cadence®

Virtuoso What's New

Product Version ICADVM20.1
October 2020

Printed in the United States of America.

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

Contents

Preface

This guide provides a high-level overview of the new and enhanced features in the ICADVM20.1 base release and subsequent ISR releases. Each chapter in this guide maps to a Virtuoso application and may contain one or more of the following sections:

- New and enhanced features
- Removed features
- Documentation updates, including a list of new and updated videos

Note: The table of contents lists only those releases in which new features or enhancements have been introduced.

This preface contains the following topics:

Scope

The functionality described in this guide can be used only in ICADVM20.1 advanced nodes and advanced methodologies releases.

Label	Meaning
(ICADVM20.1 Only-95512)	Features supported only in the ICADVM20.1 release and which require the Virtuoso_Adv_Node_Opt_Lay_Std license (95512).

(ICADVM20.1 Only-95511)	Features supported only in the ICADVM20.1 release and which require the Virtuoso_Adv_Node_Opt_Layout license (95511).
(ICADVM20.1 Only – Virtuoso MultiTech Framework)	Features supported only in the ICADVM20.1 release and which require the Virtuoso_MultiTech_Framework (95022) license.

Licensing Requirements

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

Related Documentation

Release README.txt

- The contents of `README.txt` file are organized under the following heads:
  - Supported Platforms and Operating Systems
  - Cadence Products and Standalone Software Compatible with the IC Release
  - Installation Information
  - Accessing Product Documentation
  - Contact Cadence
  - CCRs Fixed in <*release information*> (this comprises a list of customer-reported bugs that have been fixed in the current release)
- To view the contents of `README.txt`, do one of the following:
  - Locate the file in the installation directory.
  - Log on to downloads.cadence.com and follow the 'Get release information' instructions.

Known Problems and Solutions

For information about Known Problems and Solutions (KPNS), see *Virtuoso Known Problems and Solutions*.

Additional Learning Resources

Video Library

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

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

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

Virtuoso Videos Book

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

Rapid Adoption Kits

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

To explore the full range of training courses provided by Cadence in your region, visit Cadence Training or write to training_enroll@cadence.com.

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

Help and Support Facilities

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

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

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

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

For more information, see Getting Help in *Virtuoso Design Environment User Guide.*

Customer Support

For assistance with Cadence products:

- Contact Cadence Customer Support

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

- Log on to Cadence Online Support

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

Feedback about Documentation

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

- Find erroneous information in a product manual
- Cannot find in a product manual the information you are looking for
- Face an issue while accessing documentation by using Cadence Help

You can also submit feedback by using the following methods:

- In the Cadence Help window, click the *Feedback* button and follow instructions.
- On the Cadence Online Support Product Manuals page, select the required product and submit your feedback by using the *Provide Feedback* box.

Typographic and Syntax Conventions

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

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

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

Overview

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Overview of Virtuoso ICADVM20.1

In ICADVM20.1, Cadence introduces the following enhancements, which are seamlessly integrated into the Virtuoso platform to provide a wider range of high performance capabilities.

Products and Features Introduced in ICADVM20.1

- **3nm Process Support**

  The new Virtuoso platform incorporates advanced-node process design rule support and methodologies from 22nm down to 3nm. Virtuoso can now automatically manage process complexities that enable you to focus on the design intent. In layout design, a unique multi-grid system abstracts complex design rules of the latest 7nm through 3nm processes, while helping you increase your use of placement and routing technologies to significantly increase layout design productivity.

- **Virtuoso ADE Assembler: Merging Histories**

  You can now merge simulation results from multiple histories. This helps you bring together all relevant results without running simulations again. The feature provides the flexibility to merge selected results or add new points and simulate those while merging existing results. For more information, see *Virtuoso ADE Assembler User Guide*.

- **Virtuoso ADE Explorer and Virtuoso ADE Assembler: Distributing Simulations Using LSCS**

  LSCS is now the default job control mode for job policies. LSCS implements a new architecture to run simulations with better resource utilization. It also enables future scalability requirements, mainly cloud simulations. For more information, see *Virtuoso ADE Assembler User Guide*.

- **Virtuoso Design Planning and Analysis**

  The Design Planner tool has been replaced by the Design Planning and Analysis (DPA) tool, which provides a unified set of commands under the new Plan menu to support all design planning, floorplanning, global route planning, and analysis tasks. The Design Planning and Analysis tool supports hard blocks for use in floorplanning designs that require no changes, soft blocks for generating cells with real hierarchy that have boundary and optimization support for pins, and virtual hierarchy for generating groups of matching, flat schematic cells within a boundary.

  Virtual hierarchy enables editing in the context of the entire hierarchy. Because the virtual hierarchy is flat, no pins are needed during early design planning, avoiding rework due to changes in the design. Being contained in the top-level layout, virtual hierarchies

alleviate the need to create multiple cellviews for planning purposes. The virtual hierarchies can also be easily converted to real hierarchies on demand by using the *Make Cell* command, and multiple selected cells can be integrated back into a design using the Make Virtual Hierarchy command.

The Design Planning and Analysis tool also supports soft block generation in the absence of schematics. This soft block support can also be useful for pin optimization. For more information, see the *Virtuoso Design Planning and Analysis User Guide*.

■ Virtuoso Power Manager

Virtuoso Power Manager provides an interface to specify, import, and export low power intent for designs. It provides the capability to perform static low power verification on designs by using Conformal Low Power (CLP) integration with Virtuoso and supports analog, digital, and mixed-signal implementations. To reduce the complexity, you should be able to perform a few basic-to advanced-level circuit design checks on schematic designs. The In-Design Checks feature in Power Manager provides such static checks that do not require circuit simulation.

■ Virtuoso Schematic Editor: Enhanced Connectivity Extractor

The schematic connectivity extractor has been fundamentally redesigned to facilitate improvements in performance, memory usage, and efficiency of connectivity. Previously, each time you ran a connectivity check, the schematic extractor would delete and rebuild all the connectivity in a design. Now, the connectivity is updated such that only the necessary changes are made to the database connectivity.

As a result of this change, a new incremental extraction mode has been added, which means that the connectivity of your design is updated automatically as you create and edit the schematic. If you introduce issues that prevent the connectivity from being made when updating your design, error and warning messages and markers are generated and placed in the canvas, the Annotation Browser, and the CIW to make you aware of the problem.

■ Virtuoso Schematic Editor: Support for Split Symbols

Additionally, you can now create splits of large symbols in the Virtuoso Schematic Editor. Splitting large symbol masters improves the legibility of pin symbols, especially in large PCB and package schematics.

Products and Features Removed from ICADVM20.1

- The following Virtuoso products and the documentation related to these have been removed:
  - Virtuoso Analog Design Environment L
  - Virtuoso Analog Design Environment XL
  - Virtuoso Analog Design Environment GXL

  Use the more efficient and powerful environments, Virtuoso ADE Explorer and Virtuoso ADE Assembler, to run simulations for your analog or mixed-signal designs.

Documentation Updates in ICADVM20.1

Restructured Documentation

The environment variables that you can use for Virtuoso ADE Explorer and Virtuoso ADE Assembler have been repackaged in the following reference guides:

- *Virtuoso ADE Environment Reference - Part I*
- *Virtuoso ADE Environment Reference - Part II*
- *Environment Variables* appendix in Virtuoso ADE Explorer User Guide
- *Environment Variables* appendix in Virtuoso ADE Assembler User Guide

The SKILL functions that you can use for Virtuoso ADE Explorer and Virtuoso ADE Assembler have been repackaged in the following reference guides:

- *Virtuoso ADE SKILL Reference - Part I*
- *Virtuoso ADE SKILL Reference - Part II*
- *Virtuoso ADE Explorer and Virtuoso ADE Assembler SKILL Reference*

Removed Documentation

The documentation for the following products is not available from the ICADVM20.1 release.

- *AMS Design and Model Validation*
- *SKILL IDE Developer's Quick Start Guide*

The information it contained is now available in the *Cadence SKILL IDE User Guide*.

- *Virtuoso Analog Design Environment L User Guide*
- *Virtuoso Analog Design Environment XL User Guide*
- *Virtuoso Analog Design Environment GXL User Guide*

Overview of Virtuoso ICADVM18.1

In ICADVM18.1, Cadence introduced the new Virtuoso Design Platform that integrates a wide range of tools, products, and solutions with multi-technology support. It lets you seamlessly edit and analyze the most complex heterogeneous systems, including analog, mixed-signal, and RF products through a single platform.

Products and Features Introduced in ICADVM18.1

- **5nm**

  The new Virtuoso platform incorporates advanced-node process methodologies from 22nm down to 5nm. Virtuoso can now automatically manage process complexities that enable you to focus on the design intent. In layout design, a unique multi-grid system abstracts complex design rules of the latest 7nm and 5nm processes, while allowing you to increase your use of placement and routing technologies to significantly increase layout design productivity.

- **Concurrent Layout Editing**

  Virtuoso Concurrent Layout creates an editing environment within Virtuoso that allows several designers to concurrently work on the same cellview. As the designers work in parallel, using Concurrent Layout helps reduce design time and, as a result, increases productivity. For more information, see *Virtuoso Concurrent Layout User Guide*.

- **Congestion Analysis**

  The Congestion Analysis assistant facilitates the quick and accurate modeling of routing congestion to help improve floorplanning, optimize pin generation and placement, and reduce overall die size. The feature lets you extract, display, and analyze routing congestion both visually and statistically, and offers sophisticated tools facilitating the targeted optimization of routing paths for critical nets and net groups. For more information, see *Running Congestion Analysis*.

