# Innovus Stylus Common UI Mixed Signal Flow Task Assistant

Product Version 21.11
Last Updated in July 2021

Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

**Disclaimer:** Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information. Cadence is committed to using respectful language in our code and communications. We are also active in the removal and replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

# Contents

1

# Mixed Signal Flow Task Assistant

The *Mixed Signal Flow Task Assistant* is aimed at designers who wish to implement the digital Mixed Signal flow. The how-to format of the assistant makes it quick and easy to get answers to your Mixed Signal queries and complete the task at hand.

### Getting Started with the Mixed Signal Flow

Why should I use a Mixed Signal solution?

What are the supported Mixed Signal design flows?

### Preparing for the Mixed Signal Flow

What are the software requirements for the Mixed Signal flow?

What are the library and technology requirements for the Mixed Signal flow?

How to bring a design implemented in Virtuoso into Innovus?

How to set up the libraries needed for a Mixed Signal flow?

How to handle the vias in the design?

Are any special settings required for enabling the Mixed Signal flow?

### Implementing the Mixed Signal Flows

How to implement the Analog-on-Top (AoT) flow?

How to implement the Digital-on-Top (DoT) flow?

How to implement the Mixed-Signal-on-Top (MSoT) flow?

### Additional Information

What is the purpose of the oaZip utility?

How to ensure VDR interoperability?

FAQs and Useful Tips

# Why should I use a Mixed Signal solution?

Success in today's electronics market place requires highly integrated and low-cost solutions for wireless, consumer, computer, and automotive applications. Advanced process nodes now make it possible to manufacture analog and RF circuits down to 45 nm and below. Consequently, analog and Mixed Signal IP content is significantly increasing in system-on-chip (SoC) devices that in the past contained mostly digital circuitry. This situation creates new challenges for design, integration, and verification.

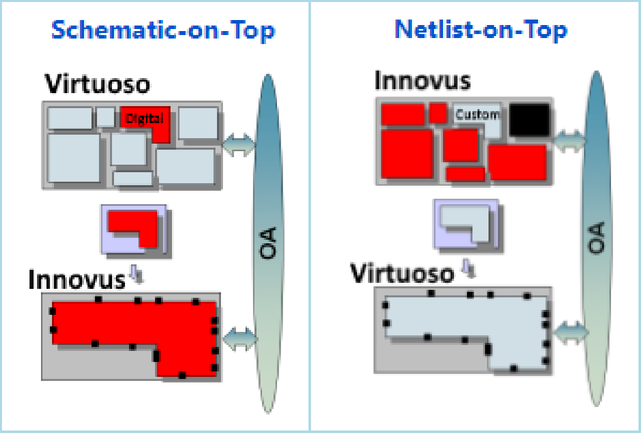
Cadence’s Mixed Signal solution addresses implementation and verification challenges and delivers a comprehensive, interoperable, and production-proven methodology. All design steps, including early design planning, front-end design, functional verification, physical implementation, signoff, and packaging, are shared responsibilities between analog and digital teams.

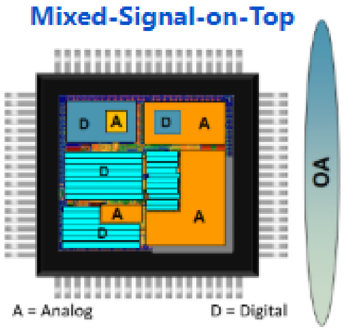
## Related Topics

- What are the supported Mixed Signal design flows?
- Mixed Signal Solution - Introduction

# What are the supported Mixed Signal design flows?

The Cadence Mixed Signal solution supports the following design flows. Click a flow to learn more about it.



