

**© Agilent Technologies, Inc. 2000-2011**

5301 Stevens Creek Blvd., Santa Clara, CA 95052 USA

No part of this documentation may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

**Acknowledgments**

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

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

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

Intel® Math Kernel Library, <http://www.intel.com/software/products/mkl>

SuperLU\_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser

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

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: <http://www.cise.ufl.edu/research/sparse/umfpack> UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at <http://www.cise.ufl.edu/research/sparse> . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at <http://www.cise.ufl.edu/research/sparse> . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. <http://www.mathworks.com> . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at <http://www.netlib.org> ). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

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

## Advanced Design System 2011.01 - ADS Quick Reference

Things You Must Know About ADS 2011	6
ADS Workspace	6
New Menu Options	9
EM Simulation	9
User Compiled Models (UCMs)	10
Design Kit (PDK)	10
ADS Project Upgrade to ADS Workspace	12
Conversion of Project with Layout Components	13
ADS 2011 PDK Upgrade	17
Contents	17
Upgrading PDKs for ADS 2011	17
PDK Vocabularies in ADS 2011	23
Backward Compatibility in PDKs	24
Working with Upgraded PDKs in ADS 2011	25
Frequently Asked Questions (FAQs)	29
ADS 2011 DesignGuide Upgrade	32
Contents	32
Upgrading DesignGuides for ADS 2011	33
Before You Begin	33
Procedure to Upgrade DesignGuide to be ADS 2011 Compatible	33
Preparing AEL Files to ADS 2011 Compatible Format	35
Upgraded DesignGuide Folder Structure	37
Troubleshooting DesignGuide Upgrade	39
Glossary and Abbreviations	40
ADS	40
AEL	40
ASM	40
Cell	40
Library	40
License File	40
OpenAccess	40
OpenAccess Database	40
SiP	40
Technology	41
UCM	41
View	41
Workspace	41
Differences: ADS 2011 Vs. ADS 2009 Update 1	42
ADS Main Window	42
Design and Architecture	43
Layout Window	44
Schematic Window	45
Symbol Window	47

# Things You Must Know About ADS 2011

Advanced Design System (ADS) 2011 provides a complete new *design environment* (adstour) and terminology for existing ADS users. ADS Projects are replaced with workspaces, designs are now Cells and Views. To use your existing ADS Projects and designs (created using earlier versions of ADS) with ADS 2011, you must upgrade them to an *ADS Workspace* (oaqkref).

For more information on getting started with ADS 2011, see:

1. *ADS Quick Reference* (oaqkref), for quick instructions on how to upgrade existing ADS Projects, PDKs, and Design Guides.
2. *ADS Differences* (oaqkref), for information on key differences between ADS 2009 Update 1 and ADS 2011.
3. *ADS Quick Install and License Setup* (qinstall), for quick installation and license setup instructions (using ADS License Setup Wizard).
4. *Advanced Design System Quick Start* (adstour), to get started with ADS 2011.

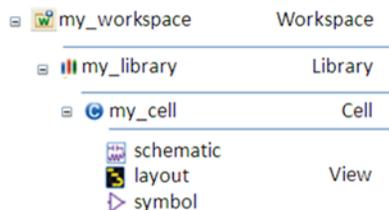
## ADS Workspace

ADS 2011 uses Workspaces, instead of Projects as a directory, to store and organize the design work. A Workspace is similar to an ADS 2009 Update 1 and earlier Project. It contains data that includes simulation results, data display files, and other data files. The major difference between Workspace and Project is that the design work is now stored in libraries instead of the networks directory. Figure [ADS Project Vs. ADS Workspace](#) displays the hierarchy differences between an ADS Project and an ADS Workspace.

A Workspace includes:

- **Library:** A Library is a directory that holds cells and a definition file, such as `lib.defs`. This file contains a summary of all the libraries selected into the given workspace and their mode of operation (Read-only, Non-Shared, or Shared). A Library also defines the technology (layers, resolution, and layout units) to be used by the Views created in that library. For more information, see [Library](#).
- **Cell:** A cell is a sub-directory that contains zero or more views. For more information, see [Cell](#).
- **View:** A View is a sub-directory in a cell that stores the design work such as schematic, symbol, or layout, and can store other related data, such as an EM Setup. For more information, see [View](#).

### Library:Cell:View hierarchy



In ADS, the design work is organized into *Library:Cell:View* hierarchical architecture. Instead of residing in a *networks* sub-directory, a design is now organized into one or more libraries. Each Library is associated with a technology and the design data is broken into multiple storage points. Each view of a design is stored separately (in a cell). There is no limit on the type or number of views that can be stored in a cell.

Main files included in ADS Workspace are:

- `lib.defs`: This file contains the library name associated with a given workspace along with their mode of operation. For more details, see [Library Definition File](#).
- `de_sim.cfg`: This file is general ADS configuration file and is used to customize the user interface and simulation environment. The Customization is possible for an individual workspace or all workspaces.
- `hpeesofsim.cfg`: This file is used to set simulation and design kit variables and set their path locations.

## Workspace Features

The following are the key features of an ADS workspace:

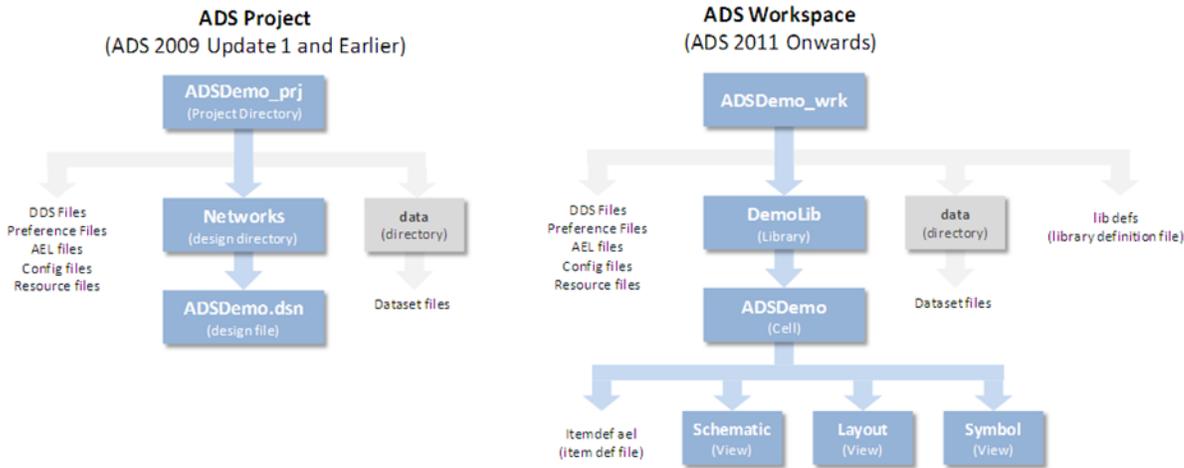
- Workspaces can have multiple libraries. Each Library can have multiple Cells, and each Cell can have multiple Views.
- Workspaces can be archived or un-archived.
- Workspaces contain data that includes simulation results, data display files, and other

data files.

- In an ADS Workspace,
  1. A component name consists of the library name and cell name of the component.
  2. A design name consists of the library name, cell name, and view name of the design.
  3. Forms and Formsets are defined in a library.

The following figure displays the hierarchy differences between an ADS Project and an ADS Workspace:

ADS Project Vs. ADS Workspace



See also

**Using Workspace (adstour)**

**Library**

A Library is a subdirectory that includes cells. Libraries are used by specifying the subdirectory in a library definition file, such as `lib.defs`. This file defines name of the library and the mode of operation (Read Only, Non-Shared, or Shared). Files within the library itself define the technology (layers, resolution, and layout units) to be used by Views created in that library.

All types of designs are contained in libraries. Design kits in ADS 2011 contain libraries with cells. User workspaces contain libraries. Native ADS components themselves are also contained in Libraries.

The process information such as layer definitions, units, and substrates are defined in the technology of a library. You can create multiple libraries within a complete design hierarchy. Library usage is defined by a file `lib.defs`. A library may reside physically in the workspace directory.

See also

**Library Definition File**

**Key Features**

The following are the key features of a Library:

- Each Library has a unique name and path, specified in a library definition file. (Note that the library itself does not specify its name. The name must be specified correctly in a `lib.defs` file.)

**Note**  
Two libraries with the same name cannot be open at the same time.

- Associated with a Technology that defines physical layers, units, etc. The technology can be a native one created for a library, or it can be referenced to another library's.
- Contains zero (0) or more Cells.
- Loaded/unloaded independently or collectively loaded through a `lib.defs` file.
- Defines the technology (layers, resolution, layout units) that it will use. All views in a library uses the same technology.

- Can be opened in following three modes:
  1. **Read-only Mode:** In this mode you cannot modify or commit changes to the library's cells and the cellViews. Only Save As will work.
  2. **Shared Mode:** In this mode you can jointly modify and/or commit changes to the library's cells and cellViews with others that might have access to the same ADS library location.
  3. **Non-Shared Mode:** In this mode only you can commit any changes to the library's cells and cellViews when library is open.
- Can reside physically anywhere in the system, with the following conventions:
  1. All ADS components are inside Read-only ADS libraries which are stored with the product.
  2. PDK's are also libraries, except that the addition of the `eesof_lib.cfg` file into the library gives ADS additional information about the other things stored in the kit. The PDK's library is usually stored as a directory inside the PDK, and the `lib.defs` file is stored in the root of a PDK.
  3. Normal end-user libraries are usually stored in a workspace. You can add a library from another workspace into the current workspace.

**Note**  
 In ADS 2011, Workspaces are not included inside another workspace. Instead, a `lib.defs` file from another workspace can be added to the current workspace using **DesignKits > Manage Libraries...** from the ADS Main Window.

See also

### Using Libraries (adstour)

## Technology

The process information such as layer definitions, units, and substrates are defined in the technology of a library. A library may:

- Completely define its technology
- Contain no technology.
- Contain a single technology.
- Contain a technology that inherits multiple technologies from other libraries.
- Point to a technology defined in another library.

## Cell

A Cell is a container for Views. It is an equivalent of an ADS design (ADS 2009 Update 1 and earlier releases), as it may contain schematics, layouts, and symbols.

**Note**  
 Many other OpenAccess based tools will refer to a single View within a cell as a "Design."

Each Cell:

- Must have a unique name in the library, although cells with the same name may exist in another library.
- Contains zero or more views.
- May have multiple views of the same type.
- May have a component definition, edited by choosing **File > Design Parameters...** while editing one of its views.

See also

### Using Designs (adstour)

## View

A View is a sub-directory in a cell that stores design information such as schematic, symbol, or layout. Views may also store an EM simulation setup or an EM Model. Each view is a container that stores a file or a database object. It is a specific representation of a cell. All views in a given library use the same technology.

Each view:

- is associated with a Cell.
- must have a unique name in the Cell.

**Note**  
 Schematic, Layout, and Symbol Views should generally be named as *schematic*, *layout*, and *symbol* unless you are using polymorphism.

- Has a type (schematic, layout, symbol, EM Setup, EM Model).

## Library Definition File

A library definition file (such as `lib.defs`) defines the name, path, and access mode of a library. It also includes references to other library definition files that are present on the file system. When a library definition file loads, each of its defined libraries opens and every referenced library definition file gets loaded. Each workspace has its own Library Definition file.

Library definition file:

- defines zero (0) or more libraries.
- can exist anywhere in the system.
- can also have references to other library definition files.
- can have any user defined name (`.defs` extension is recommended for library definition file).

### Notes

- Multiple library definitions files can exist in the same directory.
- You cannot reliably rename a library simply by editing a `lib.defs` file. You must use the **File > Rename Library...** menu command from the ADS Main Window.

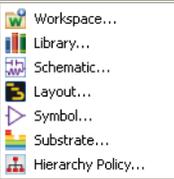
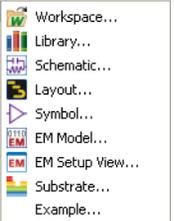
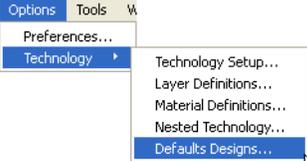
## Creating a New Design

For instructions on creating a new:

- Schematic, see *Creating a New Schematic* (adstour).
- Layout, see *Creating a New Layout* (adstour).
- Symbol, see *Creating a New Symbol* (adstour).

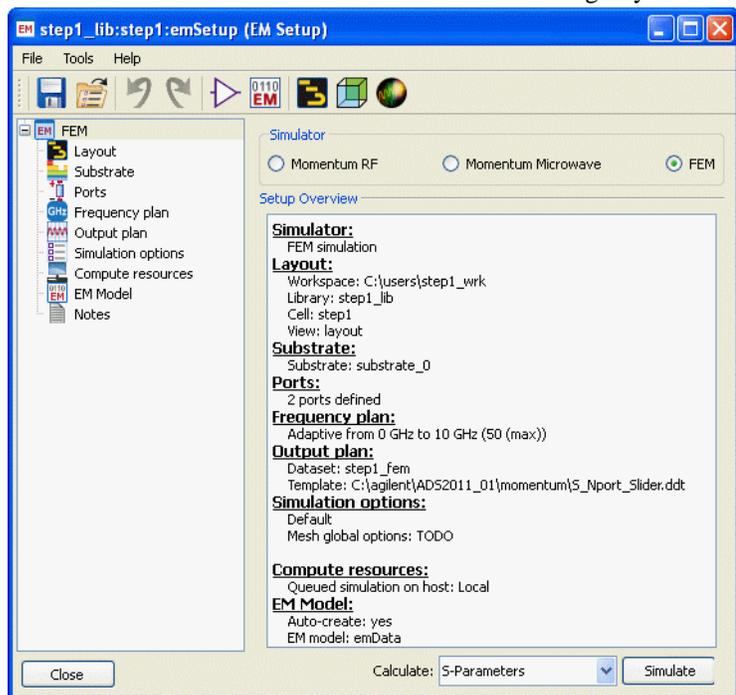
## New Menu Options

This section lists some of the new menu options introduced in this release.

Menu Name (location)	Screen	Remarks (if any)
File > New (from ADS Main Window)		From ADS Main Window, this menu provide options to create new workspace, design, substrate, etc.
File > Open		From ADS Main Window, this menu provide options to open workspace, design, substrate, etc.
Options > Technology		From ADS Main Window, this menu provide options to access Technology Setup, Layer and Materials definition, Nested Technology, etc.

## EM Simulation

You can perform you all your Momentum and FEM simulation activities from a central place, i.e., EM Setup dialog box. To start EM setup dialog box, choose **EM > Simulation Setup** menu from Layout window.



## Substrate Editor

ADS 2011 has a new Substrate Editor. A *substrate in EM simulation* describes the media where a circuit exists. An example is the substrate of a multilayer circuit board, which consists of layers of metal traces, insulating material, ground planes, *vias* that connect traces, and *air* that surrounds the board. A substrate definition enables you to specify properties such as the number of layers in the substrate, the dielectric constant, and the height of each layer for your circuit.

For more details, see *Substrate Editor* (adstour)

## User Compiled Models (UCMs)

To use your existing User Compiled Models (UCMs) with ADS 2011, you must recompile them separately for each platform, using the compiler officially supported by ADS.

