DesignKits > Unzip Design Kit**.
2. This will unpack the design kit in the chosen path add add the IMPORT_ALLEGRO design kit to your list of favorite design kits.
3. Enable the *IMPORT_ALLEGRO lib.defs* file in every workspace that you create to do Allegro import operations. This will add the **Allegro Tools** menu in the menu ADS Layout Windows for this workspace. The items under this menu form the Allegro DFI functionality on the ADS side of this link.

Note
This design kit does not have dynamic unload capability. You need to restart ADS to remove the design kit from ADS. You must remove the *libs.defs* by selecting **DesignKits > Manage Favorite Design Kits...** from the workspace lib.defs file. After restarting ADS, the Allegro DFI import design kit will not be available in this workspace.

Using Allegro DFI in Allegro Package Designer and Cadence SiP

Allegro DFI is supported on Cadence SiP and Allegro Package Designer from Allegro SPB 16.01, 16.2 and 16.3. However, automatic cross-section stack conversion for Package Designer (APD) might not work due to ambiguity in the definition inside the Allegro platform. It is available in Allegro DFI version 3.1.4 of the *eemom.cxt* file and relies on the hierarchical export process.

Cadence SiP Basics

The following figure displays a cross section:

	Subclass Name	Type	Material	Thickness (MIL)	Conductivity (mho/cm)	Dielectric Constant	Loss Tangent	Negative Artwork	Shield	Width (MIL)
1		SURFACE	AIR			1	0			
2	TOP	CONDUCTOR	COPPER	1.2	595900	1	0	☐		5.00
3		DIELECTRIC	FR-4	8	0	4.5	0.035			
4	GND	PLANE	COPPER	1.2	595900	4.5	0.035	☒	☒	
5		DIELECTRIC	FR-4	8	0	4.5	0.035			
6	VCC	PLANE	COPPER	1.2	595900	4.5	0.035	☒	☒	
7		DIELECTRIC	FR-4	8	0	4.5	0.035			
8	BOTTOM	CONDUCTOR	COPPER	1.2	595900	1	0	☐		5.00
9		SURFACE	AIR			1	0			

In the cross section:

- Red represents package substrate that is a physical stack.
- Blue represents a die stack where only layer ordering is relevant.
  - No physical properties are attached.
  - This feature is different for APD.
  - DIESTACK is a conductor layer but properties can be defined on the dies.