- **Design Intent**

  Design Intent complements the Virtuoso Schematic Editor XL and Virtuoso Layout Suite XL applications, providing a method for schematic designers to capture and communicate their design goals and allowing layout designers the freedom to decide how those goals are implemented. For more information, see *Virtuoso Design Intent User Guide*.

- **Design Planner**

  Design Planner provides an innovative layout-place-route solution for advanced node and mature node designs. It offers you the capabilities to make more informed planning

decisions earlier in the design life cycle, backed by real-time congestion analysis data. For more information, see *Virtuoso Design Planner User Guide*.

- **Design Rule-Driven Editing**

  The Design Rule Driven (DRD) tool has been redesigned with new interaction paradigms to make in-design verification smarter and more efficient than ever. It offers a new consolidated interface, smart use models for improved visualization, and enhanced user control over the constraint and layer filters for design verification. For more information, see *Virtuoso Design Rule Driven Editing User Guide*.

- **Simulation-Driven Layout**

  The Simulation-Driven Interactive Routing (SDR) feature has been designed to elevate Virtuoso from an electrically aware environment to a simulation-driven environment. It addresses many of the electromigration and parasitic challenges of critical circuits and advanced node designs. SDR provides a new way for a layout designer to have a predictable flow to meet the current density constraints, and in turn, significantly reduces the sign-off time and improves the productivity and design reliability. For more information, see *Virtuoso Simulation Driven Interactive Routing User Guide*.

- **Technology File Integrated Development Environment**

  The Techfile IDE offers several useful features over the conventional text editors to help you work more efficiently with ASCII technology files. It makes it easier for you to understand, write, and modify technology files. The Techfile IDE is particularly useful for grouping and managing the constraints within constraint groups in a compiled technology database. For more information, see *The Techfile IDE*.

- **Virtuoso RF Solution**

  The Virtuoso RF solution has enhanced Virtuoso with package design capabilities to enable you to implement, analyze, and simulate RFIC and RF modules all within the Virtuoso design environment. The Co-Design feature allows package and die designs to be developed simultaneously and in concert with one another. The VRF solution automates the process to run electromagnetic simulations, allowing you to run in-design experiments. It also simplifies the effort of running layout versus schematic checks and design rule checks for complete physical design checks. For more information, see *Virtuoso RF Solution Guide*.

- **Virtuoso Photonics Solution**

  The Virtuoso Photonics Solution introduces an integrated electronic-photonics design automation environment in the Virtuoso design framework. The Photonics Solution provides an end-to-end designing, simulation, and implementation solution for photo-electronic designs. With its integrated CurvyCore infrastructure, the Photonics Solution is capable of providing simple design solutions to the complex mathematical problems of

Photonics Integrated Circuits (PIC). For more information, see *Virtuoso Photonics Solution Guide*.

In addition to the above features of the Virtuoso Design Platform, there are several other notable features and enhancements in Virtuoso. These include:

- **Virtuoso SystemVerilog Netlister**

  The new Virtuoso® SystemVerilog Netlister utility helps generate netlists for digital SystemVerilog designs. This utility imports configuration views of digital designs for netlist generation, directly parses and accesses SystemVerilog and Verilog text models, and creates LRM-compliant SystemVerilog configurations to generate compatible netlists. For more information, see *Virtuoso SystemVerilog Netlister User Guide*.

- **cdsTextTo5x for Virtuoso Design Environment**

  Verilog netlists for very large designs can be imported very quickly by using the new `cdsTextTo5x` command. For example, Cadence benchmarks indicate that a large design netlist featuring up to 20M pins and 10M instances can be imported within an hour, representing a 10x performance improvement. Additionally, you can use the command-line option `-BLACKBOX` to import a cell as a blackbox when you do not require its hierarchy to be imported. For more information, see *cdsTextTo5x for Virtuoso Design Environment User Guide*.

- **Virtuoso Automated Device-Level Layout**

  The Virtuoso automated device-level layout flow enables you to quickly generate placed and routed layouts that are constraint-compliant and LVS-correct, and follow DRCs as captured in the Virtuoso technology file. The layouts also incorporate base layer fill, as typically required in advanced nodes. These layouts can be used to extract parasitics for re-simulation to identify issues early on, without waiting for the final sign-off, and can be easily modified and updated to generate the final layout for sign-off. For more information, see *Virtuoso Automated Device-Level Layout Flow Guide*.

- **Virtuoso Layout Suite**

  ICADVM18.1 introduces two new applications, Layout Viewer and Layout EXL, which now sit alongside Layout XL in the Virtuoso Layout Suite. For detailed information, see Virtuoso Layout Suite Changes.

- **Multi-Technology Framework Design Environment**

  The Virtuoso MultiTech Framework Design Environment allows you to choose the physical implementation options of a module or a package. In this framework, you can access all the features related to Virtuoso System Design Platform through the *Module* menu in Virtuoso Schematic Editor XL and Virtuoso Layout Suite EXL.

ADE Support for Smart View

The Parasitic-Aware Design flow in ADE supports a new version of extracted view, called smart view, that is created using Quantus. The smart view provides the same functionality as provided by the extracted view, but has a highly efficient and scalable storage mechanism that helps in achieving better performance in case of large and complex designs because it reduces database and netlist size. For more information, see *Virtuoso Parasitic Aware Design User Guide*.

Analog Fault Simulation

As a part of the Legato Reliability Solution, ADE Assembler supports fault analysis, which adds an additional capability into the analog design simulations to identify the possible defects and generate useful reports to help you quantify the manufacturing defect coverage, and thereby reduce test escapes. For more information, see *Legato Reliability Solution*.

Flexible Subwindows in ViVA

The subwindows layout in Virtuoso Visualization and Analysis XL has been enhanced to provide more flexibility. You can now customize subwindows, drag and resize them, open blank subwindows for rectangular, polar, impedance, and admittance plots, and plot signals into them. For more information, see *Virtuoso Visualization and Analysis XL User Guide*.

Pin Tool

Virtuoso Floorplanner includes the Pin Tool, which is a unified interface for all pin-related tasks, such as creating, resizing, planning, and optimizing pins, editing pin attributes, and setting location constraints for pins. For more information, see *Virtuoso Floorplanner User Guide*.

Electrically Aware Pin Sizing

The Pin Tool provides a functionality to identify pins that do not meet the electrical requirements of the design, view the violations in Annotation Browser, and automatically fix them. For more information, see *Virtuoso Floorplanner User Guide*.

Automatic Generation of Hierarchies

Auto-Generate Hierarchy (AGH) is a command that you can use to generate layouts for the entire schematic hierarchy, with instances placed and pins positioned at every level in the hierarchy. For more information, see *Virtuoso Floorplanner User Guide*.

Deep Hierarchy Editing

After generating the hierarchy, for example using Auto-Generate Hierarchy, you can enable deep hierarchy editing to view and edit the full depth of the layout hierarchy

directly from the top level for floorplanning. There is no need to descend into the hierarchy. For more information, see *Virtuoso Floorplanner User Guide*.

- **Plotting Templates**

  Plotting templates in ADE Explorer and ADE Assembler are used to plot resultant waveforms, signals, or scalar values in a specified format in the Virtuoso Visualization and Analysis XL window. All the formatting, markers, layout changes, and other interface changes are retained as specified in the plotting template. For more information, see *Virtuoso ADE Explorer User Guide*.

- **Predefined Setup-Driven MPT Flows**

  You can now use a set of newly defined SKILL APIs to set up the MPT engine with predefined MPT flows. For more information, see *Virtuoso Layout Suite SKILL Reference*.

- **Simulation Planning and Coverage**

  ADE Assembler and ADE Verifier now allow you to plan and create project-specific master setups that you can use to create setups for individual tests in ADE Assembler. This enables quick and efficient setup creation at the test level. You can also review the setup at the project level by using the coverage report in ADE Verifier.

  To support this feature, a new assistant, Setup Library, has been provided in ADE Assembler and ADE Verifier. For more information, see *Virtuoso ADE Assembler User Guide* and *Virtuoso ADE Verifier User Guide*.

- **Spectre Interactive Environment**

  The Spectre Interactive Environment tool has been introduced to debug Spectre simulations interactively by using events, triggers, and breakpoints. You can run this tool in standalone mode or from the results of a simulation run in ADE Assembler. For more information, see *Spectre Interactive Environment User Guide*.

Virtuoso Layout Suite Changes

ICADVM18.1 introduced two new applications, Layout Viewer and Layout EXL, which work alongside Layout XL in a revamped Virtuoso Layout Suite.

- Layout Viewer is a new application that provides a viewing-only environment for layout designs of any complexity. It does not allow editing or saving operations or plug-ins or customizations of any kind. See *Virtuoso Layout Viewer User Guide* for more information.

- Layout EXL represents the most advanced editing environment in the Virtuoso Layout Suite, offering the industry's first fully integrated electrically and simulation-driven layout design environment. See *Virtuoso Layout Suite EXL Reference* for more information.

The Layout L, Layout GXL, and Layout EAD applications have been removed from the current release. Although the Layout GXL application has been removed, all Layout GXL technologies remain available to run on top of either Layout XL or the new Layout EXL platform. The Layout GXL flexible token licensing scheme continues as before to enable automatic and assisted layout technologies, such as floorplanning, placement, routing, and so on.

As a result of these changes:

- Layout L flows become Layout XL flows
- Layout XL flows remain unchanged
- Layout GXL flows become Layout XL flows

  The Layout XL and Layout GXL cockpits were identical in previous releases. Layout GXL flexible token licensing can still be used with Layout XL in ICADVM18.1.
- Layout EAD flows change to Layout EXL flows.