## Design Kit (PDK)

If you want to upgrade your existing PDKs (created using previous ADS versions) in ADS 2011, you must know that:

- PDKs created using ADS 2009 Update 1 and prior releases must be upgraded, as described in *ADS 2011 PDK Upgrade* (oaqkref), to use in ADS 2011. A PDKs can be upgraded in such a way that it works in ADS 2011 as well as previous ADS releases.
- Agilent recommends you to create new PDK using a previous ADS release and then upgrade the PDK in ADS 2011.
- There may be changes required in AEL files (of PDKs) to make them ADS 2011 compatible to support new features introduced in ADS 2011. Also, as ADS 2011 is a completely new platform, AEL modifications may also be required in your PDKs where there are some use-model changes. For more information, see *ADS 2011 PDK Upgrade* (oaqkref).
- During the PDK conversion process you may encounter some errors or warnings that have not been reported in the previous versions of ADS. ADS 2011 has enhanced error reporting capabilities and it is strongly suggested that you fix these errors and warnings in the original PDK and then upgrade the same before using in ADS 2011.

## Upgraded PDKs

If you are using upgraded PDKs in ADS 2011, you must know that:

- PDKs are no longer installed or uninstalled as they were in prior ADS releases. Just like libraries, they are added or removed from the workspace. Hence, PDKs are used at the workspace level in ADS 2011. They do not have scoping at project/user/site levels – they are managed like other libraries. Instead, Site Administrators can add "Favorite PDKs" for their users as a convenience.
- Every library (and thus a PDK) has a unique namespace in ADS 2011. So if you create a component named R in your PDK, it will not clash with the ADS in-built

component R.

- While converting a project to a workspace, if you don't include the converted PDK that is used by that project, your designs in the workspace will not work properly. So first convert your PDK, and then convert your project which uses that PDK.

For more information on PDK Upgrade, see *ADS 2011 PDK Upgrade* (oaqkref).

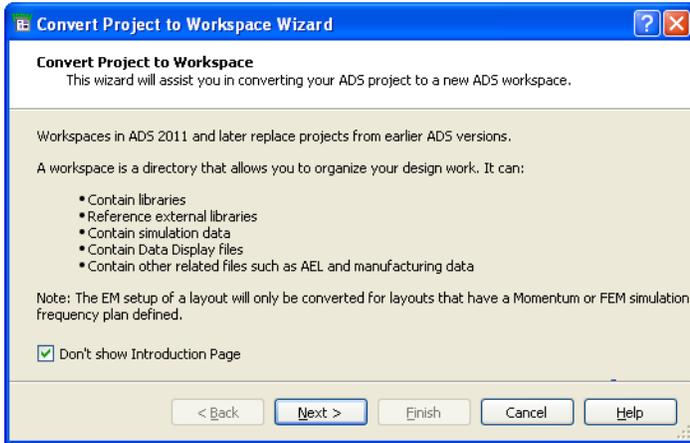
# ADS Project Upgrade to ADS Workspace

Follow the steps below to convert your ADS project, created using ADS 2009 Update 1 or earlier version, into workspace:

## Important

All ADS projects and PDKs created using ADS 2009 Update 1 or earlier version (or even in previous ADS alpha releases) must be upgraded or reconverted in the latest **ADS 2011** release.

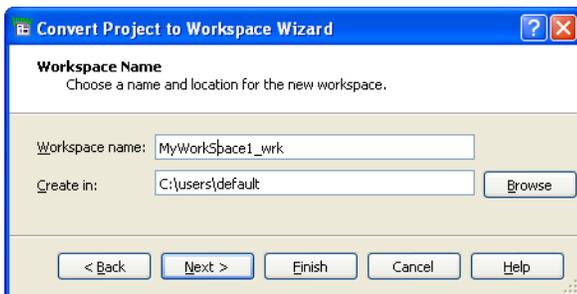
1. Start ADS.
2. From the main ADS Window, choose **File > Convert Project to Workspace** to start the **Convert Project** dialog.
3. Select the ADS project you want to convert to workspace and click **Choose** to start the **Convert Project to Workspace Wizard**.



## Note

**Convert Project to Workspace Wizard** starts automatically if you try to open any ADS project created in ADS 2009 Update 1 or earlier versions.

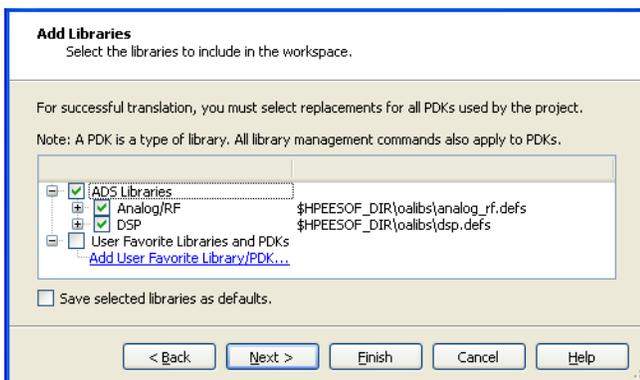
4. Click **Next**.
5. Enter the new workspace name, e.g. MyWorkSpace1\_wrk, and location. Click the **Browse** button to select the location. Click **Next**.



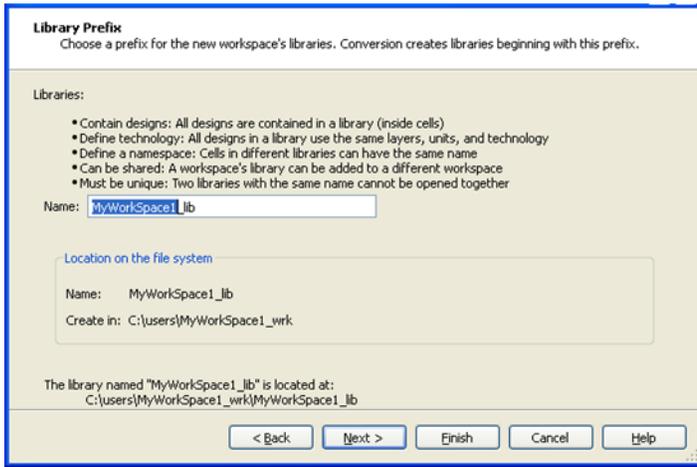
## Note

The default workspace name is project name \_ "\_wrk", which you can change.

6. Under Add Libraries, select the libraries to be added in workspace and click **Next**. You can also include ADS in-built libraries or add custom libraries.



7. Enter a prefix name for the library that will be created for this workspace and click **Next**.



7.

**Note**  
The default library name is the project name with `_lib` suffix, which you can change or remove.

8. A Summary window is displayed with workspace name and libraries included in the workspace. You can click **Back** button to go back and make changes in the previous screens.
9. Click **Finish** if no further changes are required. On successful conversion, 'Conversion finished' message is displayed and `MyWorkSpace1_wrk` workspace is successfully created.

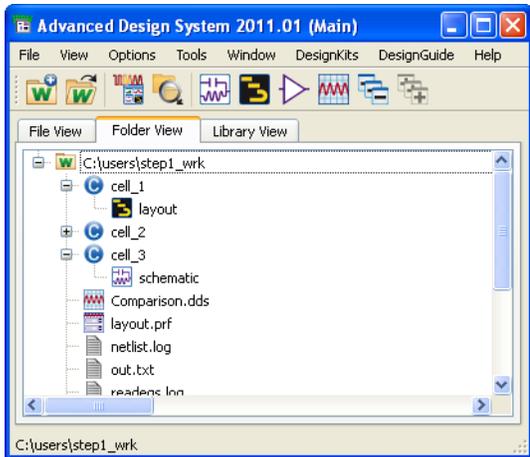
**Notes**

- If you receive any warning message, verify that you have included all the necessary libraries. For any further help, contact Agilent EEsof EDA Technical Support.
- For more details about ADS workspace, please see *Workspace* (oaqkref).
- If you try to open an ADS project created in ADS 2009 Update 1 or earlier version, Convert Project to Workspace Wizard starts automatically.

After conversion,

- Designs in `networks` directory gets converted to Cells (available in the library created during conversion process)
- Representation of each Design gets converted to corresponding Views in the Cell.
- The workspace contains a `lib.defs` file which contains a summary of all the libraries selected into the given workspace and their mode of operation.
- The Technology files gets created for each individual library.

The following figure displays the **Folder View** of the converted project:

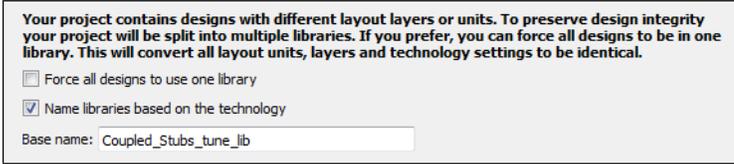


## Conversion of Project with Layout Components

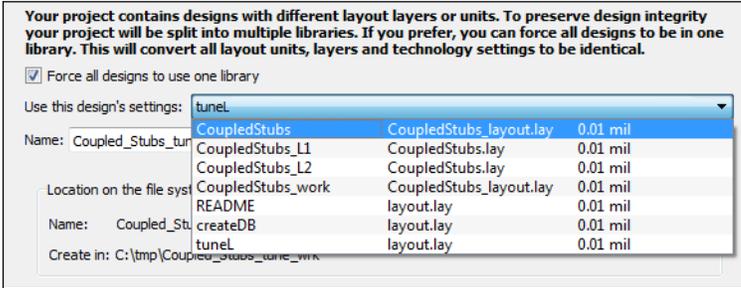
### Avoid the Split in Multiple Libraries

EM/Circuit co-simulations may fail when a project gets split into multiple libraries because substrates defined in one library cannot be used from another library.

During project conversion, you will see a note that the project contains designs with different layout layers or units. Each Layout Component references its own copy of a layout layers file.

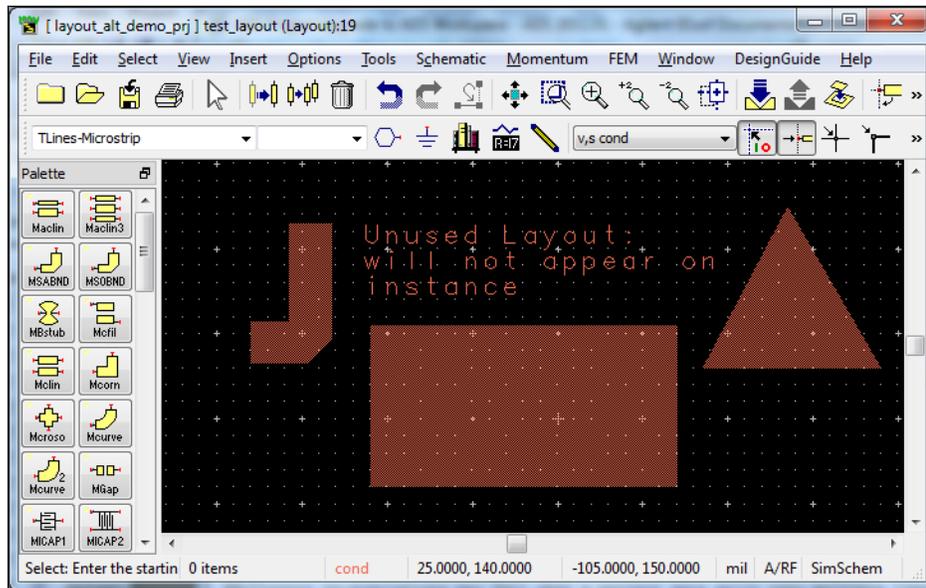


To avoid the split in multiple libraries, choose *Force all designs to use one library* and select to use the nominal Layout Component design settings for the single library.

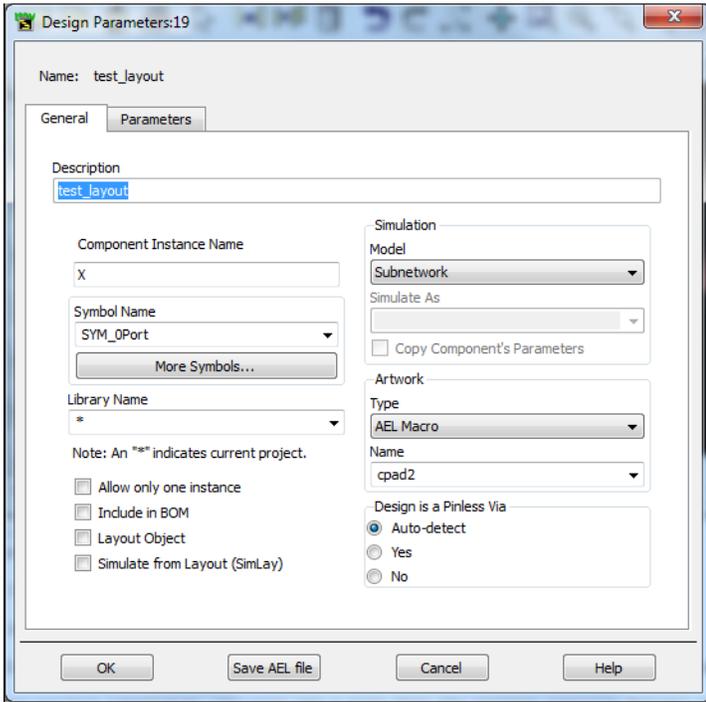


### Why does the translator sometimes make a "layout\_alt" view for my layouts?

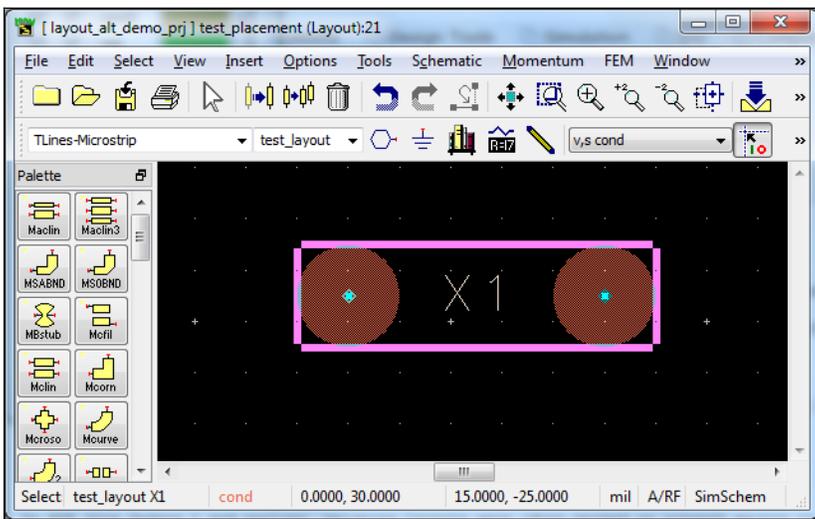
In ADS 2009 Update 1 and earlier, existing layouts were often marked as artwork macros or as fixed artwork. When placed as an instance on another design, the contents of the layout were not used. However, these layout contents were still saved and could be used for other purposes. The image below displays good example of this. For example, this layout:



has these **File > Design Parameters** set in ADS 2009 Update 1:



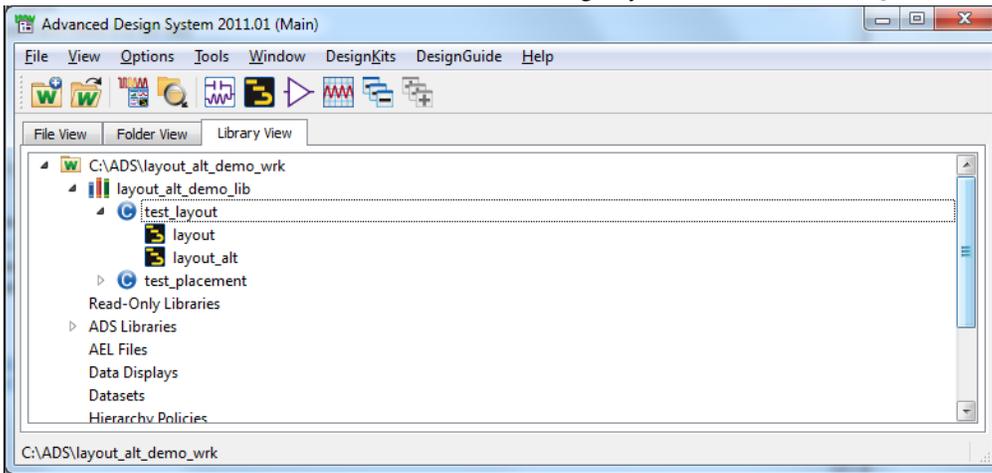
This means that the layout shown above will not be used when an instance is placed on another design. Instead, the cpad2 artwork macro will be run. When placed on another design, the result is:



When this layout is converted to ADS 2011 or later, these two uses are split up.