Rules for SiP Cross Sections

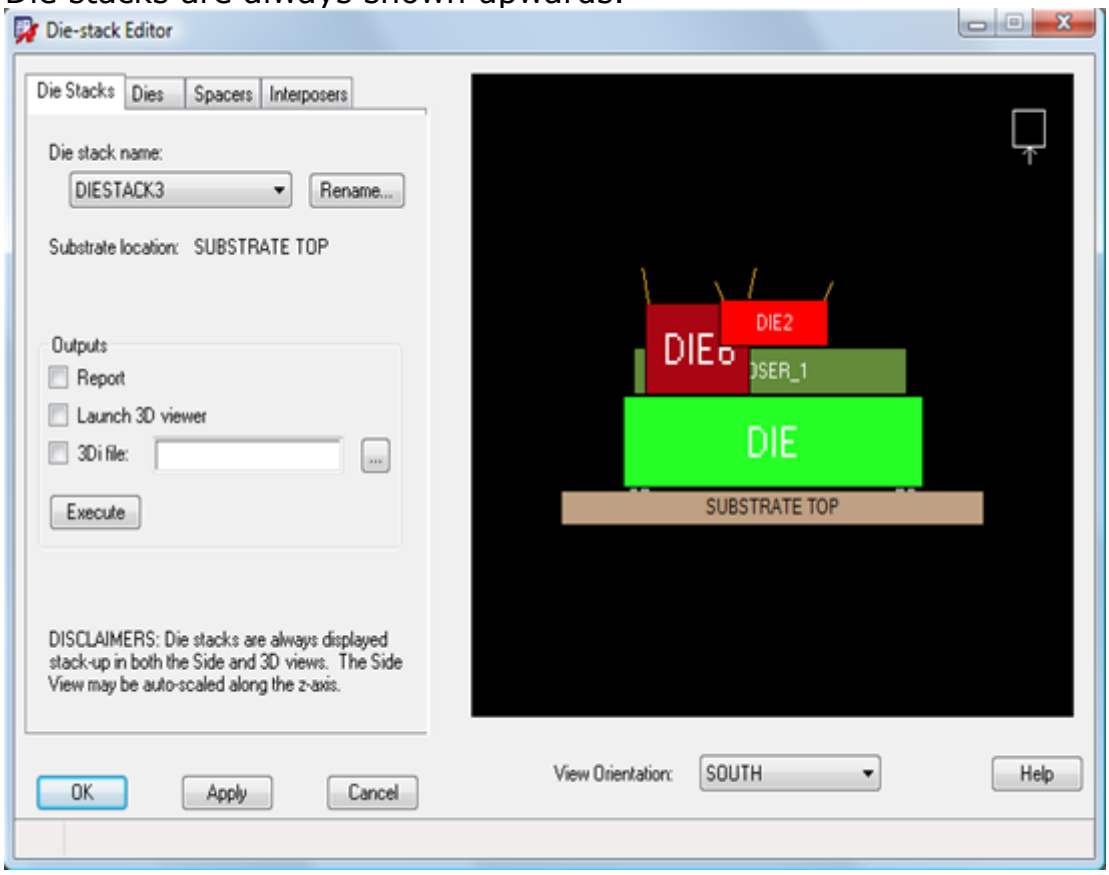
To define the Package Cross-Section (CDN SiP manual), select **Setup > Cross-Section** from SiP Layout to open the Allegro stackup editor.

- Add the appropriate layers between the surfaces.
- Each flipchip die requires a CONDUCTOR layer and a DIELECTRIC layer above and below it. Generally, you place flipchip dies on the top package substrate or bottom package substrate, or both.
- Each wire bond die requires a DIESTACK layer outside the package substrate with a DIELECTRIC layer above and below it.
- Each interposer requires a DIE layer outside the package substrate with a DIELECTRIC layer above and below it.
- Each spacer requires a DIELECTRIC layer that is named to allow the placement of geometry on it.

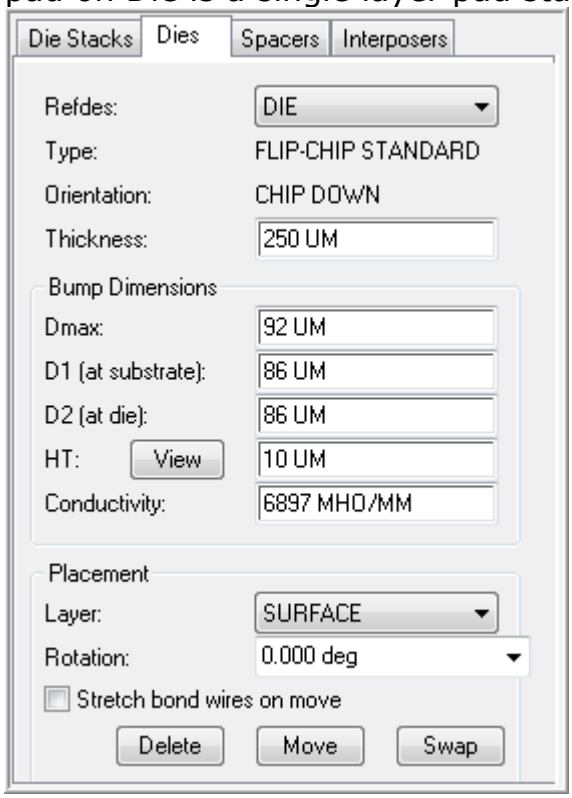
Assembly in the Die Stack

Access the device assembly editor through **Open Edit > Die Stack**.