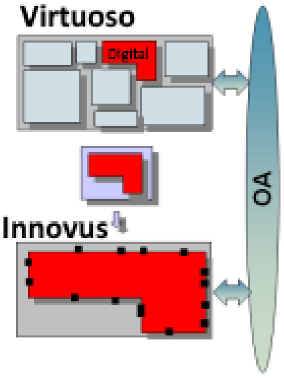


## Related Topics

- Overview of Mixed Signal Interoperability

## What is the Schematic-on-Top flow?

Use the schematic-driven Mixed Signal flow with an Analog-on-Top (AoT) methodology for designs that have large analog and small digital content.

| | |
|---|---|
|  | In the Cadence environment, if schematics are used for representing the top level, the design cockpit of choice is generally the Virtuoso® environment. The digital content is partitioned out and taken to Innovus. After the content is implemented and verified, it is brought back and re-mastered in the top-level design in Virtuoso. |

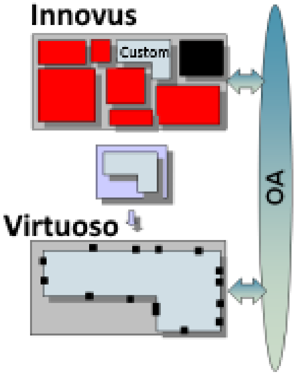
## Related Topics

How to implement the Analog-on-Top (AoT) flow?

## What is the Netlist-on-Top flow?

Use the netlist-driven Mixed Signal flow with a Digital-on-Top (DoT) methodology for designs that have a large digital content and a smaller analog content.

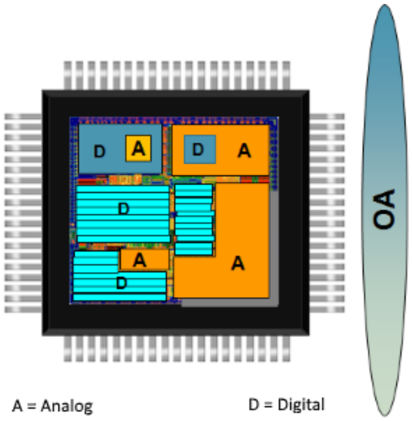
In the Cadence environment, if a Verilog netlist is used for the top-level design, the Innovus™ Implementation System software is generally used as the design cockpit. The part of the design that is targeted for implementation in Virtuoso is partitioned out. After the content is implemented and verified, it is brought back and re-mastered in the top-level design in Innovus.



For details, see How to implement the Digital-on-Top (DoT) flow?

## What is the Mixed-Signal-on-Top flow?

Use the concurrent Mixed Signal flow with a Mixed-Signal-on-Top (MSoT) methodology for designs that tightly integrate analog and digital functionality. This flow allows the mixing of digital and analog content throughout the design hierarchy. The underlining OpenAccess infrastructure allows you to take advantage of either the Innovus or Virtuoso cockpit, irrespective of whether the top-level design is represented as a schematic or a netlist.



For details, see Design Methodologies Supported by Cadence

# What are the software requirements for the Mixed Signal flow?

To build the Mixed Signal flow, you need Innovus Implementation System 17.1 or later with IC 6.1.6 ISR6  or later. Older versions of Innovus and Virtuoso have also been qualified for the Mixed Signal flow.

## Related Topics

Technology Data Preparation

# What are the library and technology requirements for the Mixed Signal flow?

- Technology and IP data on OpenAccess 2.2 Data Model 4 format
- A common (interoperable) Process Design Kit that has all the necessary technology information for Innovus and Virtuoso.
- Liberty timing library for standard cells and IP blocks and chip-level SDC file. If you do not have Liberty files and the chip-level SDC file, it will not be possible to perform static timing analysis of the top-level design.
- Extended FE capacitance table and QRC technology file to support extraction in Innovus.
- Power analysis libraries for enabling VoltageStorm analysis in Innovus.

## Related Topics

Technology Data Preparation

# How to bring a design implemented in Virtuoso into Innovus?

You can a bring a design implemented in Virtuoso into Innovus for further editing or analysis. Note that this flow requires an interoperable PDK to be in place before the design can be opened in Innovus. After the interoperable PDK is available, the design can be loaded in Innovus by using the following settings/commands after launching Innovus:

```
set init_design_netlisttype {OA}   # Tells Innovus that the netlist connectivity for
the design is coming through an OA DB, and not a Verilog netlist

set init_oa_design_lib {mylib}     # Specifies the name of the library containing the
design to be opened in Innovus

set init_oa_design_cell {top}      # Specifies the name of the top-level cell in the
design. In this example, the cell name is "top".

set init_oa_design_view {layout}   # Tells Innovus which view of the cell "top" to
open. In this example, the layout view is opened.
```

```
set init_oa_ref_lib {gsclib045}    # Tells Innovus which reference/technology is being
used for this design. In this example, gsclib045 is the interoperable PDK to be used.

set init_pwr_net {VDD AVDD}        # Defines power nets being used in the design. In
this example VDD and AVDD are used as power nets.

set init_gnd_net {GND AGND}        # Defines ground nets being used in the design. In
this example GND and AGND are used as ground nets.

set init_mmmc_file {viewDefinition.tcl}  # Optional setting pointing to the view
definition files used to set up the environment for Multi Mode Multi Corner (MMMC)
analysis and implementation. Use if you are bringing the design into Innovus to perform
timing analysis or optimization.

init_design                        # The actual command that will load the design into
Innovus
```