1. A "layout" view is created which contains the artwork macro. The layout itself will be empty, but the **File > Design Parameters** view-specific settings will be set to use the AEL macro. This view gets the name "layout" because it is the one that will be placed by default when the design is used as an instance, yielding results identical to ADS 2009.
2. A "layout\_alt" view is created which contains the layout as drawn. The **File > Design Parameters** view-specific settings will be set to use the layout as entered. This "layout\_alt" view can be placed as an instance on another design only by explicitly specifying the view to place. This layout is created by the translator because it is possible that it contains important information. It will not be created if the ADS 2009 Update 1 design's original layout was empty.

After conversion, the library view will be:



A "layout\_alt" can also be created in the same way if the ADS 2009 Update 1 **File > Design Parameters** artwork settings is "Fixed."

## ADS 2011 PDK Upgrade

### Important

PDKs cannot be successfully upgraded by the end users of the PDKs. The upgrade process requires that all source be present and frequently requires modifications to the PDK before conversion is successful. Contact your PDK vendor to inquire about upgraded PDKs for ADS 2011.

This topic describes the procedure to upgrade your existing PDKs and projects that use those PDKs, to the new ADS 2011 environment. It also describes the upgraded PDK's structure and some important points that you should know about PDKs in ADS 2011.

### Note

If you need upgraded PDKs, please contact your foundry. Your existing ADS PDKs will not work as it is in ADS2011. These PDKs need to be upgraded to ADS2011 compatible format. You can also use the Non Linear Demokit that comes within ADS for your evaluation if you do not get access to upgraded foundry PDKs.

## Contents

- [Upgrading PDKs for ADS 2011](#) (oaqkref)
- [PDK Vocabularies in ADS 2011](#) (oaqkref)
- [Backward Compatibility in PDKs](#) (oaqkref)
- [Working with Upgraded PDKs in ADS 2011](#) (oaqkref)
- [Frequently Asked Questions \(FAQs\)](#) (oaqkref)

## Upgrading PDKs for ADS 2011

### Important

PDKs cannot be successfully upgraded by the end users of the PDKs. The upgrade process requires that all source be present and frequently requires modifications to the PDK before conversion is successful. Contact your PDK vendor to inquire about upgraded PDKs for ADS 2011.

The following topics cover the procedure to upgrade PDKs for ADS 2011 and provide details about the upgraded folder structure:

- [Three Things You Must Know for a Successful PDK Upgrade](#)
- [Procedure to Upgrade PDKs to ADS 2011 Compatible Format](#)
- [What's New in the Upgraded PDK](#)

### Three Things You must Know for a Successful PDK Upgrade

- PDKs created using ADS2009U1 and prior releases do not work in their present form in ADS 2011. These PDKs must be upgraded to make them ADS 2011 compatible. The current procedure to create PDKs for ADS 2011 is to first create a PDK for ADS releases prior to ADS 2011 and then convert it to ADS 2011 compatible format. For this release, there is no recommended procedure to create a PDK from scratch. The PDKs are upgraded in such a way that it works in ADS 2011 and prior releases.
- There may be changes needed in the PDKs *ael* files to make them ADS 2011 compatible. This is required to support some new features that have been introduced in ADS 2011. Also, as ADS 2011 is a completely new platform, *ael* modifications may also be required in your PDKs where there are some use-model changes. These will be described in the following sections of this topic.
- During the PDK conversion process you may encounter some errors or warnings that have not been reported in the previous versions of ADS. ADS 2011 has enhanced error reporting capabilities and it is strongly suggested that you fix these errors and warnings in the original PDK and then convert it for ADS 2011.

### Procedure to Upgrade PDKs to ADS 2011 Compatible Format

### Note

PDKs created using ADS2009U1 and previous releases must be modified to work in ADS 2011. The current procedure to create PDKs for ADS 2011 is to first create the PDK for ADS releases prior to ADS 2011 and then convert it to ADS 2011 compatible format as explained in this section.

This procedure involves following steps:

1. [Preparing AEL Files to ADS 2011 Compatible Format](#)
2. [Upgrading PDK to ADS 2011 Compatible Format](#)

#### Preparing AEL Files to ADS 2011 Compatible Format

Run aelcheck.exe (located in <ADS\_INSTALLATION\_DIRECTORY>/bin) from the OS command prompt on all the PDK AEL files. Make changes in the ael code according to the

comments that `aelcheck.exe` adds to those AEL files. These changes can be made in a way that the PDK is compatible with both ADS 2011 and prior ADS releases. (The database of this tool is updated periodically and the updated version is provided to you in this file if needed).

 `aelcheck` is a tool that parses the `ael` files and identifies AEL functions that need modification to achieve ADS 2011 platform compatibility.

Each AEL file processed by the `aelcheck` tool will have comment lines beginning with `///**ADS2011: ...` for each function that requires further modification, along with tips and suggestions on how to edit the AEL code to make compatible with ADS 2011. The `aelcheck` tool also generates a backup of the original `ael` file that is saved with the `.bak` extension. For example, the original `my_XYZ.ael` file is saved as `my_XYZ.ael.bak` in the same directory where the `my_XYZ.ael` file is located.

The comments inserted by the `aelcheck` tool contain one of the following headers:

- **Deleted/Classic Only** - Only available in ADS 2009 Update 1 and prior releases of ADS. Use the alternative suggested function, but note that alternative functions are generally not compatible with ADS releases prior to ADS 2011.
- **Deprecated** - The AEL function is no longer recommended. You should consider using the alternative suggested function in new code, but note that alternative functions are generally not compatible with ADS releases prior to ADS 2011.
- **Unknown** - Contact us for instructions or get the latest status of these functions.
- **PDEOA\_TODO** - Contact us for instructions or get the latest status of these functions.

**Invoking aelcheck Tool**

To invoke `aelcheck` tool type the following command in the command line window:

```
aelcheck [options] input_file.ael
```

The following table describes the `aelcheck` command line options:

Option	Description
-p	Primary mode, tests input for any <i>Deleted, Classic Only</i> compatibility issues.
-s	Secondary mode, tests input for any <i>Deprecated</i> compatibility issues.
-d	Developer mode, tests input for any <i>PDEOA_TODO, Unknown</i> compatibility issues.
-n	Normal mode, this is the default mode, tests the Primary mode issues and <i>Unknown</i> issues.
-a	All modes, test input for all primary + secondary + developer mode issues.
-c	Clear: No modes. Useful for clearing comments from a file.
-b	Batch mode. Scan all <code>ael</code> files in subdirectory.
-t	Tag files: Add an ADS 2011 status comment to top of line of all scanned files as either passed or failed.
-l	Log results: logs the results into an <code>aelcheck_results.log</code> file in the current directory.

**Usage Examples:**

```
aelcheck input_file.ael
aelcheck -a input_file.ael
aelcheck -p -d input_file.ael
aelcheck -pd input_file.ael
aelcheck -b pdk_subdir
```

Use `-p` to tell the `aelcheck` tool to spare the UNKNOWN and PDEOA\_TODO comments. Note that the functions identified with these comments are under development, and should eventually be marked with pertinent compatibility tips with the newer updates of `aelcheck` tool.

If you do not want to review the deprecated functions then it is recommended to use the `-pd` option.

 **Note**  
 The `aelcheck` tool might generate false errors if your `ael` code includes quotes and a word that is also a function name. For example, if your code contains "info", the `aelcheck` will prompt a hint asking you to use `de_info` instead of `info`. This is because the `aelcheck` tool can not differentiate between a call to a function and a simple string (both in quotes) if they have the same name.

**EXAMPLE 1: Get design rep. from design name**

**Original ADS Code:**

```
...
decl design_name = de_current_design_name();
decl designH = db_get_design(design_name);
decl design_rep = db_get_rep(designH,REP_SCHEM);
...
```

**Modified ADS Code (compatible with ADS 2011 and prior releases):**

```

If(is_pdeoa_mode()) // ADS being used is ADS 2011
{
...
decl context = de_get_current_design_context();
if (!de_is_schematic_context(context))
return; // or error out if pre-condition is there to be a schematic design
...
}
else //For ADS releases prior to ADS 2011
{
...
decl design_name = de_current_design_name();
decl designH = db_get_design(design_name);
decl design_rep = db_get_rep(designH,REP_SCHEM);
...
}

```

**EXAMPLE 2: Traversing instances of a design to get first selected instance.****Original ADS Code:**

```

...
decl design_name=de_current_design_name();
decl design_rep=db_get_rep(db_get_design(design_name),REP_SCHEM);

// Get first selected instance.
decl insthandle=db_first_instance(design_rep);
while (insthandle&&(db_get_instance_attribute(insthandle,INST_SELECT )==0))
{
insthandle=db_next_instance(insthandle);
}
if (!insthandle)
return FALSE; // No selected instances available.
...

```

**Modified ADS Code (compatible with ADS 2011 and prior releases):**

```

If(is_pdeoa_mode()) // ADS being used is ADS 2011
{
...
decl context = de_get_current_design_context();
if (!de_is_schematic_context(context))
return FALSE;

decl instIter = db_create_inst_iter(context);
// Limit instance iterator to only selected instances.
instIter = db_inst_iter_limit_selected(instIter);
if (!db_inst_iter_is_valid(instIter))
return FALSE; // No selected instances available.

// Get first selected instance.
decl insthandle = db_inst_iter_get_instance(institer);
...
}
else //ADS version being used is prior to ADS2011
{
...
decl design_name=de_current_design_name();
decl design_rep=db_get_rep(db_get_design(design_name),REP_SCHEM);

// Get first selected instance.
decl insthandle=db_first_instance(design_rep);
while (insthandle&&(db_get_instance_attribute(insthandle,INST_SELECT )==0))
{
insthandle=db_next_instance(insthandle);
}
if (!insthandle)
return FALSE; // No selected instances available.
...
}

```

**Upgrading PDK to ADS 2011 Compatible Format**

**Note**  
PDKs translated in this 2nd step using Alpha/EA releases of ADS 2011 will need to be retranslated using the final ADS 2011 release to ensure that they are converted using the final ADS 2011 code.

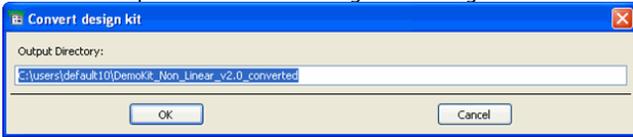
Following are the steps to convert a PDK:

1. Click **Tools->Command Line...** from the ADS Main window.

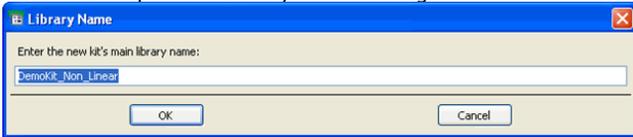
2. Type the command `de_convert_kit()`.
3. Click **Apply** the Information Message box is displayed. (If you have a workspace open, you'll need to close your open workspace before converting a PDK)



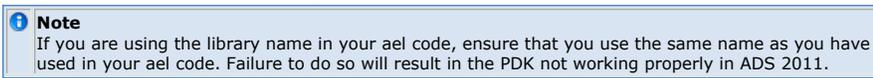
4. Click **OK**.
5. Choose the PDK you want to convert.
6. Click **OK** to open the *Convert design kit* dialog box.



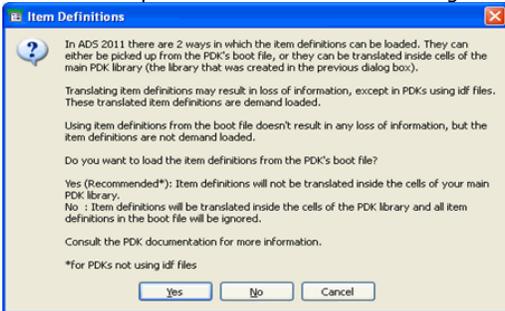
7. Enter the name and path where you want the PDK to be converted.
8. Click **OK** to open the *Library Name* dialog box.



9. Enter the new kit's main library name.



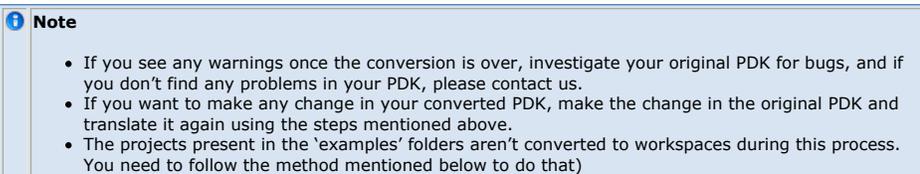
10. Click **OK** to Open the *Item Definitions* dialog box.



11. Choose the option as required.
  - If you choose **Yes** and load the item definitions from the boot file as in previous versions of ADS (RECOMMENDED):
    - Advantage – No loss of information
    - Disadvantage – No demand loading of components
  - If you choose **No** to write the item definitions in the cells of the PDK library:
    - Advantage – The components are demand loaded.
    - Disadvantage - You may lose information because only the forms, formsets and item definitions get translated in the cells. But if your PDK/VCL uses idf files, you can use this option without loss of information. This option only affects the item definitions in the PDK, by deciding whether these item definitions are loaded from the boot file or from the component cells (through itemdef.ael).
12. On successful conversion, the *Design Kit Conversion Succeeded* message is displayed.



The PDK has now been upgraded to work with ADS2011 Format.

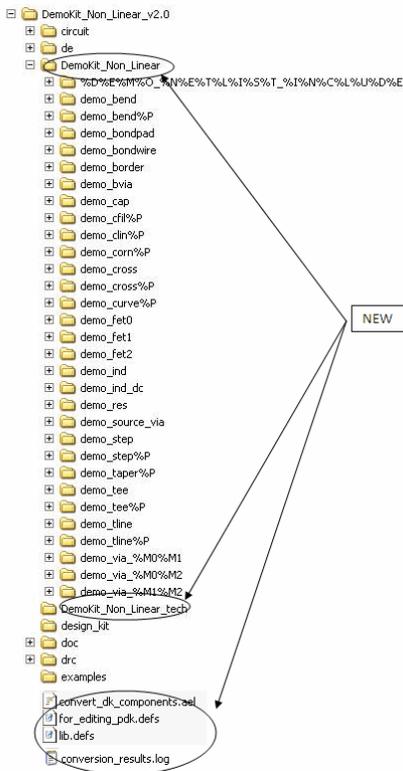


**Original PDK:**

Original PDK:



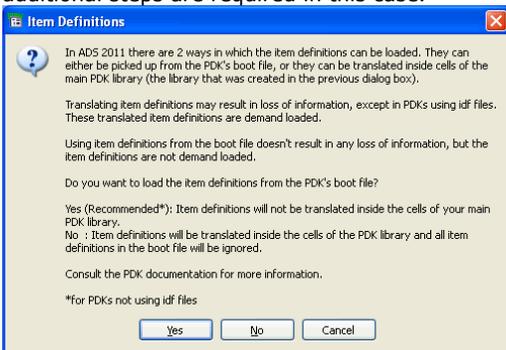
Converted PDK:



## Conversion of Components Created using Model Composer and Advanced Model Composer

### Model Composer Components

- You can recognize the component as the *net\_data* (netlist value) field in the *create\_item* is **pmlg\_device**.
- During the PDK conversion process, click **Yes** in the *Item definitions* dialog box. No additional steps are required in this case.