- Die stacks are always shown upwards.

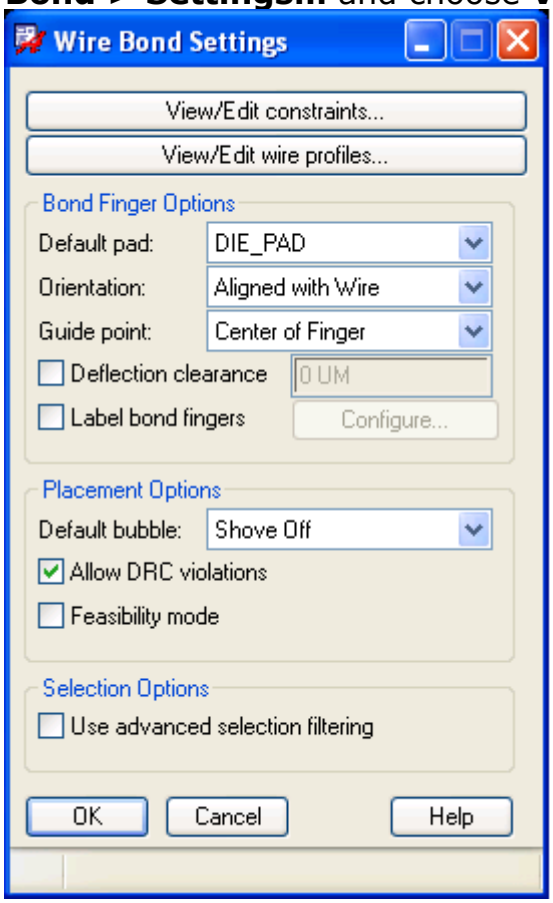


- a Die Stack starts either from SUBSTRATE TOP or BOTTOM.
- each device Die must use only one conductor (ETCH) layer (Die placement layer)! A pad on Die is a single layer pad stack under this rule.



Wire Bond Definitions

Allegro wire bond profile definitions can be accessed by selecting **Open Route > Wire Bond > Settings...** and choose **View/Edit wire profiles** or you can use toolbar buttons.

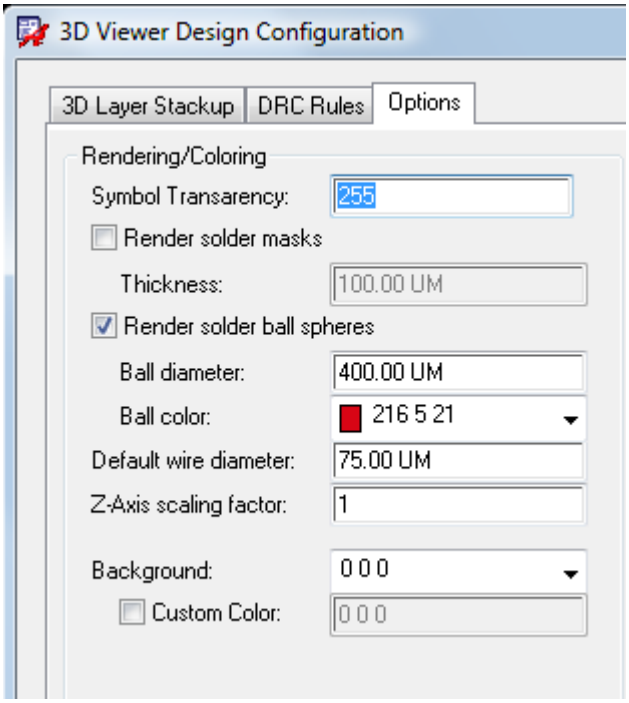


Make sure that wire bond profiles are properly defined before you try the export:

- Make sure the diameter is above zero.
- Use a bond wire material for which the properties are available in the Allegro/APD materials database. If not available the Allegro DFI will switch to GOLD.