  EAD licenses will automatically forward-enable Layout EXL.

Before Launching Virtuoso Layout Suite for the First Time

Before you launch Virtuoso Layout Suite for the first time in the new release, check the following settings, especially if you intend to use either of the new applications, Layout Viewer or Layout EXL.

- License Checkout Order
- New View Type, Application Names, and Application Type
  - Bindkeys
  - User Triggers

License Checkout Order

If your environment uses the `VLSLicenseCheckoutOrder` environment variable to change the default checkout order for layout licenses, the setting must include `EXL` if you want to launch Layout EXL. It must also include `GXL` if you intend to use GXL flexible token licensing when launching Layout XL.

In ICADVM20.1, check the value of `VLSLicenseCheckoutOrder` in your environment. By default, it is set as follows:

```
envGetVal("license" "VLSLicenseCheckoutOrder")
"XL, GXL, EXL"
```

If you see a value other than the default shown above, it means the checkout order has been modified. In this case:

- Consider whether you still need to modify the checkout order and, if so, make sure that the new setting includes `EXL` and `GXL`.
- If your checkout order includes `EAD`, replace it with `EXL`.

New View Type, Application Names, and Application Type

`Layout EXL` has a corresponding new view type (`maskLayoutEXL`) and associated application type (`VLS-EXL`). The table below summarizes the changes (Layout applications only).

	Valid Return Values in ICADVM20.1
View Types	maskLayout maskLayoutXL maskLayoutEXL
Application Names	Layout Viewer Layout XL Layout EXL
Application Types	Layout Virtuoso XL VLS-EXL

If you have any custom SKILL code that uses any of the APIs listed below to query the view type, application name, or application type in Virtuoso, you must update your code to correctly handle the new possible return values for it to continue to work as intended in the new release.

Calls to the following APIs must be modified as suggested:

- `deGetAllApplicationNames(`*`t_rootViewType`*`)`

For a *rootViewType* of "maskLayout", valid return values now include "Layout Viewer", "Layout XL", and "Layout EXL".

"Layout L", "Layout GXL", and "Layout EAD" are no longer returned.

If you pass in "maskLayoutGXL" or "maskLayoutEAD", the API returns nil.

- deGetAllViewTypes()

  Valid return values now include "maskLayout", "maskLayoutXL", and "maskLayoutEXL".

  "maskLayoutGXL" and "maskLayoutEAD" are no longer returned.

- deGetAppInfo(*t_viewType*)

  If you pass in "maskLayoutEXL", returns a DPL with an appName of "VLS-EXL".

  If you pass in "maskLayoutGXL" or "maskLayoutEAD", the API returns nil.

- deGetEditViewType([*w_window*])

  Valid return values for layout windows are now "maskLayout", "maskLayoutXL", and "maskLayoutEXL".

  "maskLayoutGXL" and "maskLayoutEAD" are no longer returned.

- deGetLowerTierApp(*t_appName*)

  Valid return values for layout windows are now "Layout" and "Virtuoso XL".

  If you pass in "VLS-GXL", it treats it "Virtuoso XL" and returns "Layout".

- deGetViewType([*w_window*])

  Valid return values for layout windows are now "maskLayout", "maskLayoutXL", and "maskLayoutEXL".

  "maskLayoutGXL" and "maskLayoutEAD" are no longer returned.

- hiGetAppType(*w_window*)

  Valid return values for layout windows are now "Layout", "Virtuoso XL", and "VLS-EXL".

  "VLS-EAD" and "VLS-GXL" are no longer returned.

Bindkeys

Any bindkeys defined for "VLS-EAD" or "VLS-GXL" no longer work. You must redefine them for a valid application type; for example, "VLS-EXL".

If this is not feasible, contact your Cadence customer support representative for additional options.

User Triggers

If you register user triggers to be called when a particular layout application launches, you must enable these explicitly for the `maskLayoutEXL` view type before running Layout EXL. For example,

```
when(member("maskLayoutEXL" deGetAllViewTypes())
             deRegUserTriggers("maskLayoutEXL" nil nil 'myTriggerfunction)
        )
```

If you are unable to migrate your triggers explicitly to EXL, then the system can do so automatically if you set the `addVLSXLUserTriggersToEXL` CDS environment variable in your `.cdsenv` or `.cdsinit` file:

```
designEditor.appReg addVLSXLUserTriggersToEXL boolean t
```

You can also set this variable during a Virtuoso session by typing the following in the CIW:

```
envSetVal("designEditor.appReg" "addVLSXLUserTriggersToEXL" 'boolean t)
```

All user menu and post-install triggers registered on `maskLayoutXL` are automatically fired when installing Layout EXL.

Note: Triggers registered for view type `maskLayoutEAD` are automatically migrated to `maskLayoutEXL` by the Virtuoso Design Environment.

Products and Releases in Support Mode in ICADVM18.1

Power IR/EM

From ICADVM18.1, Power IR/EM is in support mode. The software is still available for use but will be removed in a future release.

It is recommended that Voltus-Fi-L now be used to perform IR drop and Electromigration (EM) analysis using extracted views.

Voltus-Fi-L provides a number of benefits over the earlier VPS-L product, including:

- Enhanced UI similar to VFI-XL, thus making it easier for customers to switch between VFI-L and VFI-XL
- Enhanced EM engine which will be used with VFI-XL. Key benefits include multi-threading and run-time improvements

Products and Features Removed from ICADVM18.1

The following features are no longer supported. As a result, software and documentation support have been removed.

- CDB to OpenAccess Translator (from ISR13 onwards)
- OSS- and cellview-based netlisting in AMS simulations
  Cadence recommends that you run the AMS Unified Netlister (AMS UNL), the default AMS netlister.
- Schematic Model Generator (SMG)
- Technology file constructs `compactorRules`, `dlrRules`, `lasRules`, `streamLayers`, `cdsMos`, and `cdsRes`
  These constructs are now ignored if present and should be removed from technology files.
- Support for the Power Intent Export Assistant (PIEA) has been removed from Virtuoso Schematic Editor XL.
- Running AMS simulations by using the AMS plugin in the Virtuoso Hierarchy Editor.

  Note: It is recommended to run the AMS simulations from ADE Explorer or ADE Assembler.

Documentation Updates in ICADVM18.1

New Documentation

The following new books were added to the documentation set in ICADVM18.1:

- *Virtuoso Automated Device-Level Layout Flow Guide*
- *Virtuoso Basic Library Reference*
- *Virtuoso Concurrent Layout User Guide*
- *Virtuoso Design Intent User Guide*
- *Virtuoso Design Planner User Guide*
- *Virtuoso Electromagnetic Solver Assistant User Guide*
- *Virtuoso Import Tools User Guide*
- *Virtuoso Layout Viewer User Guide*

- *Virtuoso Layout Suite EXL Reference*
- *Virtuoso Photonics Solution Guide*
- *Virtuoso Simulation Driven Interactive Routing User Guide*
- *Virtuoso System Verilog Netlister User Guide*
- *Virtuoso RF Solution Guide*
- *Spectre Interactive Environment User Guide*

Restructured Documentation

- *Virtuoso Layout Suite*

  To accommodate the enhancements described in Virtuoso Layout Suite Changes, the following user guides have been provided to cover the functionality of the three VLS cockpits:

  - *Virtuoso Layout Viewer User Guide*
  - *Virtuoso Layout Suite XL: Basic Editing User Guide*
  - *Virtuoso Layout Suite XL: Connectivity Driven Editing Guide*
  - *Virtuoso Layout Suite EXL Reference*

  The Layout GXL cockpit is no longer available in ICADVM18.1, so the corresponding *Virtuoso Layout Suite GXL Reference* has been removed.

- *Virtuoso Schematic Editor*

  The user guides for Virtuoso Schematic Editor L and Virtuoso Schematic Editor XL cockpits have been merged into a single book called *Virtuoso Schematic Editor User Guide*.

- *Virtuoso Space-based Router*

  The router documentation has been split into two user guides, one for the automated space-based router functionality and the other covering the interactive and assisted routing commands.

  - *Virtuoso Space-based Router User Guide*
  - *Virtuoso Interactive and Assisted Routing User Guide*

In addition, the Tcl commands and supported constraints that were previously documented in the Finale manuals are now available in the following Virtuoso Space-based Router manuals.

- *Virtuoso Space-based Router Command Reference*
- *Virtuoso Space-based Router Constraints Reference*

Removed Documentation

The documentation for the following products is not available from the ICADVM18.1 release. You can access the related documentation from a previous installation hierarchy, for example, ICADV12.3. Also, the documentation will remain available on Cadence Online Support as long as ICADV12.3 is supported.

- *Cadence Space-based Router and Cadence Chip Optimizer*
- *Virtuoso AMS Designer Environment*
- *Virtuoso Layout Optimize*
- *Virtuoso Layout Suite GXL Reference*
- *Virtuoso Mixed-Signal Flow Guide*
- *Virtuoso Routing IDE*

Help and Support Enhancements

Training Courses

Cadence provides a number of training courses developed to teach a specific tool or design discipline. These courses are delivered through instructor-led live, virtual classes, or online training. You can view the Cadence Learning Maps to understand the recommended course flows according to tool knowledge levels.

To explore the full range of training courses provided by Cadence in your region, visit Cadence Training or contact training locations near you.

Videos

The Virtuoso Videos book provides access to:

- Videos that can be played directly from Cadence Help
- Recommended videos available in the Cadence Video Library
- Training Bytes videos that are part of Cadence Training courses

Blogs

Cadence publishes a number of blogs under different series to highlight several new features of Virtuoso. These blogs are published on cadence.community.com. We recommend that you take a look at these blogs and, if you like what you see, subscribe to the Custom IC Design blog category to have them delivered directly to your mailbox.