- If however, you click **No** in the *Item definitions* dialog box, you need to fix the 'resolved path' conversion error that you received while conversion of the PDK, in the translated item definition. Simply replace the netlist format string in the translated item definition (itemdef.ael in the cell) with the original variable 'NET\_pmlg\_<itemName>'

### Example to Change create\_item()

```
create_item("bend_m1","Bend, arbitrary angle","bend_m1_",0,-
1,NULL,"Component Parameters","*","%d:%t %# %44?0%:%31?%C%:_net%c%;%;%e
file=\"C:/users/default11/Demokit_v3_converted //circuit/data/bend_m1.pml\"")
```

```
%0b%29?%:d0=%p%;%1e", "pmlg_device", ...
to
create_item("bend_m1", "Bend, arbitrary angle", "bend_m1_", 0, -
1, NULL, "Component Parameters", "*", NET_pmlg_bend_m1, "pmlg_device", ...
```

### Advanced Model Composer Components

- You can recognize the component as the *net\_data* (netlist value) field in the create\_item is **MomCmpt**.
- During the PDK conversion process, it is recommended to click **Yes** in the *Item definitions* dialog box. However before doing that, there are a few changes required in the create\_item:
  1. itemLabel - itemName ('Layout Component' isn't needed any more...)
  2. attribute - 0
  3. netlistFormat - standard\_netlist
  4. netlistData - itemName
  5. displayFormat - standard\_symbol
  6. Remove callback: list(dm\_create\_cb(ITEM\_NETLIST\_CB...)),

Moreover, if you want to retain backward compatibility with ADS2009U1, use conditional statements to load both attributes in the item definitions.

For example, If the original create\_item is:

```
create_item
(
  "Demokit_tlin_m1", // itemName
  "Layout Component - tlin_m1", // itemLabel
  .
  .
  .
```

Create a variable tlin\_m1\_itemlabel:

```
decl tlin_m1_itemlabel=(is_pdeoa_mode()) ? "tlin_m1" : "Layout Component - tlin_m1";
```

and then change the create\_item to:

```
create_item
(
  "Demokit_tlin_m1", // itemName
  tlin_m1_itemlabel, // itemLabel
  .
  .
  .
```

- If however, you click **No** from the *Item Definitions* dialog box, then no additional steps are required.

**Note**

- If all components in your PDK is created using **Model Composer**, click **Yes** from the *Item Definitions* dialog box during PDK conversion.
- If all components in your PDK is created using **Advanced Model Composer**, click **No** from the *Item Definitions* dialog box during PDK conversion.

### What's New in the Upgraded PDK

- **PDK's main library** – The default PDK library name is same as the PDK (mentioned in *ads.lib*) for example, in the above figure, the PDK library is named as *DemoKit\_Non\_Linear* and contains the cells for all the PDK components. These cells further contain views (schematic, layout, and symbol) depending on whether the component is schematic only, layout only, schematic and layout both. The *eesof\_lib.cfg* is the only file in the PDK's main library that is edited using a text editor if required. The remaining files are edited through ADS only, if needed.
- **PDK's Technology library** – The default name for PDK Technology library is <PDK name>\_tech. For example, in the above figure, the PDK Technology library is named as *DemoKit\_Non\_Linear\_tech* and stores the information related to layers, preferences, and substrates. The files in this library are edited only through ADS.
- **convert\_dk\_components.ael** – Includes a log of all the functions that were run during the PDK conversion.
- **lib.defs** – The library definitions file that should be pointed to, if the user wants to add a PDK to the workspace in read-only mode. This should not be edited in a text editor. This can however be edited using the 'DesignKits' menu option.
- **for\_editing\_pdk.defs** – Adds a PDK to a workspace in editable mode. This is for PDK vendors to edit the PDK in ADS after conversion, if required and must be deleted before sending the PDK to customers. This should not be edited in a text editor. This can however be edited using the 'DesignKits' menu option.

- **conversion\_results.log** – Includes log of warnings or errors that occurred during PDK conversion. You must resolve these errors or warnings and then reconvert the PDK.
- **eesof\_lib.cfg** – This configuration file is present in the PDK's main library. It makes additional data accessible when a library is loaded. For example, it can specify an AEL file to be loaded every time the library is loaded, specify directories to be added to the simulator's search path when loaded etc. This file is edited using a text editor, if required.
- **itemdef.ael** – This file gets created in the component cells in the converted PDK, if you choose *No* in the *Item Definitions* dialog box that is displayed during the PDK conversion. This contains the translated item definitions which are then demand loaded. This file should never be edited by the user.
- **\*.subst files** – The \*.subst files are individual substrates that are created and edited from the Technology dialog 3D/EM Substrates tab, Substrates sub-tab. If your PDK has a .slm file in circuit > substrates folder, that also will be converted to a .subst file during the PDK conversion process.
- **materials.matdb** – The *materials.matdb* file contains the material properties that are entered in the Technology dialog on the 3D/EM Substrates tab, *Materials* sub-tab, and *Conductors, Dielectrics, and Semiconductors* tabs. This file is present in the PDK's technology library  
If materials and substrates are in a pdk library and that library is referenced by another library using the Technology dialog Type tab, those materials and substrates are available for EM simulations of layouts of the other library. This is much like layers and units of a referenced library are available to another library.
- **LayerMap.map** – This file contains the name of the Layer, Purpose, GDSLayer, and GDSPurpose. This file extracts information from the original .lay file of the kit. As the GDS information is no longer maintained in the technology dialog box, this is where the user can get it from, using the function *de\_get\_gds\_number()*. This file is present in the PDK's technology library.

## PDK Vocabularies in ADS 2011

### Important

PDKs cannot be successfully upgraded by the end users of the PDKs. The upgrade process requires that all source be present and frequently requires modifications to the PDK before conversion is successful. Contact your PDK vendor to inquire about upgraded PDKs for ADS 2011.

Every PDK's AEL in ADS 2011 will load into a separate vocabulary (which will have the default name as the PDK's main library name). This allows all PDKs to be truly independent, and it also allows proper support of loading and unloading kits. These vocabularies are active and especially important during:

- PDK Boot
- Loading item definitions
- Netlist callbacks
- Lapi callbacks

So if a file in the kit needs to be loaded from the command line, the *load()* function should be provided in the kit's vocabulary as the 2nd argument. For example, previously if you were using *load("C:\PDK1\file1.ael")* in the command line, this should now be changed to *load("C:\PDK1\file1.ael", "Vocab1")*, where *Vocab1* is the vocabulary in which you want to load the file. If *file1.ael* belongs to a PDK, this should be the name of that PDK's vocabulary.

On the other hand, files loaded, say in the boot file of the kit, shouldn't be loaded in any other vocabulary such as *CmdOp* or *SimCmd*. For example, if your PDK's boot file used *load("myfile", "CmdOp");*, this should now be changed to *load("myfile");*  
You should also remove any references to *#voc* in your PDK.

If you have a load statement inside a function that may be called later, this will still be OK in most cases, but if you use something like:

```
defun myFunction()
{
    load("ael");
}
```

then it is possible that you may need a vocabulary specified in the load statement.

The major side effect of this change is that PDKs will not be able to see code defined inside other PDKs. This is almost always a good thing, but if two kits are designed to interact, then special steps must be taken.

You can have a kit use the same vocabulary as another kit by adding the variable *AEL\_VOCABULARY* in the *eesof\_lib.cfg* file of the 2nd kit. For example, if you want PDK2 to use the same vocabulary as PDK1, in the *eesof\_lib.cfg* file of PDK2's main library, add the line *AEL\_VOCABULARY=PDK2* (where PDK2 is assumed to be the name of PDK2's vocabulary).

## Summary

- All design kit ael now loads into a unique vocabulary with the default name equal to the library name.
- You can control the vocabulary that a library uses by putting `AEL_VOCABULARY=vocabularyName` into the `eesof_lib.cfg` file of that library.
- The default parent vocabulary of the design kit vocabulary is `SimCmd`.
- The default parent vocabulary of `SimCmd` is `CmdOp`.
- You can control the parent vocabulary of the design kit vocabulary by putting `AEL_PARENT_VOCABULARY=vocabularyName` into the `eesof_lib.cfg` file of the design kit.

## Backward Compatibility in PDKs

**Important**  
PDKs cannot be successfully upgraded by the end users of the PDKs. The upgrade process requires that all source be present and frequently requires modifications to the PDK before conversion is successful. Contact your PDK vendor to inquire about upgraded PDKs for ADS 2011.

To know how to make your code backward compatible in general, see *Examples* (oqkref)

## AEL Functions

To make your ADS 2011 PDKs backwards compatible, follow the steps mentioned below. These will allow you to use new functions introduced in ADS 2011 in previous releases as well.

1. Unzip and put the `compat.zip` located under `<ADS_Installation_Directory>/design_kit/pdk_backward_compatibility_files` directory in the following location:

```
"/de/ael/compat"
```

So the `dk_compatibility.atf` and `dkcompatLoadFile***.atf` files are all found under this directory:

```
"/de/ael/compat/dk_compatibility.atf"
"/de/ael/compat/dkcompatLoadFile_350.atf"
"/de/ael/compat/dkcompatLoadFile_370.atf"
"/de/ael/compat/dkcompatLoadFile_COMMON.atf"
```

Modify the design kit's `boot.ael` file to have the following load line:

```
// designKitRecord[1] holds path to the design kit.
load( strcat(designKitRecord[1], "/de/ael/compat/dk_compatibility") );
```

Now the design kit can use the following functions, in releases prior to when they are actually available:

- `is_pdeoa_mode()` – Was originally made available in ADS 2009.
- `dkcompat_flatten_layout_into_current()` – Available currently only via this compatibility method at this time.
- `netlist_instance_cb()` – Was originally made available in ADS 2011. A netlist format string is preferred for use over this function since the format string syntax is the most backwards compatible way of netlisting components in all ADS versions and is most efficient. This function is limited in what is netlisted, it won't netlist repeated parameters and assumes that parameter values have a left and right side `<parameter name> = <param value>`. The type of parameter forms is somewhat limited in the compatibility version of this function. It has the same limitation on what parameter values can be properly netlisted as `generic_netlist_cb()` had.
- `de_is_converting_dsn_pdk()` – A backward compatibility function that gets loaded into ADS 2009 Update 1 or prior releases, and could be used in PDK ael code that is expected to run in both ADS 2009 Update 1 or prior releases as well as in ADS 2011. This function was first implemented in ADS 2011.01 to support a special need for converting two PDKs one of which was primary and the other secondary, and that the secondary PDK conversion had a dependency on a certain AEL in the primary one. This function could be called from the secondary PDK's ael where it relied on this function to detect that the primary PDK to be available, and therefore retrieve the necessary information from the primary PDK. This function always returns FALSE when used in ADS 2009 Update 1 or prior releases.

**Note**  
The files in `compat.zip` will be updated periodically as and when we add more functions for backward compatibility. For more information on the AEL backward compatibility, refer the document provided with the `compat.zip` file.

## User Compiled Models

Since the compilers have been upgraded for ADS2011, to use the same PDK in ADS 2011 and a prior release, follow this folder structure when placing your User Compiled Models in the PDK:

```
designkit1/bin/win32 (for ADS 2009 and before)
designkit1/bin/win32_VS2008 (for ADS 2011)
designkit1/bin/win32_VS2010 (for next compiler upgrade)
designkit1/bin/win32_64 (for ADS 2009 and before)
designkit1/bin/win32_64_VS2008 (for ADS 2011)
designkit1/bin/win32_64_VS2010 (for next compiler upgrade)
designkit1/bin/linux_x86 (for ADS 2009 and before)
designkit1/bin/linux_x86_GCC442 (for ADS 2011)
designkit1/bin/linux_x86_GCC462 (for future compiler upgrade)
designkit1/bin/linux_x86_64 (for ADS 2009 and before)
designkit1/bin/linux_x86_64_GCC442 (for ADS 2011)
designkit1/bin/linux_x86_64_GCC462 (for future compiler upgrade)
designkit1/bin/sun_sparc_64 (for ADS 2009 and before)
designkit1/bin/sun64_sparc_SS12 (for ADS 2011)
designkit1/bin/sun64_sparc_SS13 (for future compiler upgrade)
```

To generalize, the format is `designkit1/bin/$SIMARCH$COMPILER_VER`

- On the Windows platform, `COMPILER_VER` is set to `_VS2008`
- On the Linux platform, `COMPILER_VER` is set to `_GCC442`
- On the Solaris platform, `COMPILER_VER` is set to `_SS12`

## Working with Upgraded PDKs in ADS 2011

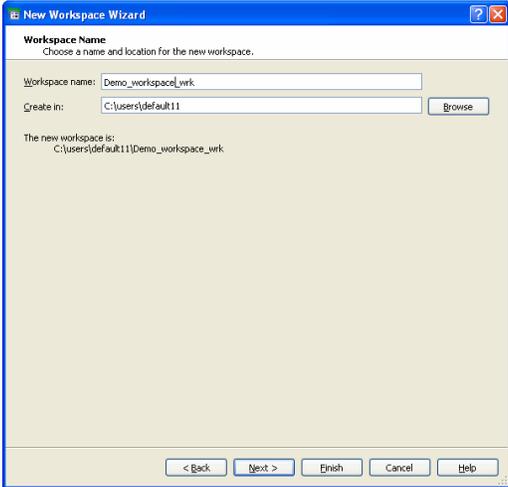
### About the Upgraded PDKs

- PDKs are no longer installed or uninstalled as they were in prior ADS releases. Just like libraries, they are added or removed from the workspace. Hence, PDKs are used at the workspace level in ADS 2011. They do not have scoping at project/user/site levels – they are managed like other libraries.
- Every library (and thus a PDK) has a unique namespace in ADS 2011. So if you create a component named *R* in your PDK, it will not clash with the ADS in-built component *R*.
- While converting a project to a workspace, if you don't include the converted PDK that is used by that project, your designs in the workspace will not work properly. So first convert your PDK, and then convert your project which uses that PDK.
- In ADS 2011, strings must be explicitly written in quotes (using `\`).
- The *x*, *y*, *winInst*, *winInstP* and *dlgH* variables were defined globally in prior ADS releases. These global declarations are removed in ADS 2011. So you must explicitly declare these variables if your PDK code is using them.
- Existing User Compiled Models does not work in ADS 2011, these UCMs must be recompiled for each platform using the compiler officially supported by ADS 2011.