For more details, see Static Timing Analysis for Mixed Signal Designs.

# How to set up the libraries needed for a Mixed Signal flow?

- How to prepare the technology library?
- How to prepare the IP library?

## Related Topics

Technology Data Preparation

## How to prepare the technology library?

The OpenAccess-based, interoperable Mixed Signal flow requires the implementation of a common OpenAccess Process Design Kit (MSOA PDK). The MSOA PDK contains the information typically found in a base PDK and in the technology LEF. The base PDK, which is the PDK used for Virtuoso, is expected to have at least the foundry rules for the masterslice layers and will typically include the foundry rules for all the metal and cut layers. The LEF would have unique information not present in the PDK, such as default and non-default routing rules.

Starting from the 18.1 release of Innovus, you now have multiple options for creating an MSOA PDK:

- Innovus-only MSOA PDK (useful for experimenting with Innovus in OpenAccess mode)
- Traditional MSOA PDK (available prior to the 18.1 release)
- Rapid MSOA PDK (available in 18.1 and later releases)

## Related Topics

Preparing the Technology Library

# How to prepare the IP library?

1. Ensure that the IP library contains the abstract views of all IP blocks used at the top level in the design.
2. Set the SYMMETRY attribute.
3. Add the analog property, `oacSigTypeAnalog`, on analog nets so that Innovus maintains such nets as dbIsAnalog true. (Optional)
4. Use Virtuoso Abstract Generator to generate abstracts for the analog IPs that are finally used by Innovus.
5. Save the OpenAccess database of IP abstracts to be used by the digital toolsets. Use verilogAnnotate, a stand-alone UNIX utility, to annotate the bus terminal list and order the distributed terms in the physical abstract.
6. Unset the environment variable OA_HOME. Source the Virtuoso path setting file so that you can get Virtuoso XL/GXL in your PATH environment variable.
7. Open the design with parameterized cell (PCell) in Virtuoso XL/GXL, and regenerate the PCell cache. This step enables inter-operation of data between Innovus and Virtuoso platforms.

## Related Topics

Preparing the IP Library

# How to handle the vias in the design?

In an OpenAccess database, the technology information (layers, layerRules, vias, constraintGroups) about a specific process and metal stack can be spread across multiple tech files. This approach is called an Incremental Technology Database (ITDB). A key advantage of the incremental approach is that you can make changes in one library without having to modify the baseTech library. For example, you can add a new via or constraint group to a library for Innovus.

In the ITDB approach, several methods are available for defining vias in the OpenAccess libraries for use in interoperable flows by both Virtuoso and Innovus. These include customViaDefs, standardViaDefs, and standardViaVariants.

If you think there is a need for special handling of vias in your design, refer to the "*Working with Vias in OpenAccess*" chapter of the *Mixed Signal (MS) Interoperability Guide*.

## Related Topics

Working with Vias in OpenAccess

# Are any special settings required for enabling the Mixed Signal flow?

Some special settings are required to make the software work properly for Mixed Signal users. To learn about these settings, study the **Special Settings/Instructions To Enable the Mixed Signal Flow** section in the "*Design Data Preparation"* chapter of the *Mixed Signal (MS) Interoperability Guide*.

In addition, pay special attention to the `oa_read_write` category attributes in Innovus because these attributes control many of the Innovus Mixed Signal features. For more information on the `oa_read_write` category attributes, refer to the *Innovus Stylus Common UI Text Command Reference*.