We are now also translating some of our blog content to Chinese and publishing them in the Custom IC Design (Chinese) blog category. To get these blogs delivered directly to your mailbox, subscribe to the blog category.

Virtuoso Release Announcement Blog Series

The Release Announcement blogs are published immediately after a new Virtuoso release is available for download. These blogs not only announce new release arrivals but also point you to key features to look out for in the release, related documentation, and other release collaterals.

The blogs in this series are as follows:

- Virtuoso IC6.1.8 ISR13 and ICADVM18.1 ISR13 Now Available (New)
- Virtuoso IC6.1.8 ISR12 and ICADVM18.1 ISR12 Now Available
- Virtuoso IC6.1.8 ISR11 and ICADVM18.1 ISR11 Now Available
- Virtuoso IC6.1.8 ISR10 and ICADVM18.1 ISR10 Now Available
- Virtuoso IC6.1.8 ISR9 and ICADVM18.1 ISR9 Now Available
- Virtuoso IC6.1.8 ISR8 and ICADVM18.1 ISR8 Now Available
- Virtuoso IC6.1.8 ISR7 and ICADVM18.1 ISR7 Now Available

Virtuosity Blog Series

The Virtuosity blogs give you an overview of important and useful Virtuoso features, both old and new, along with supporting links to a range of associated resources including RAKs, videos, application notes, product manuals, and so on.

The latest blogs in this series are as follows:

- Examining Post-Layout Capacitance Using Virtuoso ADE Assembler and ADE Explorer (New)
- In the Line of Veri-Fire - Episode 5 (New)
- Do Rulers Rule Your Layout Designs? (New)
- What's New in Run Plan - Part IV (New)
- In the Line of Veri-Fire - Episode 4 (New)
- In the Line of Veri-Fire - Episode 3 (New)
- In the Line of Veri-Fire - Episode 2

Virtuoso Video Diary Blog Series

The Virtuoso Video Diary blogs cover new and important Virtuoso features and provide links to related video content and other resources.

The latest blogs in this series are as follows:

- The SKILLed Way of Using Plotting Templates (New)

- Enhancements in Reliability Analysis
- Knowledge Booster Training Bytes – Part 2
- Knowledge Booster Training Bytes – Part 1
- Click - Take a Snapshot - Smile!
- schTraceNet, a Simple Solution to Complex Questions!
- Technology File Maintenance Made Easy

Virtuoso Meets Maxwell Blog Series

The Virtuoso Meets Maxwell blogs explore the capabilities and potential of Virtuoso RF and Virtuoso MultiTech.

The latest blogs in this series are as follows:

- Thinking Outside the Chip – Advantages of Interoperability Between Best-In-Class IC and IC Packaging Design and Verification Tools (New)
- Unified Libraries – Making Way For Cross-Platform Flows (New)
- Magic! – Dynamic Voiding in Virtuoso RF Solution (New)
- How Come There is No Mention of Wirebonded ICs? (New)
- Cross-Fabric Electromagnetic Extraction – Eliminating the Tedious Work of Merging IC, Package, and Board
- Full CellView EM Extraction
- Finite Element Can Add Clarity

Spectre Tech Tips Blog Series

Spectre Tech Tips is our blog series that focuses on the capabilities of Spectre® and its integration and interactions with Virtuoso® Analog Design Environment (ADE).

The latest blogs in this series are as follows:

- Spectre Local Options
- Spectre APS Save Overview - Part 2
- Spectre APS Save Overview - Part 1

- Measuring Noise in Digital Circuits
- Spectre Assert and Design Check Overview
- Device Aging? Yes, even Silicon wears out
- Optimizing Spectre APS Performance

Part 1: Application Infrastructure

Cadence Application Infrastructure

No new features or enhancements have been introduced in Cadence Application Infrastructure in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Cadence Application Infrastructure User Guide*

Cadence Library Manager

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Cadence Library Manager User Guide*
- *Cadence Library Path Editor User Guide*

ICADVM18.1 ISR11

What’s New and Enhanced

Environment Variables

cdsLibManager.displayOptions enableDmQuery	New
Enables the querying and retrieval of DM data so that state information can be shown for DM libraries. This information includes any data for the DM system if the extra columns are available.	
cdsLibManager.displayOptions showExtendedStates	New
Allows DM tables to be shown in any viewing mode.	

ICADVM18.1 ISR10

What’s New and Enhanced

Environment Variable

cdsLibManager.open windowBehavior	New
Specifies whether an application always opens a cellview in a new window or brings up a window that already has the cellview open, with the cellview tab as the active one.	

ICADVM18.1 ISR8

What’s New and Enhanced

Environment Variable

CDS_MAX_CELL_NAME_LENGTH	New
Determines the maximum cell name length allowed while creating, copying, or renaming cells.	

ICADVM18.1 ISR3

What’s New and Enhanced

Feature

New View Filter Option	New
A new option has been added in the View filter options menu: *Filter Cells by View Names*. It helps you set the filter to display only those cells that match the filtered views.	

ICADVM18.1 ISR1

What’s New and Enhanced

Feature

Rename Support for Design-Managed Objects	New
The Rename dialog boxes in Library Manager now support renaming design-managed (DM) objects, while preserving their history. To check whether DM rename support is enabled by your design management system, use the `ccpDmHasRename` SKILL function. DM objects can also be renamed using the `ccpDmRename` SKILL function.	

ICADVM18.1 Base Release

What’s New and Enhanced

Features

Viewing the Current Cellview Status	New
You can view the current cellview status by clicking the *Show open cellviews in use* button on the Library Manager toolbar.	
Selecting View Types	New
The Library Manager copy functionality now provides a way to select view types to copy. You can specify view names or view types or a combination of both in copy forms.	

Cadence Library Path Editor

No new features or enhancements have been introduced in Cadence Library Path Editor in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Cadence Library Path Editor User Guide*

Virtuoso Hierarchy Editor

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Hierarchy Editor User Guide*

ICADVM18.1 ISR3

What’s New and Enhanced

Feature

Setting Display Stop Points for maskLayout Databases	New
You can use the `maskLayoutStopLimit` variable in the `hed.env` file to specify a size limit for maskLayout cellview databases. If a configuration includes such databases that are larger than the specified size limit, they will not be opened in Hierarchy Editor and will be shown as leaves with display stop point icons.	

ICADVM18.1 ISR1

What’s New and Enhanced

Feature

View Bindings for Individual Bits	New
You can now set view bindings separately for individual bits of an iterated instance when using the AMS UNL netlister.	

ICADVM18.1 Base Release

What’s New and Enhanced

Feature

Customizing Hierarchy Editor Menus	New
You can now customize Hierarchy Editor menus.	

Cadence SKILL Language

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Cadence SKILL Language User Guide*
- *Cadence SKILL Language Reference*
- *Cadence SKILL Development Reference*
- *Cadence Interprocess Communication SKILL Reference*
- *Cadence SKILL++ Object System Reference*

ICADVM18.1 ISR9

What’s New and Enhanced

SKILL Functions

ipcSleepMilli	New
Causes the parent process to be delayed for the given number of milliseconds.	

ICADVM18.1 ISR2

What’s New and Enhanced

SKILL Functions

removeListDuplicates	New
Removes duplicate entries from a SKILL list and returns a new list with the duplicates removed.	
sstatus	Enhanced
A new variable has been added: `floatPrecisionChars`.	

ICADVM18.1 ISR1

What’s New and Enhanced

SKILL Functions

getMuffleWarnings	New
Returns a list of warnings that were called and suppressed by the preceding muffleWarnings command.	
muffleWarnings	New
Returns the result of the last expression evaluated.	
ilgResetWarningMarker	New
Clears a warning marker on a specified line in the SKILL IDE editor window.	
ilgSetWarningMarker	New
Sets a warning marker on a specified line in the SKILL IDE editor window.	

ICADVM18.1 Base Release

What’s New and Enhanced

SKILL Function

ilgScrollToLocation	New
Scrolls to the specified location in the specified SKILL IDE editor window or tab.	

Cadence SKILL IDE

No new features or enhancements have been introduced in Cadence SKILL IDE in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Cadence SKILL IDE User Guide*

ICADVM18.1 ISR1

What’s New and Enhanced

Features

Additional SKILL Status Options	New
Two additional status options have been added in *Options – SKILL Status – Stack Trace* in the SKILL IDE. These are *warningTrace* and *warningBreak*.	

Cadence User Interface SKILL

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Cadence User Interface SKILL Reference*

ICADVM20.1 Base Release

What’s New and Enhanced

SKILL Functions

perfDiagInstall	New
Installs the *Performance Diagnosis* submenu under the *Tools* menu in CIW.	
hiAddExtraRepeatCommand	New
Adds the specified SKILL command to the extra repeat command list so that the command becomes repeatable next time the `hiRepeat` function is called.	
hiGetExtraRepeatCommands	New
Retrieves all commands added by the `hiAddExtraRepeatCommand` SKILL function in the extra repeat command list.	
hiIsExtraRepeatCommand	New
Checks whether the specified SKILL command is in the extra repeat command list.	

ICADVM18.1 ISR12

What’s New and Enhanced

SKILL Functions

`hiSetDBoxDefaultLocation`	New
The `hiSetDBoxDefaultLocation` SKILL function now supports the *`g_autoUpdate`* optional argument that lets you change the default location of a dialog box.	
`hiDisplayFixedMenu`	New
The `hiDisplayFixedMenu` SKILL function now supports the *`g_staysOnTop`* optional argument that lets you specify whether you want vertical or horizontal fixed menus to stay on top of other windows.	