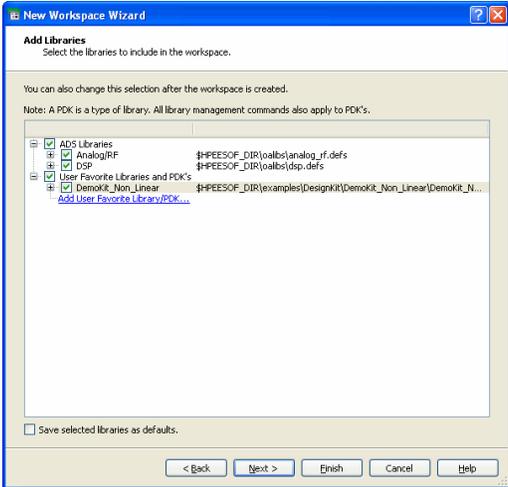
### Adding a PDK to a Workspace in ADS 2011

A PDK is added in a new workspace by including its library just like any other library. Following are steps to add a PDK in a new workspace:

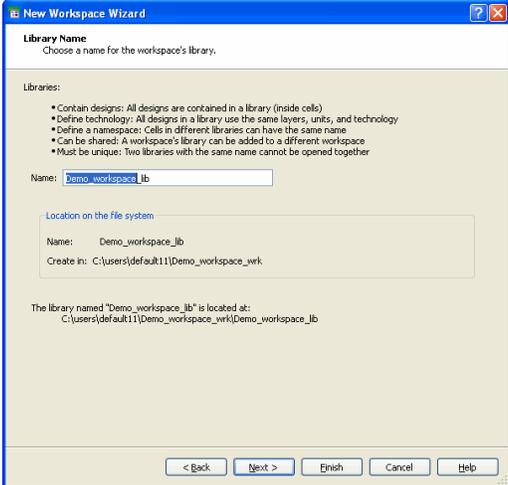
1. Click **File > New > Workspace...** from the ADS Main window to start the *New Workspace Wizard*.
2. Click **Next** from the *Introduction* page to open the *Workspace Name* page.
3. Enter the new name in the *Workspace name* field.



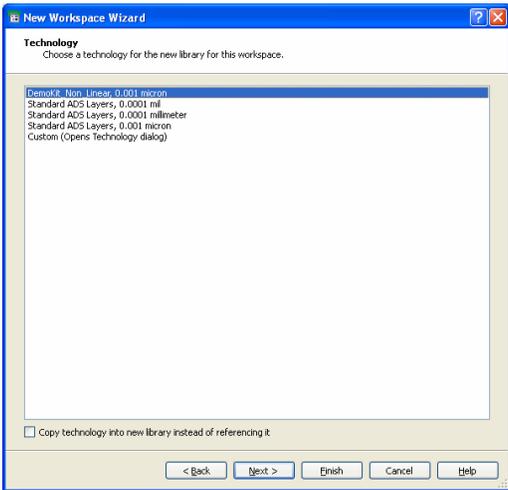
4. Click **Next** to open the *Add Libraries* page.
5. Click **Add User Favorite Library/PDK...** and select the *lib.defs* or the *for\_editing\_pdk.defs* file of the PDK you want to add in the workspace (depending on whether you want to add it in Read Only mode or Editable mode). The PDK is added and checked in the User Favorite Libraries and in PDK's list.



6. Enter the workspace library name in the *Name* field.



7. Click **Next** to open the *Technology* page.



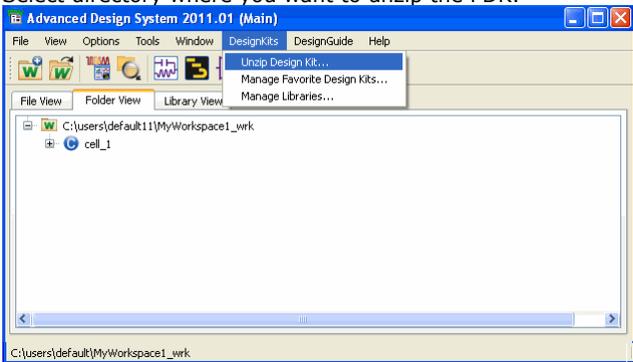
8. Click **Finish** a new workspace is created with your PDK added to it.

However, if you want to add a PDK to an existing workspace, open the workspace and perform one of the following methods.

**For zipped PDKs:** PDKs are added in read-only mode.

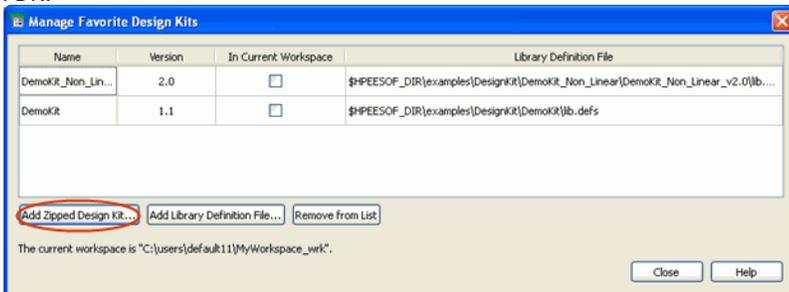
**Method 1**

1. Click **DesignKits > Unzip Design Kit...** from the ADS Main window and browse to the zip file.
2. Select directory where you want to unzip the PDK.



**Method 2**

1. Click **DesignKits > Manage Favorite Design Kits** from the ADS Main window and browse to the zip file.
2. Click **Add Zipped Design Kit....** Then select directory where you want to unzip the PDK.

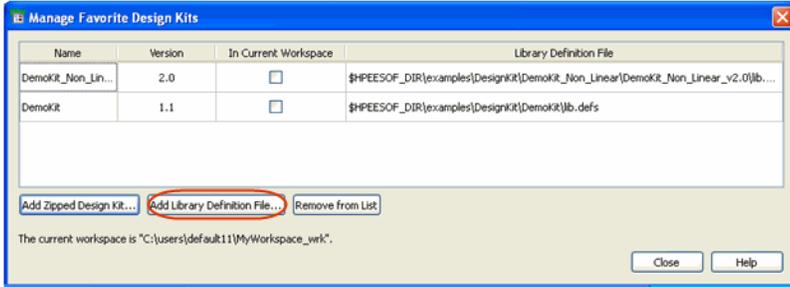


**For unzipped PDKs:**

**Method 1**

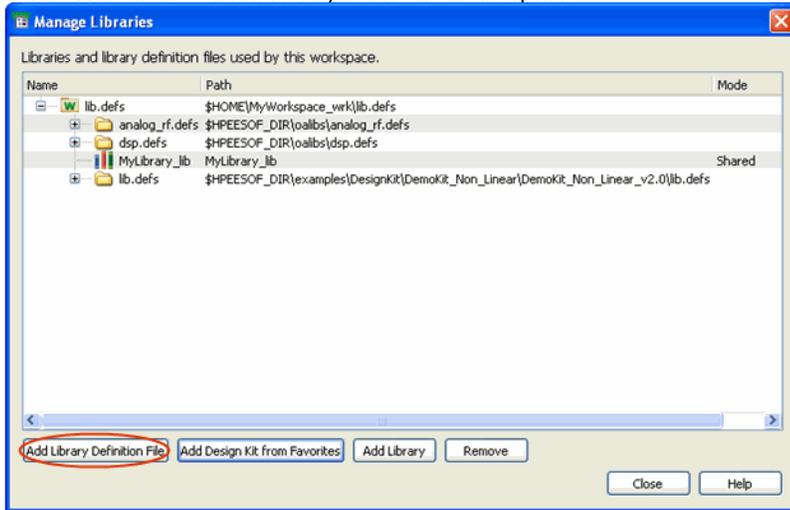
1. Click **DesignKits > Manage Favorite Design Kits** from the ADS Main window.
2. Click **Add Library Definition File** and select the *lib.defs* file or the *for\_editing\_pdk.defs* file of the PDK you want to add in the workspace (depending on whether you want to add it in Read Only mode or Editable mode respectively). You will receive a prompt if you want to add this PDK to the current workspace.

3. Click **Yes**. The PDK is added to your current workspace.



**Method 2**

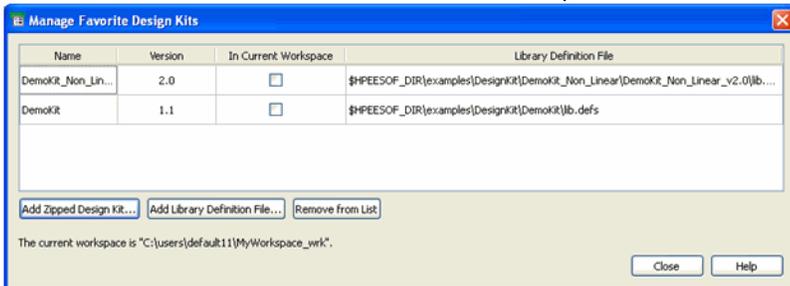
1. Click **DesignKits > Manage Libraries...** from the ADS Main window.
2. Click **Add Library Definition File** to select the *lib.defs* file or the *for\_editing\_pdk.defs* file of the PDK you want to add in the workspace (depending on whether you want to add it in Read Only mode or Editable mode respectively). You will receive a prompt if you want to add this PDK to the current workspace.
3. Click **Yes**. The PDK is added to your current workspace.



**Removing a PDK from a Workspace in ADS 2011**

**Method 1**

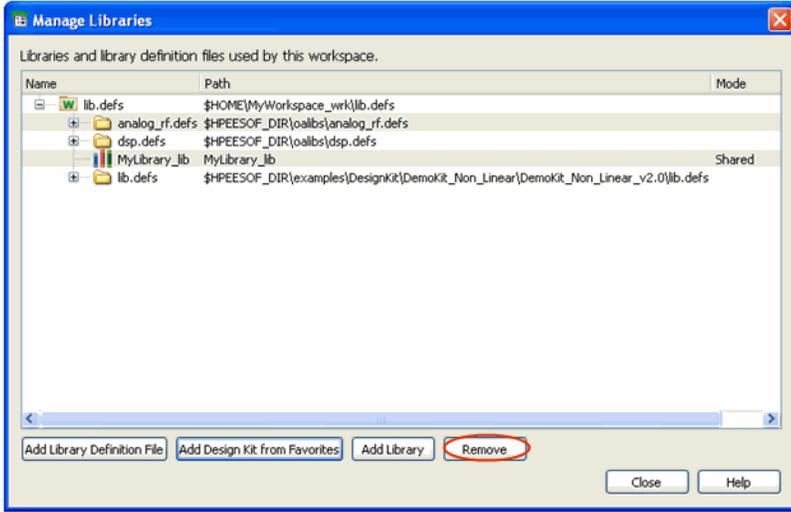
1. Click **DesignKits > Manage Favorite Design Kits** from the ADS Main window.
2. Uncheck the *In Current Workspace* field for that PDK if the PDK you want to remove is in the list. A confirmation dialog box is displayed.
3. Click **Yes**. The PDK is removed from the current workspace.



**Note**  
The **Remove from List** button does not remove a PDK from the current workspace, but removes it from the Favorite PDKs list.

**Method 2**

1. Click **DesignKits > Manage Libraries...** from the ADS Main window.
2. Select the PDK that you want to remove from the *Manage Libraries* dialog box .
3. Click **Remove**. You will receive a confirmation dialog box.
4. Click **Yes**. The PDK is removed from the current workspace.



### Procedure to Convert Projects to Workspaces

For detailed information on the procedure to convert Projects to Workspaces in ADS 2011, see *ADS Project Upgrade to ADS Workspace* (oaqkref)

### Frequently Asked Questions (FAQs)

This section includes Frequently Asked Questions (FAQs) on Upgrading PDKs. The FAQs are divided into the following categories:

- [General FAQs](#)
- [PDK Conversion FAQs](#)
- [Using PDKs in ADS 2011 FAQs](#)

**Important**  
 PDKs cannot be successfully upgraded by the end users of the PDKs. The upgrade process requires that all source be present and frequently requires modifications to the PDK before conversion is successful. Contact your PDK vendor to inquire about upgraded PDKs for ADS 2011.

### General FAQs

Question	Answer
Can I use my existing PDKs in ADS 2011 without any modifications?	No, existing PDKs (created up until ADS 2009U1) must be upgraded to be ADS 2011 compatible format. For information on PDK Upgrade, see <i>ADS 2011 PDK Upgrade</i> (oaqkref).
Where do I get ADS 2011 compatible PDKs?	Depending on whether you got your PDK from Agilent/Foundry/Vendor, contact the same source for ADS 2011 compatible PDKs. If you have created your own PDK, you need to upgrade it to be ADS 2011 compatible format. For information on PDK Upgrade, see <i>ADS 2011 PDK Upgrade</i> (oaqkref).
Can I create new PDKs in ADS 2011?	No, only upgrading your existing ADS PDKs is supported in this release. For information on PDK Upgrade, see <i>ADS 2011 PDK Upgrade</i> (oaqkref).
Is it required to maintain two sets of PDKs, one for ADS 2011 and the other for older ADS releases?	No, you can upgrade your PDKs in such a way, that the converted PDK works in both ADS 2011 and previous releases. However, if you want this backward compatibility in your PDKs, you may not be able to use the new features such as layer purposes and LayerIDs.
Are PDKs and Libraries the same in ADS 2011?	In ADS 2011, PDKs are types of libraries in that contains one or more libraries (generally the component library and the technology library).
I was informed that I need to recompile my UCMs for ADS 2011 using VS2008 SP1. Would my recompiled UCMs be backward compatible as well?	No, the UCMs compiled using VS2008 SP1 would work with ADS 2011, whereas your original UCMs would also be required in the converted PDK if you want to use the PDK in older ADS releases. For information on where to place the different UCMs in your PDK, see <i>Backward Compatibility in PDKs</i> (oaqkref).

### PDK Conversion FAQs

## Advanced Design System 2011.01 - ADS Quick Reference

Question	Answer
Is there an automated way to upgrade PDKs for ADS 2011?	Yes, there is an automatic conversion wizard, but that can be used only on PDKs, which have ADS 2011 compatible ael code. To ensure that your ael code is ADS 2011 compatible, we strongly recommend running the <i>aelcheck</i> tool before converting your PDKs, as described in <i>Prepare AEL Files</i> (oaqkref). To use the automatic conversion wizard, see <i>Upgrading PDK to ADS 2011 Compatible Format</i> (oaqkref).
Where do I find <i>aelcheck</i> tool to scan my code for ADS2011 compatibility?	This .exe file is located under <ADS_Installation_Directory>/bin folder.
What mode should I run the <i>aelcheck</i> tool?	Refer to <i>Prepare AEL Files</i> (oaqkref), for more information on various modes in which you can run <i>aelcheck.exe</i> .
The <i>aelcheck</i> tool indicates that I need to modify the <i>info()</i> that is used in the <i>ael</i> code, but I have not used this function anywhere. Is this a false error?	Yes, the <i>aelcheck</i> tool might generate false errors if your ael code includes quotes and a word that is also a function name. For example, if your code contains <i>info()</i> , the <i>aelcheck</i> tool prompts a hint asking you to use <i>de_info()</i> instead of <i>info()</i> . This is because the <i>aelcheck</i> tool cannot differentiate between a call to a function and a simple string (both in quotes) if they have the same name. You can ignore this.
Are any changes needed in the item definitions of the components?	Yes, some changes may be needed for components, generally created using Model Composer and Advanced Model Composer. For more information, see <i>Conversion of Components Created using Model Composer and Advanced Model Composer</i> (oaqkref).
With the change in PDK vocabularies in ADS 2011, does the method of loading files inside a PDK also change?	In general, files should be loaded in the PDK's vocabulary, which is also the default vocabulary. Loading files in CmdOp should be done only when loading them in the default vocabulary results in errors/issues.
You have introduced a new function <i>is_pdeoa_mode()</i> in ADS 2011, which you mention is backward compatible with ADS2009U1 as well. How is that possible?	With the help of backward compatibility files, you can use this function in some older ADS releases as well. For information on backward compatibility files, see <i>AEL Functions</i> (oaqkref).
In the <i>Item Definitions</i> dialog box that I get during PDK conversion, what is the general suggested approach?	The general approach is to click <b>No</b> for any kits that load components through <i>idf</i> files, and to click <b>Yes</b> for the remaining kits. For more detailed information, go through the text in the dialog box.
After PDK conversion, I see some errors related to my PDK. These were not being prompted in older versions of ADS. Is this a bug in ADS 2011?	ADS 2011 has enhanced error reporting capabilities, due to which it also captures errors that were missed out by previous versions of ADS. It is strongly recommended to fix all errors/warnings before upgrading the PDK.
Is there a log stored somewhere for errors/warnings during PDK conversion?	Yes, it is stored in the converted PDK with the name <i>conversion_results.log</i> .