Cadence SiP "View 3D Model"

Select **View > 3D Model >View** to open the **3D Viewer Design Configuration** dialog box, as shown in the following figure:



The Options on this dialog is important. For a BGA device you must notice that the solder balls in the Allegro Platform are defined in the 3D viewer and not in the design database. So the Allegro DFI needs to add that information also before export.

Differences between Cadence SiP and Allegro Package Designer (APD)

Cross Section

- Red represents package substrate which a physical stack package
- Blue is die level Physical properties of layers are attached. This is different for SiP.
  - DIE is a conductor layer (DIESTACK is name used in SiP)
  - All the physical properties of the single component stack are found here.
  - Solder mask like layers can cause connection difficulties if defined in the DIE stack part of the cross section.

		Subclass Name	Type	Material	Thickness (UM)	Conductivity (mho/cm)	Dielectric Constant	Loss Tangent	Negative Artwork	Shield	Width (UM)
1			SURFACE	AIR			1	0			
2		FLASH1	DIE	COPPER	30.48	595900	1	0			
3			DIELECTRIC	FR-4	203.2	0	4.5	0.035			
4		DDR1	DIE	COPPER	30.48	595900	1	0			
5			DIELECTRIC	FR-4	203.2	0	4.5	0.035			
6		M1	CONDUCTOR	COPPER	30.48	595900	1	0	☐		50.00
7			DIELECTRIC	FR-4	203.2	0	4.5	0.035			
8		M2	CONDUCTOR	COPPER	30.48	595900	4.5	0.035	☐		50.00
9			DIELECTRIC	FR-4	203.2	0	4.5	0.035			
10		M3	CONDUCTOR	COPPER	30.48	595900	4.5	0.035	☐		50.00
11			DIELECTRIC	FR-4	203.2	0	4.5	0.035			
12		M4	CONDUCTOR	COPPER	30.48	595900	1	0	☐		50.00
13			SURFACE	AIR			1	0			

No special module elements APD

- Die stacks does not exist
  - These are removed when importing Cadence SiP design in APD
  - Importing mcm into Cadence SiP may require update of cross section and layout
- Spacers and Interposers are mechanical symbols that exist between dies in die stacks. Just as die stacks do not exist within APD, neither do these elements. As a result, they are removed from the database completely.
- Cross section layer stacking rules are identical to SiP. i.e. you should not connect two DIE layers together.
- But the interpretation of the APD cross section is not uniquely defined
  - APD allows definition of molding, solder mask properties in physical cross section. SiP misses this capability in 16.01 and 16.2.
  - APD cross section translation is difficult because name dielectric layers can be both dielectric blocks as well as conducting via definitions.

Export to ADS

Export to ADS from Allegro Package Designer and Cadence SiP is very similar to Allegro PCB editor.

- Setup - Setup is same as that of PCB Editor.
- Select Traces - Nothing different from PCB Editor, except in the Layer Select tab. See next slide
- Flat Export - Nothing different from PCB Editor. Wire bonds are exported as polygons on special unmapped layer. Solder balls are ignored or created on special unmapped mask layer.