ICADVM18.1 ISR11

What's New and Enhanced

SKILL Functions

`hiFlushInfo`	New
Processes exposure events to allow the output of the previous program to be displayed in the CIW during the SKILL code processing.	
`enterLine`	Enhanced
The `?target` key argument has been added to let you specify the target point to which a flight line is drawn from the current pointer.	
`changeEnterFun`	Enhanced
The `?target` argument has been added to let you specify the target point for the `enterLine` function.	
`hiPrintToLogFile`	Enhanced
The *`g_comment`* argument has been added to let you add a comment line to the `CDS.log` file.	

ICADVM18.1 ISR10

What’s New and Enhanced

SKILL Functions

hiCreateLayoutForm	Enhanced
The ?initialSize argument can now be used to specify the width and height of the new form. This argument was being ignored earlier.	

ICADVM18.1 ISR8

What’s New and Enhanced

SKILL Functions

hiViewTextFile	New
Creates a viewfile window, inserts the menu bar, sets the window and icon names, specifies how to load the file in the window, and displays the file in the window.	
hiAddIconOverrides	New
Adds names to the list of icons that will be loaded using CSF search results directly, bypassing the hierarchy files. If the CSF search fails, the files in the hierarchy are loaded.	
hiRemoveIconOverrides	New
Removes names from the list of icons that are loaded using the CSF search results directly, bypassing the hierarchy files. If the CSF search fails, the files in the hierarchy are loaded.	
hiGetIconOverrides	New
Returns the list of icon names that will be loaded by the CSF search results directly, bypassing the hierarchy files. If the CSF search fails, the files in the hierarchy are loaded.	

ICADVM18.1 ISR4

What’s New and Enhanced

SKILL Functions

hiInsertBlankCIWOutputPage	New
Inserts a blank page to the CIW output area.	

ICADVM18.1 ISR2

What’s New and Enhanced

SKILL Functions

hiGetFieldAttachments	New
Returns the integer representing the combination of attachment bit values when the field is in an attachment form.	

ICADVM18.1 Base Release

What’s New and Enhanced

SKILL Functions

hiGetCursorByName	New
Returns the ID associated with the specified cursor name.	
hiGetCursorName	New
Returns the name of the cursor associated with the specified cursor ID.	
hiDeleteToolbarBreak	New
Deletes the toolbar break before the specified toolbar from the toolbar area of a window.	
hiHasToolbarBreak	New
Returns a Boolean value indicating whether there was a toolbar break before the specified toolbar.	
hiInsertToolbarBreak	New
Inserts a toolbar break into the toolbar area of a window before the specified toolbar.	
hiPlaceToolbarBreak	New
Adds a toolbar break to the specified toolbar area of a window.	
startCdsenvEditor	New
Starts Cdsenv Editor.	

Part 2: Design Environment

Virtuoso Design Environment

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Design Environment User Guide*

ICADVM20.1 Base Release

Feature

Performance Diagnostic Tool	New
The new Performance Diagnostic tool in the Virtuoso custom IC design platform contains many debugging techniques to isolate issues that might have caused an application to slow down or freeze.	

ICADVM18.1 ISR13

Environment Variables

disablePartialRead	New
Disables partial read in a Virtuoso session. **Note:** This environment variable is available from ICADVM18.1 ISR12.	

ICADVM18.1 ISR12

Environment Variables

fitViewAttempt	New
Specifies the number of attempts made to open a window at a certain zoom level for a graphics Window in Virtuoso.	
optionFormsStayOnTop	New
Specifies whether the option forms should stay on top of all windows, including the non-Virtuoso windows. **Note:** This option is available from ICADVM18.1 ISR11.	
noDetailedRowCol	New
Determines the syntax of mosaic tile objects in the value returned by the dbGetTrueOverlaps SKILL function.	

Documentation Update

CDBA Environment Variables	New
Documentation of CDBA environment variables is now available in the chapter *CDBA Environment Variables* of *Virtuoso Design Environment User Guide*.	

ICADVM18.1 ISR9

SKILL Functions

Save Modified Data Form	New
The Save Modified Data form has been added to CIW. You can use this form to save all currently open and modified cellviews in Virtuoso.	

ICADVM18.1 ISR7

SKILL Functions

setEFunFlightLineTarget	New
Sets a target point for the `enterLine` function so that a flight line is drawn from the cursor to this point.	

ICADVM18.1 Base Release

What’s New and Enhanced

Feature

Support to Edit .cdsenv Files from GUI	New
Cdsenv Editor lets you search for environment variables across multiple .cdsenv files belonging to different tools. The editor allows you to change values of environment variables and save the changes to the specified file.	

Virtuoso Software Licensing and Configuration

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Software Licensing and Configuration User Guide*

ICADVM18.1 ISR12

What’s New and Enhanced

Features

Viewing Solution-based Licenses in the Software Product License Management Form	New
You can now view the licenses checked-out for a solution when you set the specific environment. Currently, you can view the licenses for the Virtuoso Photonics and Virtuoso RF solutions. **Note:** This feature was made available in ICADVM18.1 ISR11 release.	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

New product family added to the Software Product License Management form: Ordering tab	New
A new product family `AMSenv` has been added to the *Ordering* tab of the Software Product License Management form. `AMSenv` represents the AMS product license `AMS_environment`. Related environment variable: `AMSEnvLicenseCheckoutOrder`	

ICADVM18.1 ISR6

What’s New and Enhanced

Environment Variable

CIC_ENABLE_LIC_PERF	New
This environment variable tests the license checkout performance of license features. If the license checkout time exceeds the specified threshold, a warning message is printed to the CIW.	

ICADVM18.1 ISR5

What’s New and Enhanced

Licensing Requirements

adeMaestroCheckoutOrder	New
The checkout order string in `adeMaestroCheckoutOrder` now uses `Classic` instead of `ADE` to refer to ADE L, ADE XL and ADE GXL products.	

ICADVM18.1 ISR3

What’s New and Enhanced

Licensing Requirement

Licensing for Virtuoso Photonics Platform	New
To enable photonics flows, you require either the new `Virtuoso_Photonics_Platform` or the existing `Virtuoso_Photonics_Option` license. The `Virtuoso_Photonics_Platform` license (License Number 95551) is enabled on setting the `Virtuoso_Photonics_Platform` shell variable.	

ICADVM18.1 Base Release

What’s New and Enhanced

Licensing Requirements

Virtuoso MultiTech Framework	New
Virtuoso MultiTech Framework elevates the Virtuoso platform to tackle more system design challenges by expanding physical implementation options from IC design to Module and Package designs. The new MultiTech Framework provides access to all the functionality and features of Virtuoso System Design Platform (IC6.1.7) through the *Module* menu in Virtuoso Schematic Editor XL and Virtuoso Layout Suite EXL. Additionally, the Virtuoso MultiTech Framework provides the minimum base framework required for fully integrated access to RF and Photonics system design tools and flows. To enable Virtuoso MultiTech Framework, you need to set either of the following shell environment variables: ■ `Virtuoso_MultiTech` ■ `Virtuoso_RF_Option` After setting one of the above shell environment variables, when you start Virtuoso, the following licenses are checked out: ■ Cadence Design Framework II (Product Number 111) ■ `Virtuoso_Adv_Node_Framework` (Product Number 95011) ■ `Virtuoso_MultiTech_Framework` (Product Number 95022)	
Virtuoso Layout Suite EXL	New

In ICADVM18.1, a new Virtuoso Layout Suite EXL design cockpit has been introduced.

The new EXL product tier provides access to all the functionality and features of Virtuoso Layout Suite EAD (ICADV12.3) for advanced electrically aware designs. Additionally, Layout EXL product tier provides access to all new electrical or simulation driven layout and routing, as well as new concurrent layout editing.

Note: If you are using Layout EAD application in flows in ICADV12.3, you should migrate to the Layout EXL flow in ICADVM18.1. For more information on enabling the EAD workspace and engines within Layout EXL, see *Virtuoso Electrically Aware Design Flow Guide*

Layout EXL is also the minimum base application required for fully integrated access to layout tools and flows facilitating advanced layout design at 5nm process nodes, and design planning and congestion analysis.

Virtuoso Layout Suite EXL requires the `Virtuoso_Layout_Suite_EXL` (95800) license. Additionally, you can use the Virtuoso Layout Suite EAD license to start Virtuoso Layout Suite EXL in ICADVM18.1.

Virtuoso RF Solution	New

ICADVM18.1 introduces Virtuoso RF solution, which is a complete set of integrated software tools that enable you to create RF module and package layouts in the Virtuoso environment, and run parasitic-aware system-level simulations early in the design cycle.

To enable the Virtuoso RF functionality, you need to set the `Virtuoso_RF_Option` shell environment variable.

After setting the `Virtuoso_RF_Option` environment variable, when you start Virtuoso, the following licenses are checked out:

- Cadence Design Framework II (Product Number 111)
- `Virtuoso_Adv_Node_Framework` (Product Number 95011)
- `Virtuoso_MultiTech_Framework` (Product Number 95022)

When you launch the first layout EXL window, the `Virtuoso_RF_Option` license is checked out.

Virtuoso Layout Viewer	New

Virtuoso Layout Viewer is a new application to open a layout cellview for viewing only. This application is available from the *Library Manager* and *File Open* dialog boxes.

It requires the following licenses:

- Cadence Design Framework II (Product Number 111)
- `Virtuoso_Adv_Node_Framework` (Product Number 95011)

For more information, see *Virtuoso Layout Viewer User Guide*

Software Product License Management Form- Ordering Tab	Enhanced

The *Ordering* tab of the Software Product License Management form has been revamped. This tab provides an interface to control the license checkout order of the Schematic, Layout, ADE, and Maestro product families. These product families are now listed in a tree structure.