## Related Topics

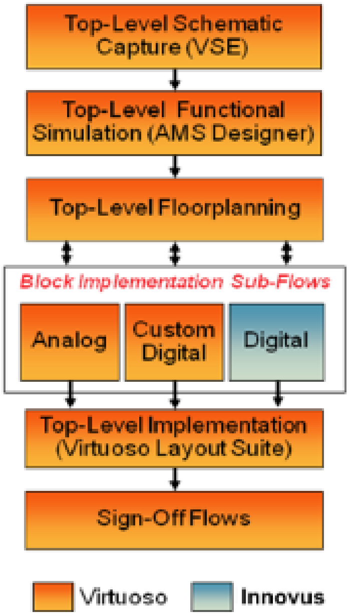
- Special Settings/Instructions To Enable the Mixed Signal Flow
- oa_read_write Attributes

# How to implement the Analog-on-Top (AoT) flow?

Use the schematic-driven Mixed Signal flow with an Analog-on-Top (AoT) methodology for designs that have large analog and small digital content (also known as big A little D or A/d). The schematic-driven Mixed Signal methodology is also referred as the schematic-on-top methodology because the top level of a design in this flow is typically represented as a schematic.

Schematic-on-top does not mean that the schematic must exist for all the blocks in the design throughout the entire hierarchy. It only refers to the top-level. In addition, schematics are used to drive the implementation of any analog portions of the design. However, netlists are used for most of the digital-design content.

At the initial stages of the design, only functional and performance specifications are available. The designer has the task of deciding the optimal-design partitioning with the major sub-systems identified as sub-blocks. Normally, the design partitioning is determined by the major functional blocks, performance, and design disciplines required for the overall implementation. As a result, the design is decomposed into analog and digital functional blocks.


Top-Level Schematic Capture (VSE)
Top-Level Functional Simulation (AMS Designer)
Top-Level Floorplanning
Block Implementation Sub-Flows
Analog
Custom Digital
Digital
Top-Level Implementation (Virtuoso Layout Suite)
Sign-Off Flows
Virtuoso
Innovus

## Related Topics

How to implement digital functionality by using VDI?

For details on implementing the AoT flow, refer to Schematic-Driven Mixed Signal Design Flow.

## How to implement digital functionality by using VDI?

In the Cadence Mixed Signal implementation flow, Innovus Implementation System is used for the complete implementation of digital content of the design. Very often the digital content is captured in one or more digital blocks contained in an Analog-on-Top (AoT) design. As the AoT design style is most often used by Virtuoso-based Mixed Signal design teams, the team may be unfamiliar with Innovus and the flow used for implementing the digital portion of Mixed Signal designs. As a result, many Mixed Signal design teams rely on digital designers to help implement the required digital functionality in a Mixed Signal design.

The Virtuoso Digital Implementation (VDI) environment in Virtuoso enables users who do not have familiarity with digital design methodology to implement the required digital functionality easily. The VDI environment invokes Innovus from within Virtuoso and creates the necessary script to implement the needed digital functionality completely.

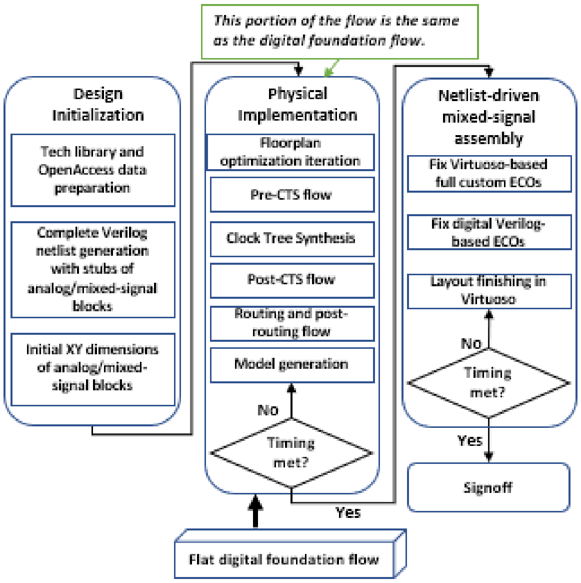
### Related Topics

Virtuoso Digital Implementation

# How to implement the Digital-on-Top (DoT) flow?

Use the netlist-driven Mixed Signal flow with a Digital-on-Top (DoT) methodology for designs with large digital content and smaller analog content (also known as big D little A or D/a). As the top level in this flow is typically represented as a Verilog netlist, designers refer to the netlist-driven mixed-signal flow as the netlist-on-top methodology.