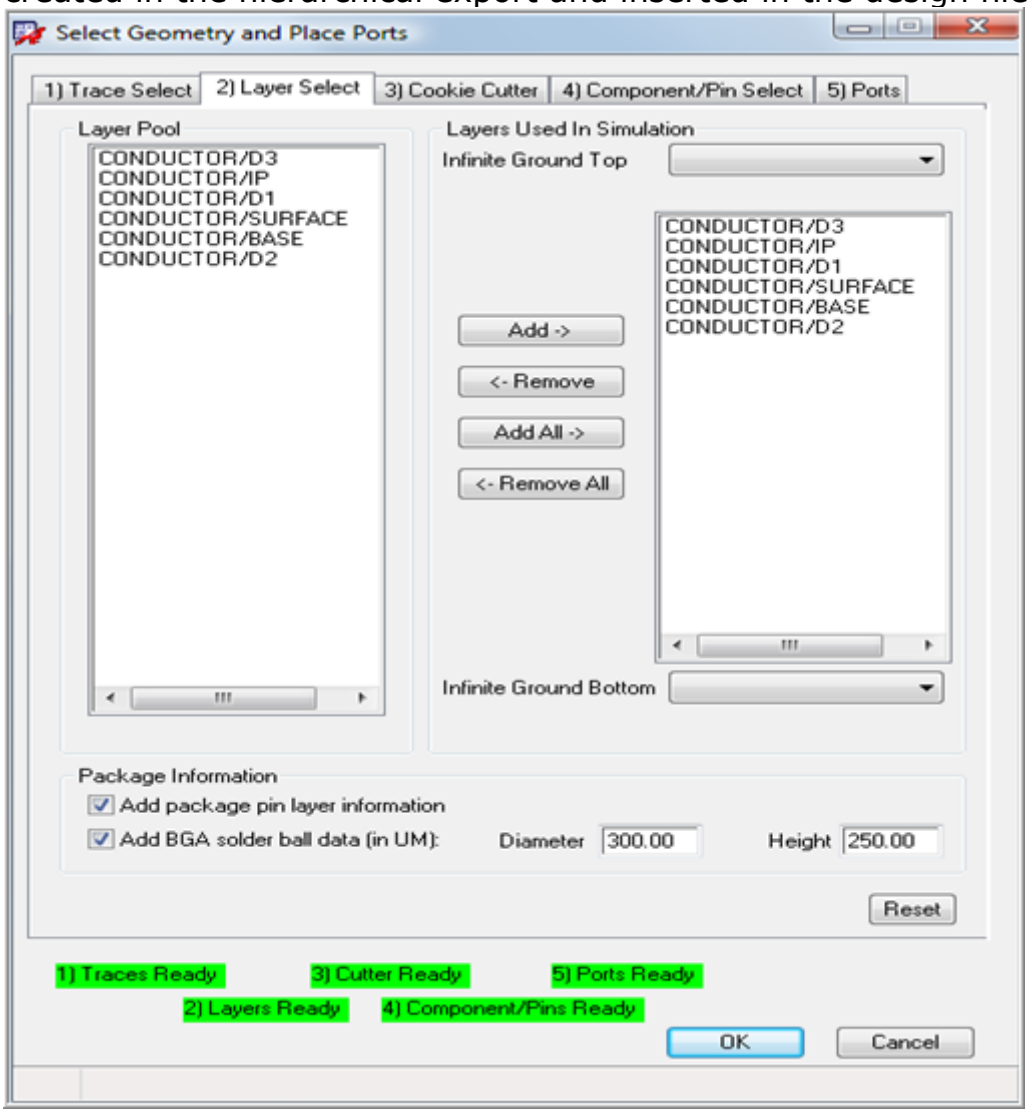
The differences are related to hierarchical export and the associated component creation. It starts at the selections and continues with the hierarchical export. This what the next few sections describe.

Layer Selection

You need to add the package pin information especially in case of BGA designs that have solder balls which are not defined inside the Allegro database.

- Decide to add package pin layer information
  - Exports package pin shape on dedicated layer.
  - With hierarchical export
    - Package pins layers are mapped in the ADS substrate.
    - Package pins are connected at the board level.
  - Not mapped with flat export!
  - Simplification rules in the eemom.option for these shapes can be "pad" or "pin" based.
- Add BGA solder ball data
  - BGA ball data not accessible through Allegro database API
  - You need to export the package ball shape on dedicated layer.
    - If specified diameter is bigger than contact area above/below, diameter can be clipped! This is done to create valid Momentum via setup.