The *Use next license for tier* radio buttons have been replaced by the *UseNext* drop-down list. The updated graphical user interface (GUI) has the *Current Checkout Order* and *Available Tiers* panes to view and control the checkout order of products.

License Tool Partition Updates	Enhanced

The tool partition for environment variables, `VIVALicenseCheckoutOrder`, `VIVA_UseNextLicense`, `adeMaestroCheckoutOrder`, has now been changed to `license`. However, the `viva.application`, `maestro`, and `asimenv` partitions are still supported for these variables.

Environment Variables

VLSLicenseCheckoutOrder	Enhanced
The default checkout order of `VLSLicenseCheckoutOrder` variable has been updated to `"XL, GXL, EXL"` If your environment uses the `VLSLicenseCheckoutOrder` environment variable to change the checkout order for layout licenses, the setting should include `"EXL"` in order to launch Layout EXL. **Note:** If you are using Layout EAD application in flows in the IC6.1.7 or ICADV12.3 releases and are migrating to Layout EXL flows in ICADVM18.1, replace `EAD` with `EXL` in the `VLSLicenseCheckoutOrder` environment variable settings.	
maestroCheckoutOrder	New
A new environment variable, `maestroCheckoutOrder`, has been introduced to replace the existing `checkoutOrder` variable in the `maestro` partition. However, the `checkoutOrder` variable is still supported.	

SKILL Function

eliCheckoutOutAndLock	New
Checks out and locks the specified license feature. Once the specified license feature is locked, it cannot be checked in until the session ends. Use this function in a scripted batch process to reduce the runtime due to licensing.	

Removed and Changed Features

Virtuoso Layout Suite L	Removed
The Layout L application has been removed from the Layout product family. In ICADV12.1 and later releases, the Advanced Node Option for Layout required a minimum Virtuoso Layout Suite XL license. However, the Layout L application was still supported for flows that required additional performance. With improvements in Layout XL performance and the addition of new flexible configuration of Layout XL in ICADVM18.1, which provides additional performance for supporting additional flows, the Layout L application has been removed from the *Launch* menu and associated *File Open* dialog boxes. **Note:** If you are using Layout L flows in ICADV12.3, you should migrate to Layout XL flows in ICADVM18.1. For more information, see *Virtuoso Layout Suite XL: Basic Editing User Guide* and *Virtuoso Layout Viewer User Guide*	
Virtuoso Layout Suite GXL	Removed
The Layout GXL application has been removed from the Layout product family. In IC6.1.0 and ICADV12.1 and later releases, the Layout GXL application provided an identical layout environment to Layout XL, with the exception of using `Virtuoso_Layout_Suite_GXL` job-based and token-based licenses, as opposed to the `Virtuoso_Layout_Suite_XL` UHD license used by Layout XL. With the ability to have license search paths in subsequent releases of Virtuoso, the identical Layout GXL application has been removed from the *Launch* menu and associated *File Open* dialog boxes. However, the `Virtuoso_Layout_Suite_GXL` job-based and token-based licenses can still be used within the license search order of Layout XL. Additionally, the `Virtuoso_Layout_Suite_GXL` job-based and token-based licenses are still required for all layout automation features like floorplanning, placement, and routing. **Note:** If you are using Layout GXL application in flows in IC6.1.7 or ICADV12.3, you should migrate to Layout XL flows in ICADVM18.1.	

Virtuoso Layout Suite EAD	Removed

The Layout EAD application has been removed from the Layout product family.

All the functionality and features of Virtuoso Layout Suite EAD (ICADV12.3) for advanced electrically aware design have been moved to a new product, Virtuoso Layout Suite EXL. Additionally, the Layout EXL product tier provides access to all new electrical or simulation driven layout and routing, as well as new concurrent layout editing.

Note: If you are using the Layout EAD application in flows in the ICADV12.3 release, you should migrate to the Layout EXL flow in ICADVM18.1. For more information on enabling the EAD workspace and engines within Layout EXL, see *Virtuoso Electrically Aware Design Flow Guide*

You can use any one of the following licenses to use EAD Browser:

- `Virtuoso_Layout_Suite_EAD` (Product Number 95600)
- `Virtuoso_Layout_Suite_EXL` (Product Number 95800)

  Note: If 95800 is already checked out, 95600 is not used.

Virtuoso Design Environment SKILL

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Cadence SKILL Language User Guide*
- *Cadence SKILL Language Reference*
- *Virtuoso Design Environment SKILL Reference*
- *Cadence Interprocess Communication SKILL Reference*
- *Cadence SKILL++ Object System Reference*

ICADVM20.1 Base Release

What’s New and Enhanced

SKILL Function

dbGetPowerDomainClusters	New
Retrieves SKILL list of clusters in the specified cellView. Only clusters with clusterBoundary having power domain attribute are returned. Each entry contains the cluster name, cluster boundary name and voltage value.	
dbLayerSize	Enhanced
A new argument, `conicSides`, has been added.	

ICADVM18.1 ISR12

What’s New and Enhanced

Features

Support Added for the placeCells rowRegionSpec Attribute	New
The rowRegionSpec object now supports the placeCells attribute, which is a DPL list.	

SKILL Functions

pcdbOpen	Enhanced
The following new optional arguments have been added: `nameSpace` and `dontOpenNextLevel`.	
dbRowRegionPointsCutOut	New
Changes the rectilinear specification of the specified row region by cutting out the portion of it defined by the specified box.	
dbCreateNamedSubNet	New
Creates a subnet with the specified name.	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

VRF Package Infrastructure Functions	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) A new set of SKILL functions has been added to represent the Allegro component definition objects in Virtuoso.	

SKILL Functions

dbCreateMultipleCurvedPolygons	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Creates multiple curved polygons in the specified cellview on the specified layer or layer-purpose pair using the specified points.	
tokenPrintCapabilities	Removed
The `tokenPrintCapabilities` function was made obsolete in the ICADVM18.1 base release. Calls to this SKILL function are silently ignored.	

ICADVM18.1 ISR9

What’s New and Enhanced

SKILL Functions

ddRegHiddenCellsFunc	New
Registers a user-defined SKILL function that will be called to obtain a list of hidden cell names.	
ddUnregHiddenCellsFunc	New
Unregisters a hidden cells function that was registered using `ddRegHiddenCellsFunc`.	
ddsHiSaveData	New
Displays the Save Modified Data form that prompts you to select one or more objects to save from virtual memory.	
dbSetTermMustConnectAllPins	New
Indicates whether to set the `mustConnectAllPins` attribute on the specified terminal.	
dbTermHasMustConnectAllPins	New
Checks whether the `mustConnectAllPins` attribute value is set on the specified single-bit terminal.	

ICADVM18.1 ISR8

What’s New and Enhanced

SKILL Functions

deBeginConfigurePlugins	New
Lets applications issue a series of `deInstallPlugin` and `deRemovePlugin` calls to add or remove plugins in the specified application window. Using `deBeginConfigurePlugins` before adding or removing plugins helps in reducing calls to the `hiApplyWinConfig` SKILL function.	
deEndConfigurePlugins	New
Indicates an end to a series of calls for `deInstallPlugin` and `deRemovePlugin` marked by the `deBeginConfigurePlugins` in the same application window.	

ICADVM18.1 ISR6

What’s New and Enhanced

SKILL Functions

xdvReplaceInstWithVias	Enhanced
Detects and replaces the via-like instances with standard vias in the existing design across the hierarchy of the given library, cell, and view. **Note:** The function `dbReplaceInstWithVias` has been renamed to `xdvReplaceInstWithVias` in ICADVM18.1 ISR6 release.	

ICADVM18.1 ISR4

What’s New and Enhanced

SKILL Functions

dbCreateGenViaVariant	New
Creates a genViaVariant object in the specified cellview according to the specified viaDef and parameter list.	
dbIsCellViewGenViaMaster	New
Checks whether the specified cellview ID is of a genViaMaster object.	
dbIsGenVia	New
Checks whether the specified database object is a genVia object.	
dbIsGenViaHeader	New
Checks whether the specified database object is a genViaHeader object.	
dbIsGenViaVariant	New
Checks whether the specified database object is a genViaVariant object.	
dbSetGenViaParamDefault	New
Sets the specified parameter to its default value based on the given genVia ID and parameter name.	
dbGetMarkerName	New
Retrieves the name assigned to the marker associated with the given marker ID.	
dbSetMarkerName	New
Assigns the specified name to the marker associated with the given marker ID. The name can be an empty string.	

ICADVM18.1 ISR3

What’s New and Enhanced

SKILL Functions

dbGetRegionBoundaryLayer	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Returns layer name or number of the specified region boundary.	
dbGetRegionBoundaryLayerType	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Returns region boundary layer type.	
dbGetRegionBoundaryName	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Returns region boundary name.	
dbIsRegionBoundary	New
Confirms if the specified area boundary is a region boundary.	

ICADVM18.1 ISR2

What’s New and Enhanced

Features

Occurrence Property Functions	New
You can use the occurrence property SKILL functions to work with the occurrence properties of designs.	

SKILL Functions

dbCreateCurvedAreaBoundary	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Creates a curved area boundary in the specified cellview, using the specified points.	
dbGetCurvedAreaBoundaryPoints	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Returns curved area boundary points for the specified curved area boundary object.	
dbSetCurvedAreaBoundaryPoints	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Sets curved points for the specified curved area boundary object.	