### Using PDKs in ADS 2011 FAQs

## Advanced Design System 2011.01 - ADS Quick Reference

Question	Answer
Why does the converted PDK have two .defs files? Which one am I supposed to point to when including that PDK in my workspace?	The <code>lib.defs</code> file adds the PDK in a read only mode to the workspace, so that the user does not make any changes to the PDK. If you want to edit the PDK in ADS 2011, point to the <code>for_editing_pdk.defs</code> file. You should ideally remove this file from the PDK before shipping it to customers, if you do not want the customers to edit the PDK.
In my converted PDK, there is a file named <code>convert_dk_components.ael</code> . Do I need to load it to finish the conversion?	No, this is just a log of the commands that were executed in the conversion process. This is just for your reference.
I converted my project, which used PDK components, to a workspace but forgot to add the PDK during conversion. Now if I add the PDK, my workspace still does not work correctly. Why is that?	You must add the PDK during project conversion, otherwise the resulting workspace will not work properly.
My PDK had a layer with number between 200 and 255. This does not show up correctly in the converted PDK. Do I need to add this layer again?	Layers with nos. 200-255 are reserved in ADS 2011. So any layers in PDKs having these numbers are shifted to 900-955.
I used GCC for creating my artworks, but GCC is not included in ADS 2011. Will my artworks work properly in ADS 2011?	Yes, artworks created using GCC in a previous ADS version would work properly in ADS 2011. However, you can not use GCC to create new artworks in ADS 2011.
How do I add my PDKs in a way that they show up in all workspaces I create, by default, just as in ADS2009?	For information on this, see <i>Adding Site Libraries</i> (adstour).
The lib browser in ADS 2011 lists more components than what are actually there in the kit. Why is that?	The library browser in ADS 2011 lists down all the cells in the libraries. So if you had extra designs/subckts etc in your PDK, which get converted to cells, they will also be listed in the library. The <code>rec/ctl</code> files still provide information about these components to the library browser, and for the extra components the default values are used. We suggest removing any extra files from the PDK that aren't required before conversion.
Some of my preferences are not retained after adding a PDK to a workspace?	In ADS 2011, there are a few preferences, which are library-specific. If these are not found in the library or any referenced library, the workspace preferences are used. For the remaining preferences, the workspace preferences are used despite what value they have in the library.
I do not see my GDSII information in the technology setup in ADS 2011. Where can I find it?	Once your PDK gets converted, a file named <code>LayerMap.map</code> gets created in the PDK's technology library, which contains this mapping information.
The PDK menu in the main ADS window seems to be changed. How do I use the new menu?	To know more about the new menu, see <i>Working with Upgraded PDKs in ADS 2011</i> (oaqkref).
I see new folders in my ADS 2011 PDK. What are these?	The PDKs behave like libraries in ADS 2011. The new folders are the libraries for components (named after the PDK) and technology (named as <code>PDKName_tech</code> ). For more information on the new PDK structure, see <i>Whats New in The Upgraded PDKs</i> (oaqkref).
As ADS 2011 is built on OpenAccess, will ADS 2011 PDKs be interoperable with other EDA tools?	No, Interoperability with other EDA tools is not supported in the current release.

## ADS 2011 DesignGuide Upgrade

This topic describes the procedure to upgrade your existing DesignGuides to the new ADS 2011 environment. It also describes the upgraded DesignGuide structure and troubleshooting tips which will assist you while upgrading DesignGuides in ADS 2011.

### Contents

- *Upgrading DesignGuides for ADS 2011* (oaqkref)
- *Troubleshooting DesignGuide Upgrade* (oaqkref)

## Upgrading DesignGuides for ADS 2011

The following topics describe the procedure to upgrade your existing DesignGuides (ADS 2009 Update 1 or earlier) and the details about the upgraded DesignGuides folder.

- [Before You Begin](#)
- [Procedure to Upgrade DesignGuide to be ADS 2011 Compatible](#)

### Before You Begin

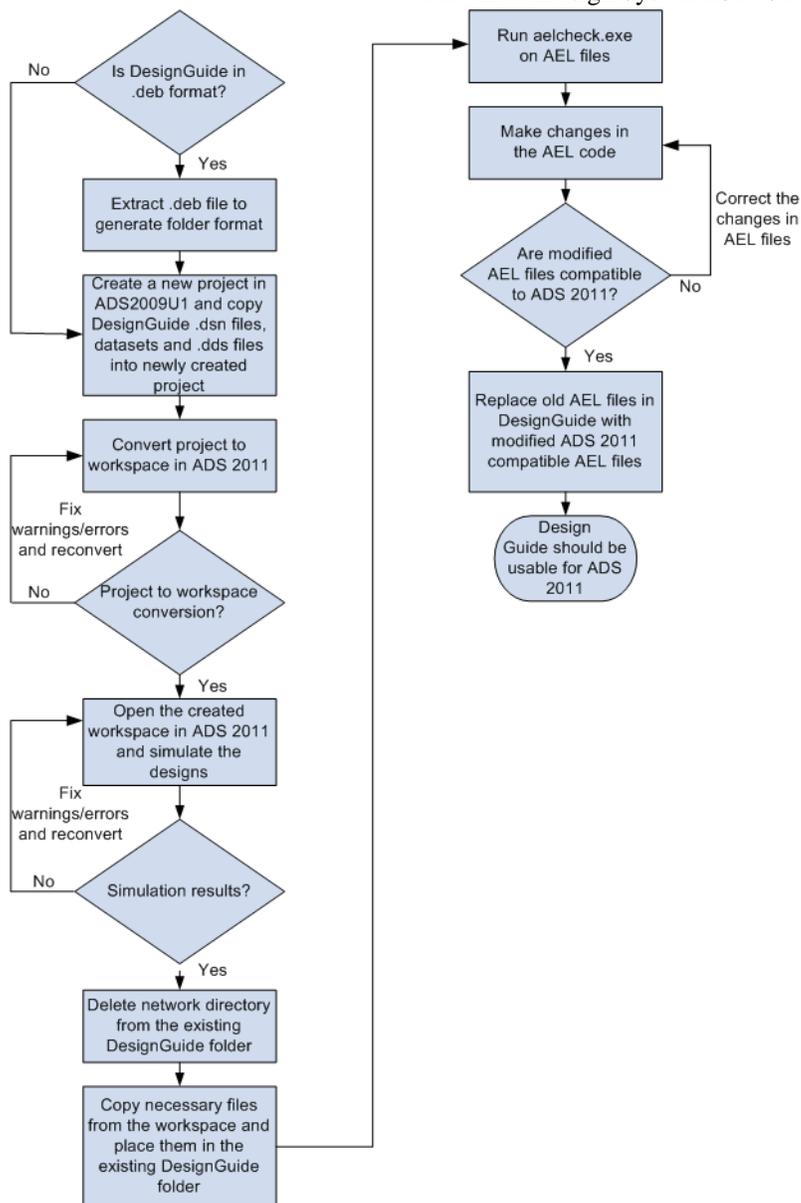
To successfully upgrade your DesignGuides, you must remember the following:

- Before starting the upgrade, you can use the **DesignGuide Developer Studio** in ADS 2011, to generate new ADS 2011 compatible DesignGuides. However, for existing DesignGuides created using DesignGuide Developer Studio on or before ADS 2009 Update 1 must be upgraded manually as shown in [Procedure to Upgrade DesignGuide to be ADS 2011 Compatible](#).
- The DesignGuides created using ADS 2009 Update 1 and earlier releases do not work in their present format in ADS 2011. These DesignGuides must be upgraded to ADS 2011 compatible format.
- There may be changes needed in your DesignGuides AEL files to make them ADS 2011 compatible. This is to support some new features in ADS 2011. Also, as ADS 2011 is a completely new platform, AEL modifications may also be required in your DesignGuides where there are some use-model changes.
- During the DesignGuide conversion process you may find some errors/warnings that had not been reported in the previous versions of ADS release. The ADS 2011 has enhanced error reporting capabilities and it is strongly suggested that you fix these errors/warnings in the original DesignGuide and then convert them for ADS 2011.

### Procedure to Upgrade DesignGuide to be ADS 2011 Compatible

The following flowchart shows the DesignGuide upgrade process.

[DesignGuide Conversion Flowchart](#)



**Note**  
 DesignGuides upgraded using the instructions given in this section must be upgraded again in the final ADS 2011 release.

Follow the steps below to upgrade an ADS 2009 Update 1 or earlier DesignGuide to ADS 2011 compatible format:

1. Extract the existing DesignGuide (ADS 2009 Update 1 or earlier) from project or .deb format to a folder format.
2. Create a new project in ADS 2009 Update 1, after you extract the DesignGuide.
3. Navigate to the ADS 2009 Update 1 DesignGuide folder to be upgraded.
4. Copy the following files from ADS 2009 Update 1 DesignGuide folder to the newly created project:
  - .dsn files in *networks* folder
  - Datasets in the *Data* folder
  - .dds files
5. Now, convert the newly created project to workspace using **File > Convert Project to Workspace** option in the ADS 2011 Main window. When converting the project to workspace, the workspace and library name must be same as the DesignGuide name. To ensure this, remove the *\_wrk* or *\_lib* appended to the workspace and library name, respectively. For example, if the existing DesignGuide name is *Oscillator.dg*, then the workspace and library name will be *Oscillator.dg*. For details on using the Convert Project to Workspace utility, refer to *ADS Project Upgrade to ADS Workspace* (oqkref).

**Note**  
You will need to reconvert the project to workspace after troubleshooting any warnings or error messages which may appear during project to workspace conversion.

- Open the converted workspace in ADS 2011 and simulate the designs, after the conversion is complete. For the simulation process, refer to *Simulating Designs* (adstour).

**Note**  
Troubleshoot any warnings or error messages and re-simulate the designs.

- Delete the *networks* folder from ADS 2009 Update 1 DesignGuide, after the simulation of the designs is complete.
- Copy the following files from the converted workspace to the ADS 2009 Update 1 DesignGuide folder:
  - Library folder (for example, *Oscillator.dg* in Step 5)
  - lib.defs* file
  - New *datasets*
  - .dds* files
- Run *aelcheck.exe* on all the AEL files (present in *ui* and *ael* folders) in the DesignGuide folder.
- Modify the ael code according to the comments added by *aelcheck.exe* to ensure that the AEL files are compatible with ADS 2011. For information on how to modify the AEL files, refer to *Prepare AEL Files to be ADS 2011 Compatible Format*.
- Replace the old AEL and ATF files in the existing DesignGuide folder with the new ADS 2011 compatible AEL and ATF files.

Once the upgrade process is complete, the upgraded DesignGuide is ready to use with ADS 2011.

## Preparing AEL Files to ADS 2011 Compatible Format

Run *aelcheck.exe* (located in `<ADS_INSTALLATION_DIRECTORY>/bin`) from the OS command prompt on all the DesignGuide AEL files. Make changes in the ael code according to the comments that *aelcheck.exe* adds to those AEL files. These changes can be made in a way that the DesignGuide is compatible with both ADS 2011 and prior ADS releases. (The database of this tool is updated periodically and the updated version is provided to you in this file if needed).

**✓** *aelcheck* is a tool that parses the *ael* files and identifies AEL functions that need modification to achieve ADS 2011 platform compatibility.

Each AEL file processed by the *aelcheck* tool will have comment lines beginning with `"/**ADS2011: ..."` for each function that requires further modification, along with tips and suggestions on how to edit the AEL code to make compatible with ADS 2011. The *aelcheck* tool also generates a backup of the original ael file that is saved with the *.bak* extension. For example, the original *my\_XYZ.ael* file is saved as *my\_XYZ.ael.bak* in the same directory where the *my\_XYZ.ael* file is located.

The comments inserted by the *aelcheck* tool contain one of the following headers:

- Deleted/Classic Only** - Only available in ADS 2009 Update 1 and prior releases of ADS. Use the alternative suggested function, but note that alternative functions are generally not compatible with ADS releases prior to ADS 2011.
- Deprecated** - The AEL function is no longer recommended. You should consider using the alternative suggested function in new code, but note that alternative functions are generally not compatible with ADS releases prior to ADS 2011.
- Unknown** - Contact us for instructions or get the latest status of these functions.
- PDEOA\_TODO** - Contact us for instructions or get the latest status of these functions.

### Invoking aelcheck Tool

To invoke *aelcheck* tool type the following command in the command line window:

```
aelcheck [options] input_file.ael
```

The following table describes the *aelcheck* command line options:

Option	Description
-p	Primary mode, tests input for any <i>Deleted</i> , <i>Classic Only</i> compatibility issues.
-s	Secondary mode, tests input for any <i>Deprecated</i> compatibility issues.
-d	Developer mode, tests input for any <i>PDEOA_TODO</i> , <i>Unknown</i> compatibility issues.
-n	Normal mode, this is the default mode, tests the Primary mode issues and <i>Unknown</i> issues.
-a	All modes, test input for all primary + secondary + developer mode issues.
-c	Clear: No modes. Useful for clearing comments from a file.
-b	Batch mode. Scan all ael files in subdirectory.
-t	Tag files: Add an ADS 2011 status comment to top of line of all scanned files as either passed or failed.
-l	Log results: logs the results into an aelcheck_results.log file in the current directory.

### Usage Examples:

```
aelcheck input_file.ael
aelcheck -a input_file.ael
aelcheck -p -d input_file.ael
aelcheck -pd input_file.ael
aelcheck -b dg_subdir
```

Use `-p` to tell the *aelcheck* tool to spare the UNKNOWN and PDEOA\_TODO comments.

Note that the functions identified with these comments are under development, and should eventually be marked with pertinent compatibility tips with the newer updates of *aelcheck* tool.

If you do not want to review the deprecated functions then it is recommended to use the `-pd` option.

**Note**  
The *aelcheck* tool might generate false errors if your ael code includes quotes and a word that is also a function name. For example, if your code contains "info", the *aelcheck* will prompt a hint asking you to use *de\_info* instead of *info*. This is because the *aelcheck* tool can not differentiate between a call to a function and a simple string (both in quotes) if they have the same name.