A typical netlist-driven MS flow may look similar to the digital implementation flow with minor differences during floorplanning, top level specialty net routing, and ECO flows.


This portion of the flow is the same as the digital foundation flow.
Design Initialization
Tech library and OpenAccess data preparation
Complete Verilog netlist generation with stubs of analog/mixed-signal blocks
Initial XY dimensions of analog/mixed-signal blocks
Physical Implementation
Floorplan optimization iteration
Pre-CTS flow
Clock Tree Synthesis
Post-CTS flow
Routing and post-routing flow
Model generation
No
Timing met?
Yes
Flat digital foundation flow
Netlist-driven mixed-signal assembly
Fix Virtuoso-based full custom ECOs
Fix digital Verilog-based ECOs
Layout finishing in Virtuoso
No
Timing met?
Yes
Signoff

## Related Topics

Netlist-Driven Mixed Signal Design Flow

# How to implement the Mixed-Signal-on-Top (MSoT) flow?

If you are bringing the entire mixed-signal design from one implementation platform to another (Virtuoso to Innovus or vice versa), you will be exercising the Mixed-Signal-on-Top (MSoT) flow.

The MSoT flow supports STA, Floorplanning, and Routing for Mixed Signal.

## Related Topics

- How to run Static Timing Analysis on a Mixed Signal design?
- How to implement Mixed Signal floorplanning?
- How to implement Mixed Signal routing?
- How to implement ECO flows?

## How to run Static Timing Analysis on a Mixed Signal design?

In analog Mixed Signal designs, the digital logic is placed inside the analog Mixed Signal hierarchy and the entire digital logic path must be accurately analyzed during full-chip Static Timing Analysis (STA).

For schematic- as well as netlist-driven Mixed Signal flows, Cadence offers two ways of running STA on Innovus through the interoperable OpenAccess database:

**Option 1**: Flatten the design to a certain physical level so that the timing paths to be analyzed are visible and extractable by Innovus.

**Option 2**: Generate full timing models (FTMs) for Mixed Signal blocks that contain the timing paths to be analyzed. The FTM contains full logical netlist information and RC parasitic information of the blocks and enables STA to be done in Innovus without providing the physical layout of the blocks.

### Related Topics

- OpenAccess Database Interoperability Checker
- Static Timing Analysis for Mixed Signal Designs

## How to implement Mixed Signal floorplanning?

To floorplan the design in Virtuoso and Innovus, you would need to share the floorplan information between the two cockpits. To understand how to do so, refer to the "*Quick Abstract Inference*" chapter and the **Controlling Floorplanning Information Read In from the OpenAccess Cellview** section in the "*Schematic-Driven Mixed Signal Design Flow*" chapter of the *Mixed Signal (MS) Interoperability Guide*.

### Related Topics

- Quick Abstract Inference
- Schematic-Driven Mixed Signal Design Flow

## How to implement Mixed Signal routing?

In the Cadence interoperable OpenAccess environment, design constraints can be shared across various platforms. For instance, in a Mixed Signal flow with the top level of the design captured as schematics in Virtuoso, routing constraints defined in Virtuoso for the digital portion of the design can later be implemented by Innovus. Conversely, in a flow where the top level of the design is captured as a Verilog netlist in Innovus, routing constraints can be defined in Innovus and can be used by block designers when such blocks are later implemented in Virtuoso.

In addition to these capabilities, Mixed Signal IPs can be embedded with the required integration constraints so that when the end users instantiate such IPs in their designs, the embedded constraints can be pulled to the top level of the design. This ensures that the integration requirements of the particular IP are met during the design process.

NanoRoute High Frequency (NRHF) router is a high-frequency (HF) router in Innovus that addresses the custom and high-frequency routing needs of Mixed Signal users. It provides full interoperability between various Cadence tools. NRHF supports a multiple-entry mechanism for routing constraints. Routing constraints can be entered through the Integration Constraint Editor (ICE) in Innovus or through the Constraint Manager in Virtuoso. For constraints entered in Virtuoso, the OpenAccess-based flow should be used.