ICADVM18.1 ISR1

What’s New and Enhanced

SKILL Functions

dbCreateCurvedPath	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Creates a curved path that has a straight centerline and round edges at all segments.	
dbCreateCurvedPolygon	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Creates a curved polygon.	
dbIsCurved	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Checks whether the specified shape is a curved shape.	
dbGetCurvedPolygonPoints	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Retrieves the curved point list for the specified curved polygon.	
dbSetCurvedPolygonPoints	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Sets the given curved point list for the specified curved polygon.	
dbConvertAnyAngleTransformFromDbToMY	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) Converts orientation-based representation of an orthogonal transformation into an MY-based representation. Orientation-based representation is a pair of DB orientation and any-angle rotation, where the rotation angle is applied first. The MY-based representation is a pair of any-angle rotation and an optional reflection over the Y-axis with the reflection applied first.	

ICADVM18.1 Base Release

What's New and Enhanced

SKILL Functions

dbGetShapeColorLockType	New
Returns the color lock type assigned to the specified shape. If the shape is not locked, nil is returned. By default, "user" lock type is assigned to locked shapes.	
dbSetShapeColorLockType	New
Sets the color lock type for the specified shape.	
dbGetViaLayerColorLockType	New
Returns the color lock type assigned to the specified via layer.	
dbSetViaLayerColorLockType	New
Sets the color lock type for the specified via layer.	
dbCellViewHasUnknownColorLockChanges	New
Checks whether any edits were done to the specified type of objects in the older versions of Virtuoso after color lock type information was set for some of them.	
dbGetShapeEffectiveColorLockType	New
Returns the effective color lock type of the specified occurrence shape.	
dbCellViewResetAllColorLockTypes	New
Changes color lock type for all color-locked objects of the specified object types to the given value.	
dbUnabutGroup	New
Unabuts members of an abutment group.	
dbReplaceInstWithVias	New
Detects and replaces the via-like instances with standard vias in the existing design across the hierarchy of the given library, cell, and view. **Note:** This function has been renamed to xdvReplaceInstWithVias in ICADVM18.1 ISR6 release.	
dbOpenParamCellView	New

Opens submaster cellview of the Pcell supermaster specified by the library, cell, and view names and the parameter values.	
dbFeaturePrintInfo	New
Lists information about the features that are being used in the specified cellview.	
dbSetAutoSaveCallback	New
Enables a user-defined callback function to control the frequency at which large designs are auto-saved.	
dbIsInstTransparent	New
Checks if the specified instance ID is set to be transparent to the Layout XL binder.	
dbSetInstTransparent	New
Sets an instance as transparent to the Layout XL binder.	
ddGetDisplayValue	New
Returns the value of the DISPLAY attribute assigned to a library.	
dbIsMultiTechEnabled	New
Confirms whether the Virtuoso Multi Tech Framework functionality is enabled in the current session.	
dbIsVRFEnabled	New
Confirms whether the VRF-related functionality is enabled in the current session.	
dbIsVRFInfraEnabled	New
Confirms whether the VRF infrastructure sub-system is enabled in the current session.	

Cadence Integrator's Toolkit Database

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Cadence Integrator's Toolkit Database Reference*

ICADVM20.1 Base Release

What’s New and Enhanced

C Functions

dbFlattenInst2	Enhanced
New argument *excludeObj* has been added to this C function. It specifies a dbExcludePCells object. **Note:** This argument is available from ICADVM18.1 ISR12.	
dbCreateExcludePCellsFromFile	New
Creates a dbExcludePCells object from the input ExcludePCells file. **Note:** This C function is available from ICADVM18.1 ISR12.	
dbExcludePCellsContain	New
Checks whether the cellview ID of specified Pcells is to be excluded during Pcell flattening. **Note:** This C function is available from ICADVM18.1 ISR12.	
dbDeleteExcludePCells	New
Deletes the dbExcludePCells object. **Note:** This C function is available from ICADVM18.1 ISR12.	

ICADVM18.1 ISR12

What’s New and Enhanced

C Functions

dbGetRowRegionSpecPlaceCells	New
Returns the placement cells attributes of the specified row region specification.	
dbSetRowRegionSpecPlaceCells	New
Sets the placement cells attributes for the specified row region specification.	
dbFreePlaceCellsListAttr	New
Deallocates the memory allocated for an array of placement cells objects.	
dbRowRegionPointsCutOut	New
Changes the rectilinear specification of the specified row region by cutting out the portion of it defined by the specified box.	
dbCreateNamedSubNet	New
Creates a subnet with the specified name.	

ICADVM18.1 ISR12

What’s New and Enhanced

C Functions

dbGetRowRegionSpecPlaceCells	New
Returns the placement cells attributes of the specified row region specification.	
dbSetRowRegionSpecPlaceCells	New
Sets the placement cells attributes for the specified row region specification.	
dbFreePlaceCellsListAttr	New
Deallocates the memory allocated for an array of placement cells objects.	
dbRowRegionPointsCutOut	New
Changes the rectilinear specification of the specified row region by cutting out the portion of it defined by the specified box.	
dbCreateNamedSubNet	New
Creates a subnet with the specified name.	

ICADVM18.1 ISR9

What’s New and Enhanced

C Functions

dbSetTermMustConnectAllPins	New
Indicates whether to set the `mustConnectAllPins` attribute on the specified terminal.	
dbTermHasMustConnectAllPins	New
Checks whether the `mustConnectAllPins` attribute value is set on the specified single-bit terminal.	

ICADVM18.1 ISR6

What’s New and Enhanced

Feature

Curvy Core Functions	New
(ICADVM18.1 Photonics Only) Curvy shapes are building blocks of photonic devices such as ring resonators, waveguide bends, and Bragg gratings. You can use these C functions to get information related to existing CurvyCore objects.	

Virtuoso Technology Data

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Technology Data Constraints Reference*
- *Virtuoso Technology Data ASCII Files Reference*
- *Virtuoso Technology Data SKILL Reference*
- *Virtuoso Technology Data User Guide*

Important

For constraint documentation, see *Virtuoso Technology Data Constraints Reference*. This reference manual contains information that was originally covered in *Virtuoso Technology Data ASCII Files Reference* in the chapter named "Constraint Groups and Constraints".
For information about constraint-to-application mapping, see Constraint Groups and Constraints.

ICADVM20.1 Base Release

What's New and Enhanced

Constraints

directionalEolKeepout	New
(ICADVM20.1 Only – 95511 and 95800) Specifies the extension of a wire of a specific width in a specific direction.	
eolViaKeepout	New
(ICADVM20.1 Only – 95511 and 95800) Specifies a keepout region against a cut.	
lineEndGap	New
(ICADVM20.1 Only – 95511 and 95800) Specifies that two perfectly aligned wires with the specified spacing can be an exemption or a triggering condition on some other constraints that use the `exceptLineEndGap` parameter.	
requiredEndOfLineShape	New
(ICADVM20.1 Only – 95511 and 95800) Specifies the spacing at which at least one end-to-end shape must be present on at least one end of a rectangular shape.	
allowedCutClass	Enhanced
(ICADVM20.1 Only – 95511) The following new parameters have been added: `minVoltage` and `maxVoltage`.	
boundaryEOLBlockage	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: `prl`, `within`, and `distance`.	
coreEOLBlockage	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameter has been added: `sideExtension`.	
libProp	Enhanced
The following set of new parameters has been added: `int` | `float`.	

endOfLineKeepout	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: eolSpacingRange, otherEolSpacingRange, jointAsEol, checkOtherWidth, and equalRectWidth.	
minConvexCornerSpacing	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: lowCornerSpacing and middleWires.	
minEndOfLineEdgeExtension	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: minVoltage and maxVoltage.	
minQuadrupleExtension	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameter has been added: lineEndGapOnlyTable. (ICADVM20.1 Only – 95511) The following new parameter has been added: useMinWidth.	
minSideSpacing (One layer)	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: otherLayer, cutClass, and cutWithin.	
minStepEdgeLength	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: jogEdge, adjJogEdge, adjNotchEdge, and adjEolSpacingRange.	
minViaGroupSpacing	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: mergedCuts and otherLayer.	
minViaSpacing (One layer)	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: exceptGap and exceptSameMetalOverlap.	
rectShapeDir	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: exceptWidth and widthHorzontal | widthVertical.	
sameMetalAlignedCuts	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: noViaMetalOverlap and otherLayer.	

trimMetalTrack	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameter has been added: halfInFiller.	
viaEdgeType	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: sameSide and noMetalLowWithin.	
viaGroup	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following set of new parameters has been added: disallowHorizontal | disallowVertical.	
viaSpacing	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameter has been added: width.	

ICADVM18.1 ISR13

What’s New and Enhanced

Constraints

maxDensityCheck	New
(ICADVM18.1 Only – 95511) Specifies the absolute maximum density of shapes on a layer against a specified checking window.	
maxRelativeColorDensity	New
(ICADVM18.1 Only – 95511) Specifies the relative maximum density of shapes on MPT layers against a specified checking window.	
minDensityCheck	New
(ICADVM18.1 Only – 95511) Specifies the absolute minimum density of shapes on a layer against a specified checking window.	
minRelativeColorDensity	New
(ICADVM18.1 Only – 95511) Specifies the relative minimum density of shapes on MPT layers against a specified checking window.	
minCornerToCornerDistance	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: `extensionLength` and `oppositeDirection`.	
minOppExtension	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new set of parameters has been added: `widthHorizontal` and `widthVertical`.	

ICADVM18.1 ISR12

What's New and Enhanced

Features

Loading a Technology File into a Technology Library	New
A new toolbar button, *Load Technology File*, has been added to the Techfile IDE. You can use this button to load the active technology file into an existing technology library.	