**EXAMPLE 1: Get design rep. from design name**

### Original ADS Code:

```
...
decl design_name = de_current_design_name();
decl designH = db_get_design(design_name);
decl design_rep = db_get_rep(designH,REP_SCHEM);
...
```

### Modified ADS Code (compatible with ADS 2011 and prior releases):

```
If(is_pdeoa_mode()) // ADS being used is ADS 2011
{
...
decl context = de_get_current_design_context();
if (!de_is_schematic_context(context))
return; // or error out if pre-condition is there to be a schematic design
...
}
else //For ADS releases prior to ADS 2011
{
...
decl design_name = de_current_design_name();
decl designH = db_get_design(design_name);
decl design_rep = db_get_rep(designH,REP_SCHEM);
...
}
```

**EXAMPLE 2: Traversing instances of a design to get first selected instance.**

### Original ADS Code:

```
...
decl design_name=de_current_design_name();
decl design_rep=db_get_rep(db_get_design(design_name),REP_SCHEM);

// Get first selected instance.
decl insthandle=db_first_instance(design_rep);
while (insthandle&&(db_get_instance_attribute(insthandle,INST_SELECT )==0))
{
insthandle=db_next_instance(insthandle);
}
if (!insthandle)
return FALSE; // No selected instances available.
...
```

### Modified ADS Code (compatible with ADS 2011 and prior releases):

```

If(is_pdea_mode()) // ADS being used is ADS 2011
{
...
decl context = de_get_current_design_context();
if (!de_is_schematic_context(context))
    return FALSE;

decl instIter = db_create_inst_iter(context);
// Limit instance iterator to only selected instances.
instIter = db_inst_iter_limit_selected(instIter);
if (!db_inst_iter_is_valid(instIter))
    return FALSE; // No selected instances available.

// Get first selected instance.
decl insthandle = db_inst_iter_get_instance(instIter);
...
}
else //ADS version being used is prior to ADS2011
{
...
decl design_name=de_current_design_name();
decl design_rep=db_get_rep(db_get_design(design_name),REP_SCHEM);

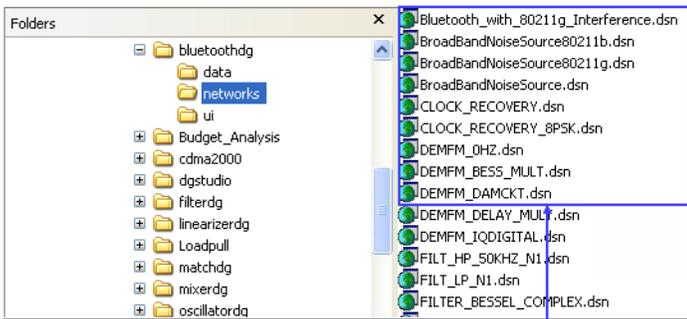
// Get first selected instance.
decl insthandle=db_first_instance(design_rep);
while (insthandle&&(db_get_instance_attribute(insthandle,INST_SELECT )==0))
{
    insthandle=db_next_instance(insthandle);
}
if (!insthandle)
    return FALSE; // No selected instances available.
...
}

```

## Upgraded DesignGuide Folder Structure

The following screenshot shows the DesignGuides folder structure in ADS 2009 Update 1. The *networks* folder displays the .dsn files in ADS 2009 Update 1 or earlier versions.

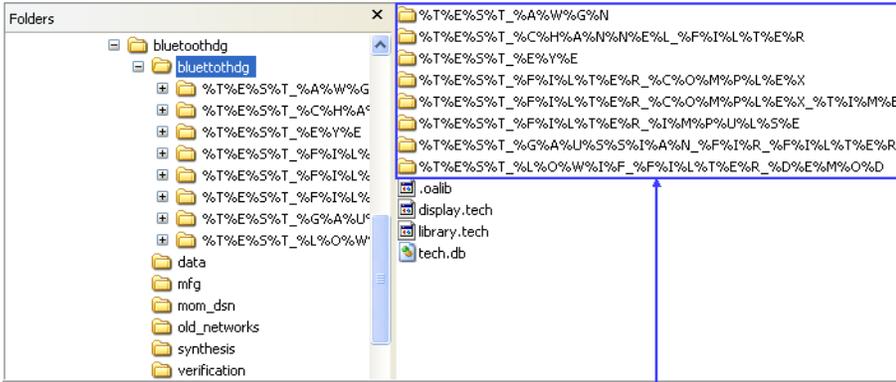
### ADS 2009 Update 1 DesignGuides Folder Structure



The ADS 2009 Update 1 with dsn files in networks folder

The following screenshot shows the upgraded DesignGuides folder structure in ADS 2011. The new folders (for example, bluetoothdgd) are added after the ADS 2011 upgrade. You can find the old .dsn files in *old\_networks* folder.

### ADS 2011 DesignGuides Folder Structure



The upgraded folder structure contains bluetoothdg Library folder with updated schematics

## Troubleshooting DesignGuide Upgrade

This section provides information on warnings and errors encountered during the internal DesignGuides upgrade process.

### Error Messages

**Problem:** Could not find cell *ap\_dio\_MV2103\_19930601* in any library.

**Solution:** Include all the necessary libraries.

**Problem:** Failed to convert symbol *Wizard\_3\_pole\_passive* for item *SYN\_CP\_FQ\_P3P*.

**Solution:** Unable to load the design.

**Problem:** Updated Design Guides not working as expected.

**Solution:** You must delete the contents of **\$HOME/designguide\_files** directory to avoid upgrade issues.

### Warning Messages

**Warning:** finding library for instance 'PFD1' of cell *TEST\_PLL\_TR* in lib 'bluetoothdg\_oalib' [schematic] (conversion continues): Could not find cell "PhaseFreqDet" in any library.

**Solution:** Include all the necessary libraries.

**Warning:** creating new instance 'G3' of 'cell *HB1TonePAE\_Fswp* in lib 'powerampdg\_oalib' [schematic] (conversion continues):

**Solution:** Instance with name 'G3' already exists.

**Warning:** finding library for instance 'ADE\_25MH' of cell *DigitalConverter* in lib 'rfsystem\_oalib' [schematic] (conversion continues):

Could not find cell "MixerIMT" in any library.

**Solution:** Include all the necessary libraries.

**Warning:** creating new instance 'Curve3' in cell *rat\_race\_coupler* in lib 'RADAR\_APPLICATIONS' [schematic] (conversion continues):

**Solution:** Instance does not supports non-90 degree rotation.

**Warning:** translating design 'sub\_clocked\_ADC' to cell 'sub\_clocked\_ADC' in library 'Analog\_Digital'(conversion continues):

**Solution:** Load the design into ADS 2009 Update 1, re-save it and then convert that project into workspace using ADS 2011.

## Glossary and Abbreviations

### ADS

Advanced Design System

### AEL

Application Extension Language (for more information, see *AEL* (ael))

### ASM

Agilent Software Manager

### Cell

A cell is a container for views that define a design for multiple tools. A Cell can have:

- Multiple views for the same tool can be defined (e.g. schematic1, schematic2, ...)
- Multiple symbols can be defined for one cell (e.g. symbol\_hor and symbol\_ver)
- Multiple layouts can be defined for a cell (e.g. layout and em\_layout)
- Should contain views using standard names (schematic, layout, and symbol) unless the cell contains multiple views.

### Library

A Library is a collection of cells. It has a file system directory to represent it.

The physical attributes (layer definitions, units, substrates, etc.) are defined in a "Technology" library. You can create multiple libraries within a complete design hierarchy. Library usage is defined by a file (e.g. lib.defs). A library does not have to physically reside in the workspace directory.

### License File

The license file activates the features of the EEs of products that have been purchased, or are being evaluated. It is a text file with the extension .lic which you receive from Agilent EEs of Business Support. It contains encrypted data that covers information like the features being activated and the duration for those features. Without a valid license file, you cannot use EEs of EDA products.

If license file is mailed as an attachment to an e-mail. You can also download it from the Agilent Software Manager(ASM) using information provided in the entitlement certificate mailed to you.

### OpenAccess

OpenAccess is a community effort to provide true interoperability, not just data exchange, among IC design tools through an open standard data API and reference database supporting that API for IC design. The OpenAccess Coalition is a neutral organization of industry leaders that are leading this effort operating under Si2 bylaws. It provides a single database format that can be used by multiple tools to allow for greater tool interoperability. The primary goal is to eliminate the need for translators between tools from different vendors

The OpenAccess API is a C++ program interface to IC design data. The API and reference implementation provide a high performance, high capacity electronic design database with architecture designed for easy integration and fast application development.

For more information, visit [www.si2.org](http://www.si2.org) .

### OpenAccess Database

ADS 2011 version and onwards is using OpenAccess database where Designs must be stored in a library. The Libraries have one technology associated to it and Design data is broken into multiple storage points. Each view of a "design" is stored separately. There is no limitation on the type or number of views that can be stored for a design. The "design" is defined by the library/cell/view database designation, not the item definition.

### SiP

System in Package

## Technology

A Technology is associated with a library either by being contained within the library or attached to the library. All designs in a library use the same technology. The name of each technology must be unique. One technology can specify some layers and the next technology can specify more layers, as long as they are unique.

OpenAccess also has the concept of Incremental Technology Databases. When an oaTech database contains technology it can also reference one or more other technologies. It inherits the information each other technology. The sets of information are merged into a single set of information as long as the sets of information do not conflict with each other.

The oaTech database contains layers ( oaLayer ). oaLayers are uniquely identified by their number and name. The layers are sub classed into 3 other types.

1. The physical layer ( oaPhysicalLayer ) also has a material ( oaMaterial ) , as mask number ( oaUInt4 ), and optionally a Manufacturing Grid. A default Manufacturing Grid is specified on the oaTech, but can be overridden for each oaPhysicalLayer.
2. The derived layer ( oaDerivedLayer ) is formed from one or two layers and a derived layer definition ( oaDerivedLayerDef).
3. The functionality provided by ( oaSizedLayer ) is now available using oaDerivedLayer. New applications should use oaDerivedLayers to represent sizing operations rather than use oaSizedLayer.

Technology databases are normally provided by a single maintainer and used by individuals as read-only databases.

## UCM

User Compiled Models

## View

A View is a specific representation of a cell. Each view is a container that can store a file or a database object. Different view types include:

1. Schematics
2. Layouts
3. EM Setup
4. Symbols
5. EM Model

## Workspace

ADS Workspace is equivalent to ADS Projects of previous ADS releases (ADS 2009 Update 1 and earlier).

A Workspace contains – or references – one or more libraries (design libraries (user data) and reference libraries (foundry components)). These libraries may be shared. The workspace contains a lib.defs file which manages the libraries of a workspace. A Library contains Cells which are similar to Designs of ADS 2009 update 1 and previous releases. A cell is a container for Views and can have as many Views as required. A View is a specific representation of a cell, e.g., Schematic View, Layout View, and Symbol View. For more information, see *Workspace* (oaqkref).

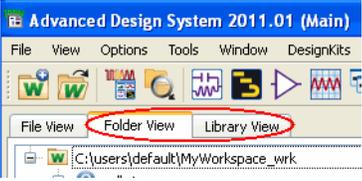
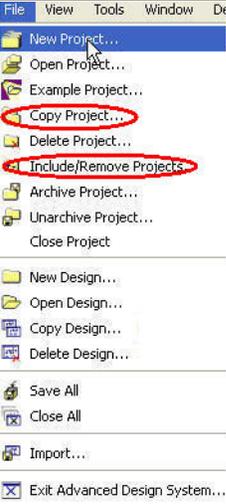
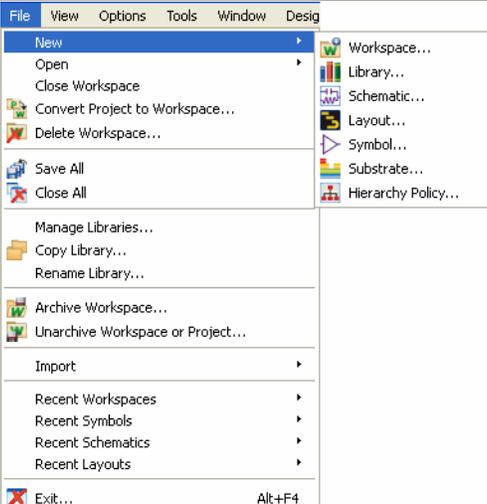
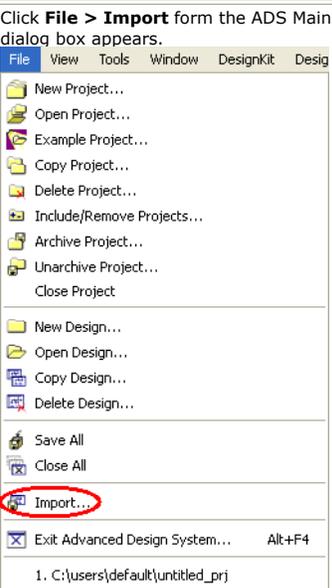
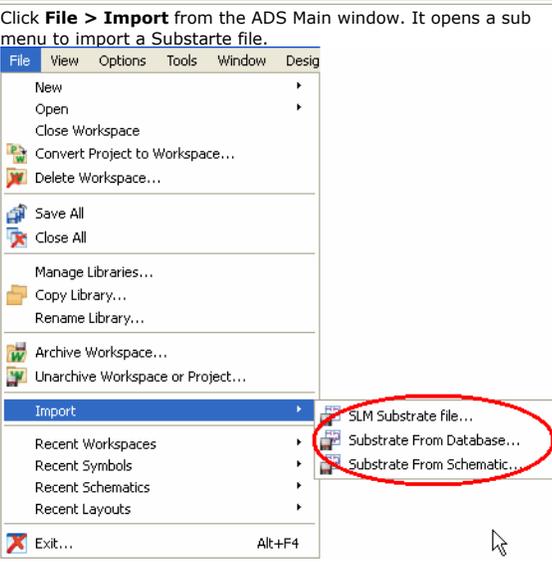
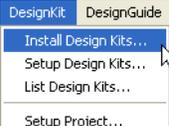
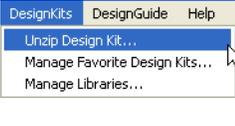
# Differences: ADS 2011 Vs. ADS 2009 Update 1

Following are the categories that lists the key differences between ADS 2011 and ADS 2009 update 1.

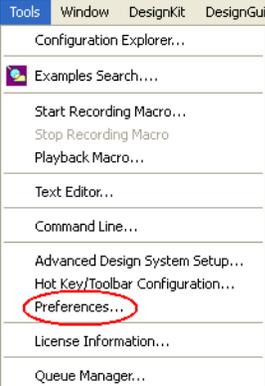
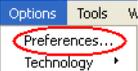
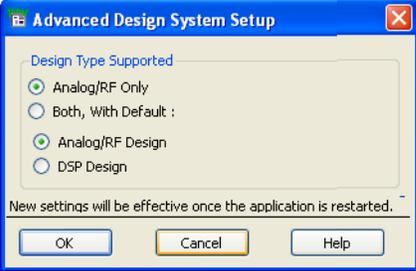
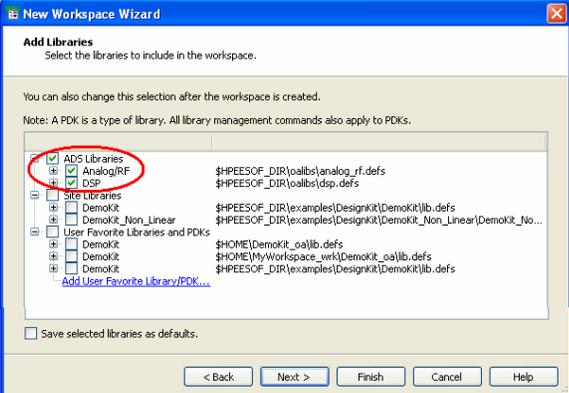
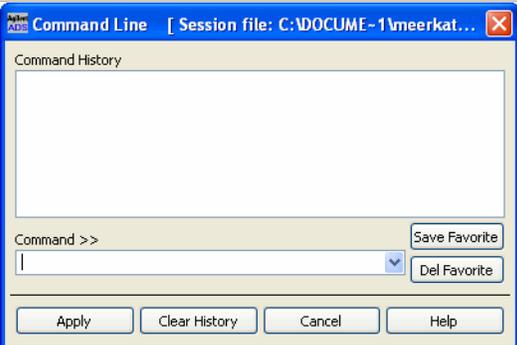
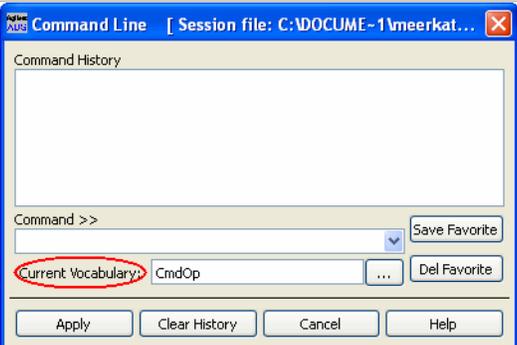
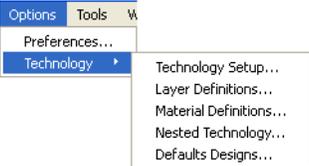
- Design and Architecture (oaqkref)
- ADS Main Window (oaqkref)
- Schematic Window (oaqkref)
- Layout Window (oaqkref)
- Symbol Window (oaqkref)