### Related Topics

- Routing Constraint Interoperability
- High Frequency Router In Innovus

## How to implement ECO flows?

Most of the custom design level sign-off analysis is done in Virtuoso. The sign-off timing analysis of the flat top-level digital portion of the design can be done in Innovus. The Mixed Signal functional simulation gives functional performance of the design at any stage of the design cycle.

Metal filling, optimization of metal density can be done in Innovus also which can be taken into account by QRC during extraction so that the timing effects due to metal fill can be addressed.

### Related Topics

- Chip Finishing and ECO Flows
    - Virtuoso-Based ECO Flow

- Innovus-Based ECO Flow

# What is the purpose of the oaZip utility?

OpenAccess 22.42 and later releases support the ability to save the design databases in a library in a compressed form. Tools based on OpenAccess 22.41 or earlier releases (such as IC6.1.5), may need to have the designs in the library decompressed with the `oazip` utility before the designs can be accessed.

## Related Topics

oaZip Utility

# How to ensure VDR interoperability?

You can create Voltage Dependent Rules (VDRs) which can be associated with signal nets using the Common Power Format (CPF). The layer purpose per voltage is carried with the VDR rules itself. With this understanding, NanoRoute and WireEditor simply use the DRC rules coming from specified VDR and voltage values on the nets coming from CPF.

**Note**: The VDR file is required only with CPF. In the case of Unified Power Format (UPF), the voltage specs are part of the view definition files and so the VDR file is not required.

## Related Topics

Voltage Dependent Rule Interoperability

# FAQs and Useful Tips

- How to view PCells in Innovus?
- How to resolve a conflict with a global net name?
- How to save a blackbox in OpenAccess?
- How are power and ground pins generated?
- What happens if cells remastered with layout views are read into Innovus?

- How to handle power routing created outside Innovus?
- How to generate abstracts with antenna information?

For answers to more such questions, refer to the "*Useful Tips*" chapter of the *Mixed Signal (MS) Interoperability Guide*.

## Related Topics

Useful Tips

## How to view PCells in Innovus?

To view pcells in Innovus, set the following variables on UNIX prompt before starting the Innovus session:

```
setenv CDS_ENABLE_EXP_PCELL TRUE

setenv CDS_EXP_PCELL_DIR <directory path>/.expressPcells
```

## How to resolve a conflict with a global net name?

If the module `dtmf_chip_block` has a local net name `?VDD?` that collides with a global net name `?VDD?`, then a local-instance-terminal power connection requires access to the global net but is blocked by the local net with the same name. As a result, a new global net `?OAX_VDD_1?` is created as an equivalent net to the `?VDD?` global net while reading a design inside Innovus. The new global net will then be used for these power connections. The two global nets are effectively the same.

To avoid this collision, do not use local net names that collide with global net names.

## How to save a blackbox in OpenAccess?

A blackbox must be committed into a partition before saving it to OpenAccess. After doing this, a cell type will not remain as blackbox, but will get saved as a block.

## How are power and ground pins generated?

Power and ground pins are created in three ways:

- When you create a power mesh in Innovus, power and ground pins are automatically generated in VDI.

- In `sroute`, you have an option to create pins at the intersection of the power grid and `prBoundary`.
- Use the `createPGPin` command in Innovus.

## What happens if cells remastered with layout views are read into Innovus?

Innovus is designed to use abstract views. If cells remastered with layout views are read in Innovus, then errors might appear with `verifyGeometry`, and so on.

## How to handle power routing created outside Innovus?

Power routes created using Virtuoso might not have the `stripe` attribute that is needed by `sroute` in Innovus. To set the shape attribute on all the special wires (power routes) of `$my_nets` to `stripe`, use the following command in Innovus:

```
dbSet $my_nets.sWires.shape stripe
```

## How to generate abstracts with antenna information?

To generate abstracts with antenna information, you can use either `write_lef_abstract` (after `verifyProcessAntenna`), `save_abstract` (after `verifyProcessAntenna`), or use the abstract tool. If you want to do cover obstruction style modeling for the completed partition, use `write_lef_abstract`/`save_abstract`. For detailed modeling, you can use the abstract tool.

For determining metal density, you can run `verifyMetalDensity -saveToDB` before `write_lef_abstract`/`save_abstract`.

**Note:** Using non-overlapping windows is recommended for this usage.