    - If you disable this flag the shape might case connection errors in a Momentum setup.
- Special components with the ball properties as parameters of the component are created in the hierarchical export and inserted in the design hierarchy.



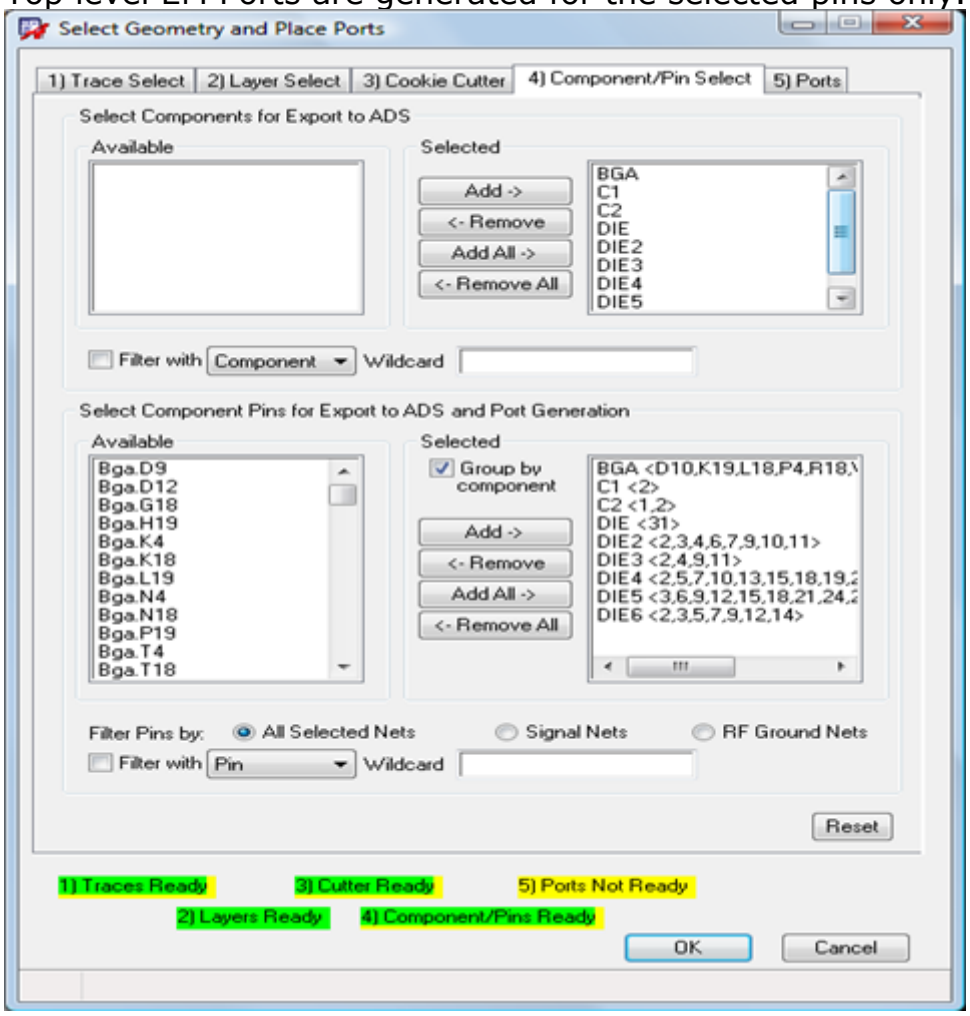
Component Select

Selection of components is identical as in Allegro PCB

- The package or module is a named component in Cadence SiP and APD, e.g. BGA
- Each die is also a component instance
- For Cadence SiP die stacks form a hidden component grouping for stacked chips.
    - Created and picked up in the background during export when multiple chips are stacked or if a stack contains bondwires
    - The diestack grouping is skipped if only one flip chip component is part of a die stack
- Export of selected components
    - Interposers and spacers are implicit components added when needed. Material properties are available in the Die Stack definition.
    - The Allegro DFI will add automatically all die shapes as a dielectric brick if needed. This should result in completed and correct diestack structures usable in

3D EM simulations.