## ADS Main Window

The following table lists the ADS Main window differences in ADS 2009 U1 and ADS 2011:

Topic/Section	ADS 2009 Update 1	ADS 2011	Notes (if any)
Main Window			In ADS 2011, <i>Project View</i> tab is replaced by <i>Folder View</i> tab, and the <i>Library View</i> tab is added.  For more information, see <i>ADS Design Environment</i> (adstour).
File Menu			Following options are not available from ADS2011 onwards: <ul style="list-style-type: none"> <li>• <b>Copy Project.</b> Use Windows explorer option to copy a workspace.</li> <li>• <b>Include/Remove Projects.</b> Use <i>Manage Libraries</i> option to manage (Include/Remove) a library definition file from another workspace.</li> </ul> For more information, see <i>Things You Must Know About ADS 2011</i> (oaqkref)
File Import	Click <b>File &gt; Import</b> from the ADS Main window. The <i>Import</i> dialog box appears. 	Click <b>File &gt; Import</b> from the ADS Main window. It opens a sub menu to import a Substrate file. 	In ADS 2011, import and export of a design (Layout/Schematic) is done from the Layout/Schematic window. For more information on Importing and Exporting Designs, see <i>Importing and Exporting Designs</i> (trans).
Design Kits Menu			

## Advanced Design System 2011.01 - ADS Quick Reference

<p>ADS Preferences Window</p>	<p>To open an ADS Preferences window: Click <b>Tools &gt; Preferences</b> from the ADS Main window.</p> 	<p>To open the ADS Preferences window: Click <b>Options &gt; Preferences</b> from the ADS Main window.</p> 	<p>In ADS 2011 this option is moved from <b>Tools</b> menu to <b>Options</b> menu of ADS Main window.</p> <p>For more information on Main Preferences options, see <i>Setting Preferences for Miscellaneous Options</i> (adstour).</p>
<p>ADS Setup</p>			<p>In ADS 2011, ADS setup is not required as there is no difference between RF and DSP Libraries. You can select and include both the libraries while creating a workspace.</p>
<p>Command Line</p>			<p>The <i>Current Vocabulary</i> option is added in ADS 2011. This option enables you to select the AEL directory.</p> <p>For more information, see <i>Command Line Window (for AEL Commands)</i> (adstour).</p>
<p>Technology Definitions</p>	<p>No such option.</p>		<p>All designs in a library will use the technology information (Layers, Purposes, Units, Database Resolution, Display Properties, Substrates, Materials) defined for that library. Technology information can be added/edited using these options.</p> <p>For more information, see <i>Schematic Capture and Layout</i> (usrguide).</p>

## Design and Architecture

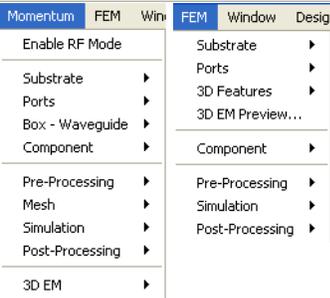
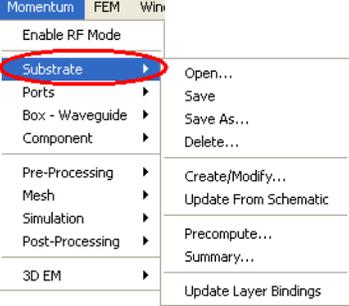
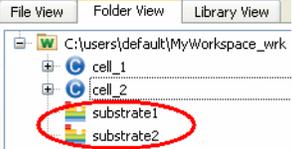
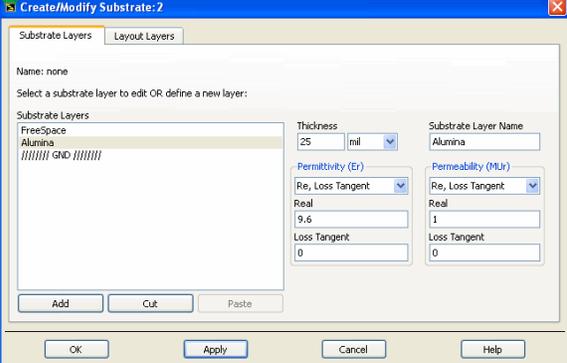
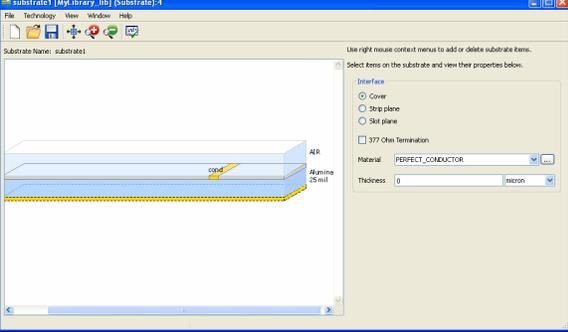
The following table lists the design and architecture differences in ADS 2009 Update 1 and ADS 2011:

## Advanced Design System 2011.01 - ADS Quick Reference

Topic/Section	ADS 2009 Update 1	ADS 2011	Notes (if any)
Project / Workspace	The data is stored and organized in projects and designs are saved to the networks directory of the project.	From ADS 2011 onwards, data is stored and organized in workspace. Each workspace contains a lib.defs file, which defines the libraries to be included in a workspace.	In ADS 2011: <ul style="list-style-type: none"> <li>Project is replaced with Workspace.</li> <li>Data is stored and organized in workspace.</li> <li>Each workspace contains a lib.defs file which defines the libraries to be included in a workspace.</li> <li>Designs are replaced with libraries, cells, and views.</li> <li>Cell contains a View. ADS workspace is similar to ADS project of ADS 2009 Update 1.</li> </ul> <p>For more details, see <i>ADS Workspace</i> (oaqref).</p>
Forms and Formsets	Most Forms and Formsets live in global space where any parameter definition can use any of the defined forms and formsets.	Forms and formsets are defined in a library. A parameter definition will have access to only the forms and formsets within the same library, except for the ADS built-in standard forms and formsets.	
Preference File (.prf)	The preference files can be applied per design in a project.	The preference file can be applied per workspace.	
Design Name Differences	Design name can be a full path or just the design name itself.	Design name consists of the following: <ul style="list-style-type: none"> <li>Library name</li> <li>Cell name</li> <li>View name of the design</li> </ul>	Examples of possible designName values for a design with name "top_level": <pre>// ADS 2009 Update 1 or earlier: // designName == C:\users\my_proj\networks\top_level // or // designName == top_level // // ADS 2011 onwards: // designName == my_workspace_lib:top_level:schematic</pre>
Component Name Differences	The component name may be a full path or just the component name itself.	The component name consists of the library name and cell name of the component.	// Examples of possible component name values for a component/cell with name "my_R": <pre>// ADS 2009 Update 1 or earlier: // componentName == C:\users\my_proj\networks\my_R // or // componentName == my_R // // ADS 2011 onwards: // componentName == my_workspace_lib:my_R</pre>

### Layout Window

The following table lists the Layout window differences in ADS 2009 U1 and ADS 2011:

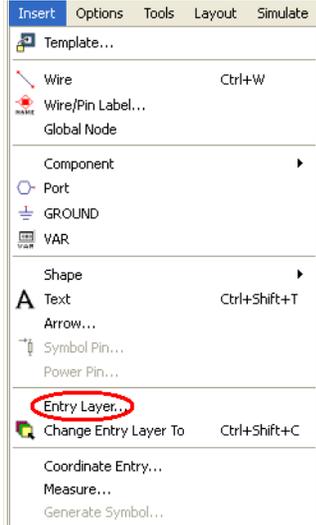
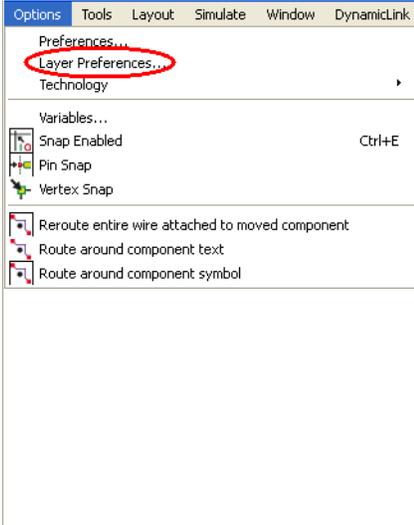
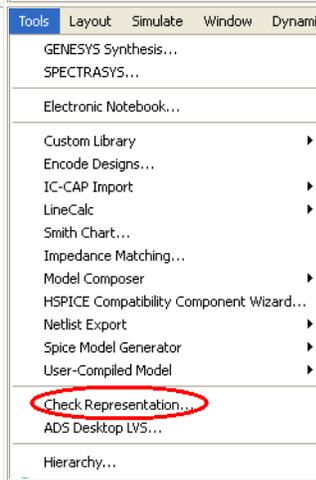
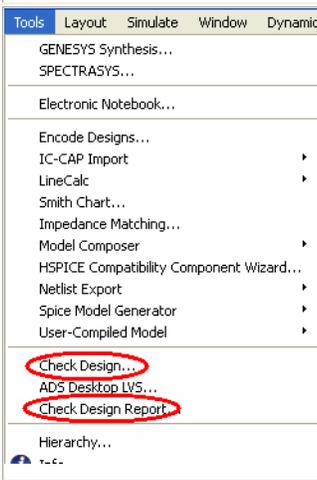
Topic/Section	ADS 2009 Update 1	ADS 2011	Notes (if any)
Momentum Menu			<p>In ADS 2011, FEM and Momentum menus are merged into One Single menu with additional features.</p>
Substrates		 	<p>In ADS 2011, you can easily create Substrates and these substrates are easily accessible just like any Cell Views.</p> <p>For more information, see <i>Substrates in EM Simulation</i> (adstour).</p>
Substrate Editor			<p>In ADS 2011 2D Graphical representation of substrate is introduced using which, you can easily modify a substrate.</p> <p>For more information, see <i>Substrate Editor</i> (adstour).</p>
Physical Connectivity Engine	<p>No such option.</p>		<p>In ADS 2011, you can enable Physical Connectivity Engine in a Layout which, was not possible in ADS 2009 U1.</p>
Accessing Bondwires			<p>In ADS 2011 Bondwires are easily accessible from <i>Insert</i> menu.</p>

**Schematic Window**

The following table lists the Schematic window differences in ADS 2009 U1 and ADS 2011:

Topic/Section	ADS 2009 Update 1	ADS 2011	Notes (if any)
New Design (Schematic)			<p>The new Schematic window in ADS 2011 can be started by selecting <b>File &gt; New &gt; Schematic</b> (Layout or Symbol option can be used as per need basis).</p> <p>For more information, see <i>Creating a New Schematic</i> (adstour).</p>
Insert Menu			<p><b>Insert &gt; Port</b> in a schematic window is replaced with <b>Insert &gt; Pin</b>.</p>
View Menu			<p>In ADS 2011, there is no option to <i>Create/Edit Schematic symbol</i> as you can directly create a new Symbol window from <b>File &gt; New &gt; Symbol</b>.</p> <p>For more information see, <i>Creating a New Symbol</i> (adstour).</p>
Layers Preferences			

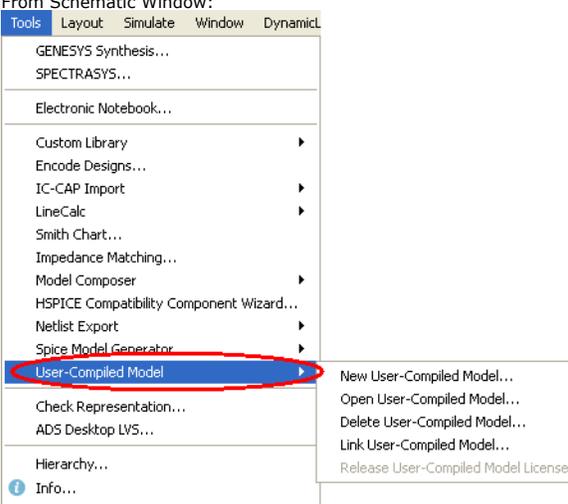
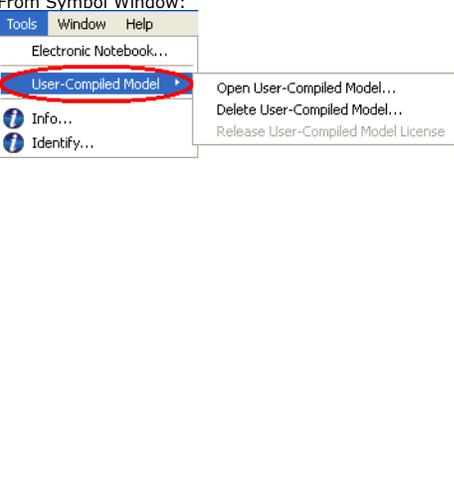
## Advanced Design System 2011.01 - ADS Quick Reference

<p>Layers Preferences</p>			<p><b>Insert &gt; Entry Layer</b> option in ADS 2009 U1 opens the <i>Layers Preferences</i> dialog box. In ADS 2011 this option is removed from the <b>Insert</b> menu, whereas you can still open the <i>Layers Preferences</i> dialog box from <b>Options &gt; Layers Preferences</b>.</p>
<p>Check Design</p>			<p><b>Tools &gt; Check Representation</b> in ADS 2009 U1 is now <b>Tools &gt; Check Design</b> in ADS 2011. Also a new option <b>Tools &gt; Check Design Report</b> is added to generate the design report. For more information see, <i>Checking Connectivity Information in Layout</i> (usrguide).</p>

### Symbol Window

The following table lists the Symbol window differences in ADS 2009 U1 and ADS 2011:

Advanced Design System 2011.01 - ADS Quick Reference

Topic/Section	ADS 2009 Update 1	ADS 2011	Notes (if any)
User-Compiled Model	<p>From Schematic Window:</p>  <p>The screenshot shows the 'Tools' menu in ADS 2009 Update 1. The 'User-Compiled Model' option is highlighted with a red oval. A sub-menu is open, listing options: 'New User-Compiled Model...', 'Open User-Compiled Model...', 'Delete User-Compiled Model...', 'Link User-Compiled Model...', and 'Release User-Compiled Model License'.</p>	<p>From Symbol Window:</p>  <p>The screenshot shows the 'Tools' menu in ADS 2011 Symbol Window. The 'User-Compiled Model' option is highlighted with a red oval. A sub-menu is open, listing options: 'Open User-Compiled Model...', 'Delete User-Compiled Model...', and 'Release User-Compiled Model License'.</p>	<p>In ADS 2009 U1 the UCM was generated from Schematic window and now in ADS 2011 the UCM is generated from Symbol window.</p>
Pin Representation	 <p>The image shows a small blue square representing a pin in the ADS 2009 Schematic window.</p>	 <p>The image shows a blue hexagon with a pin symbol inside, representing a pin in the ADS 2011 Symbol window.</p>	<p>In ADS 2011, Pin representation in Symbol window is same as it is in Schematic and Layout windows.</p>