- Top level EM Ports are generated for the selected pins only.



Hierarchical export

Select **Export to ADS > Export > Selected Hierarchy** behaves as for Allegro PCB but adds Cadence SiP and APD specific structures.

- Bondwires are translated with their profile and grouped inside a component for each diestack (gold wire are used if the bond wire material is undefined)
- Allegro 16.3 non-standard Bondwires are supported but need verification of their attachment when imported.
- Export will add the flip chip die ball information from diestack properties as these are available in the design database.
- Material properties for dies are set as silicon as these are unknown in a SiP design.
- To simplify substrate generation the Allegro DFI will add component based layer names when height differences can occur. The SiP/APD cross section layers are insufficient for complex die structures and only provide an assembly order.
- As described before the entire hierarchical design is stored in a file <name>.adfi

Note
The dynamic layer name generation might change layer numbering from export to export. It is recommended to import in a clean ADS workspace in ADS 2009 Update 1. In ADS 2011.01, the workspace library is created automatically by the import design kit.

Allegro DFI Import into ADS

This is similar to the import of Allegro PCB designs.

Importing into Layout

Allegro Tools > Import Allegro Layout automatically picks up flat and hierarchical export when they exist:

- – <library>:<design> is the flat version.
- – <library>:<design>_adfi is the hierarchical variant.

- main design i.e. BGA is the main package layout
- <library>:DIESTACKi contains stacked structures
- <library>:WB_<instance name> wire bond component
- <library>:DIE<i> chip die instances

Viewing the Flat Layout

Open **<library>:<design>** design:

1. Wire bonds are exported as polygons on special unmapped layer in flat design
2. BGA balls are added if enabled on special unmapped layers (*Ppin* and *Pball*)
3. Essentially just the package routing is defined no 3D features above and below

Viewing the Hierarchical Layout

Open the **<library>:<design>_adfi** design and in design window select **View > 3D View > All**. All available 3D information is normally converted. The Hierarchical import of components is listed below:

1. Bondwires/ Bonding ball / BGA solder balls / Die stacks
2. Dies are dielectric bricks in substrate
3. Bondwiresare combined SBOND
4. Discrete components (RLC)
5. Footprints of other components
6. Parameters and properties attached when possible

You can synchronize a design from Layout into Schematic by synchronizing it step by step at each hierarchy level.

Symbol, item def + netlistCB for bondwires

In any schematic window you can create partial circuit level designs. Place a WB_<die> component to see the available parameters. Bondwire WB_<die> components have parameters for:

- FEM simulation on the SBOND instances
- Also have a Philips model attached
  - Can be used directly from schematic
  - Netlists the real coordinates of design from the layout.
  - Height from ground needs to be set in a parameter!

Solder Ball component for BGA pins

1. Place in Layout or Schematic <package>*PIN*(TOP|BOTTOM)
2. Solder ball properties derived from Layer Select Tab in export
3. Can replace the ball by a 3D shape for FEM simulation if needed using these parameters
4. Ball substrates have been merged behind the scenes with cross section substrate

Flip Chip Dies include Bump Attributes from SiP

From **<library>:<design>_adfi:layout** double click a DIE instance. For flip chip components, die and bump attributes are translated from Cadence SiP into ADS. This is not possible for APD as this information is not available in the database. Note that the pin shapes on the chip die are defined by the pad shape on a package routing layer not by the IC mount pads as they exist on the chip. To resolve this problem, Cadence SiP and APD users often need to use a workaround in the layer stack. This might create the following import issues:

- Undefined substrate as user might try to fix this in by adding layers in SiPcross section.
- Unconnected pins issue

Note
Always verify the substrate definitions in ADS after importing the files.