Constraints

antennaRatio	New
Defines antenna ratios on a per-layer basis.	
cumPerLayerAntennaRatio	New
Defines cumulative antenna ratios on a per-layer basis.	
libProp	New
Specifies a value for a LEF library property.	
preColoredLayers	New
(ICADVM18.1 Only – 95511) Lists precolored layers or layer-purpose pairs in the `VirtuosoMPTSetup` constraint group.	
allowedSpacingRanges (One layer)	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `exceptWidthEdge`.	
maxLength	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `width`.	
maxViaArrayClusterSize	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `checkLengthOfAnyArray`.	

mergedViaCornerToCornerSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) ■ The following new parameters have been added: `lengthRange` and `otherLengthRange`. ■ The following parameter is now mandatory: `mergeSpacing`.	
minEndOfLineSpacing	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `negativePRL`.	
minEndOfLinePerpSpacing	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `negativePRL`.	
minNotchSpacing	Enhanced
(ICADVM18.1 Only – 95511) The following set of parameters has been added: `horizontal | vertical`.	
minViaSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511) The following new parameters have been added: `prlRange`, `viasOnSameNet`, `viasOnSameMetal`, `viasOnSameVia`, `exceptViasOnSameNet`, `exceptViasOnSameMetal`, and `exceptViasOnSameVia`.	
minViaSpacing (Two layers)	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `prlRange`.	
viaGroup	Enhanced
(ICADVM18.1 Only – 95511) The following set of parameters is now optional: `prlRangeOne` and `prlRangeTwo`.	

ICADVM18.1 ISR11

What's New and Enhanced

Constraints

endOfLineKeepout	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following set of parameters has been added: `horizontal | vertical`.	
mergedViaCornerToCornerSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) ■ The following new parameters have been added: `tripletCenterViaLarger`, `tripletCenterViaSpacingX`, and `tripletCenterViaSpacingY`. ■ The following parameter is now optional: `mergeSpacing`.	
minClusterSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: `enclosingLayer` and `viaEdgeType`.	
minCornerSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `cornerToCorner`.	
minViaGroupSpacing	Enhanced
The following parameter is now optional: `cutClass`.	
minViaSpacing (One layer)	Enhanced
■ The following new parameters have been added: `insideLayers`, `insidePurposes`, `outsideLayers`, and `outsidePurposes`. ■ (ICADVM18.1 Only – 95512) The following new parameters have been added: `exceptSameNet` and `exceptSameMetal`. ■ (ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `deltaVoltage`.	
minViaSpacing (Two layers)	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `deltaVoltage`.	

viaGroup	Enhanced
The following parameter is now optional: `cutClass`.	

Environment Variable

AlternateFoundryCG	New
(ICADVM18.1 Only – 95511) Specifies the name of the constraint group that contains the foundry constraints.	

SKILL Functions

Finger Definition Functions	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) The following functions help you create, delete, and find finger definitions: ■ techCreateFingerDef ■ techDeleteFingerDef ■ techFindFingerDef	

Technology File Sections

fingerDefinitions	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) Specifies a finger definition.	

ICADVM18.1 ISR10

What’s New and Enhanced

Constraints

minViaSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: `otherCutClass` and `otherViaEdgeType`.	

ICADVM18.1 ISR9

What's New and Enhanced

Constraints

antennaModelCombo	New
(ICADVM18.1 Only – 95511) Defines the antenna oxide model to use for nets that are connected to devices with at least one each of a list of other antenna oxide models.	
cutClassPreference	New
(ICADVM18.1 Only – 95511) Specifies an ordered list of cut classes.	
forbiddenSpacingRanges	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `ignoreMiddle`.	
minParallelWithinViaSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `otherWidth`.	
minSideSpacing (Two layers)	Enhanced
(ICADVM18.1 Only – 95511) The following new parameters have been added: `deltaVoltage` and `prlRange`.	
minViaSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511) The following new set of parameters has been added: `shortSideToShortSide | shortSideToLongSide | shortSideToAnySide | longSideToLongSide | longSideToAnySide | anySideToAnySide`	
minViaSpacing (Two layers)	Enhanced
The following new sets of parameters have been added: ■ `insideLayers` and `insidePurposes` ■ `outsideLayers` and `outsidePurposes`	

ICADVM18.1 ISR8

What’s New and Enhanced

Constraints

trimMinCutSpacing	New
(ICADVM18.1 Only – 95511 and 95800) Specifies the spacing between a trim and a cut.	
minCutRoutingSpacing (Two layers)	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: deltaVoltage.	
sameMetalAlignedCuts	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: cutClassSetList.	
viaGroup	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: allowBend.	

ICADVM18.1 ISR7

What's New and Enhanced

Constraints

coreEOLBlockage	New
(ICADVM18.1 Only – 95511 and 95800) Specifies the size and direction of end-of-line blockages from core edges.	
overlapViaGroup	New
(ICADVM18.1 Only – 95511 and 95800) Specifies the maximum number of cuts in a via group, where the spacing between cuts is within a specified range.	
endOfLineKeepout	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: `exceptSpacing`, `withinCut`, `cutLayers`, and `exceptEolSpacing`.	
minAdjacentFourViaSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: `exceptOtherWidthWithCut` and `otherLayer`.	
minEndOfLineEdgeExtension	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `exceptSpacing`.	
minQuadrupleExtension	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: `otherWithinWidthTable`, `otherWithinWidthHorizontal` | `otherWithinWidthVertical`, and `otherSideTable`.	
minSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `intersectLayers`.	
minViaGroupSpacing	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `groupToGroup`.	
minViaSpacing (Two layers)	Enhanced

(ICADVM18.1 Only – 95511) The following new parameters have been added: `horizontal | vertical`, `prl`, and `oppositeSides`.	
`sameMetalAlignedCuts`	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `withCut`.	
`viaEdgeType`	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: `metalWithin`, `noMetalWithin`, `horizontalEdge | verticalEdge`, and `exceptSpacing`.	

SKILL Functions

`techCreateGenViaDef`	New
Creates a generated via definition in the specified technology database.	
`techCreateGenViaVariant`	New
Creates a generated via variant matching the specified generated via definition in the specified technology database.	

Technology File Sections

`cdsGenViaVariants`	New
Specifies generated via variants and the default parameter values to use for all instances of the generated via variants created from a specified generated via definition.	

ICADVM18.1 ISR6

What’s New and Enhanced

Features

Rule Editor Supports techDerivedLayers	Enhanced
The Rule Editor now supports `techDerivedLayers`. For information about the Rule Editor, see Working with the Rule Editor.	

Constraints

`minSpanLengthSpacing`	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `diffMask`.	

Environment Variables

`defaultAttachTech`	New
Specifies the technology library to be selected by default when the user opts to attach a new library to an existing technology library.	

ICADVM18.1 ISR5

What’s New and Enhanced

Constraint

minViaSpacing (Two layers)	Enhanced
The following new parameter has been added: `overlapNotAllowed`.	

ICADVM18.1 ISR4

What's New and Enhanced

Constraints

maxViaStack	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `noCutClassList`.	
minOppEndOfLineExtension	Enhanced
(ICADVM18.1 Only – 95511) The following set of parameters has been added: `horizontal | vertical`.	
minOppExtension	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `otherExtension`.	
minQuadrupleExtension	Enhanced
(ICADVM18.1 Only – 95511) The following new parameter has been added: `includeAbuttedTable`.	
minVoltageSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511) The following set of parameters has been added: `horizontal | vertical`.	
sameMetalAlignedCuts	Enhanced
(ICADVM18.1 Only – 95511) The following new parameters have been added: `parallelEdgeWithin` and `cutWithin`.	

Environment Variable

noTechUpRev	New
Disables the upgrade process that usually takes place when a technology library is opened for the first time in a newer version of Virtuoso. This can be used to prevent the insertion of Virtuoso system-reserved LPPs into technology libraries created outside Virtuoso.	

SKILL Function

techCreateRelatedSnapPatterns	Enhanced
(ICADVM18.1 Only–95511) The following new arguments have been added: *l_extraLPP*, *f_regionSnapPitchHorizontal*, and *f_regionSnapPitchVertical*.	

Technology File Sections

functions	Enhanced
The following new function has been added: specialCut.	
relatedSnapPatterns	Enhanced
(ICADVM18.1 Only–95511) The following new attributes have been added: extraLPP, regionSnapPitchHorizontal, and regionSnapPitchVertical.	
techDisplays	Enhanced
The following new argument has been added: description.	

ICADVM18.1 ISR3

What’s New and Enhanced

Feature

Enhanced Dump Technology File Form	Enhanced
The Dump Technology File form has been enhanced to include a new check box, *Open in Techfile IDE*, to automatically open a dumped technology file in the Techfile IDE.	

Constraint

`minVoltageExtension`	Enhanced
(ICADVM18.1 Only-95511) The following set of parameters has been added: `horizontal | vertical`.	

SKILL Function

`techGetProcessNode`	New
Returns the local processNode value of the specified technology database.	

ICADVM18.1 ISR2

What's New and Enhanced

Features

Types Field for Attach and Reference Forms	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) A new field, *Types*, is included in the Attach Technology Library to Design Library and Reference Existing Technology Libraries forms to limit the list of libraries by a type. For more information, see the following topics in *Virtuoso Technology Data User Guide*: ■ Attaching a Technology Library to a Design Library ■ Creating a New Technology Library that References Existing Technology Libraries	
Ignoring Undefined techParams	New
A new variable, `ignoreUndefTechParams`, is now provided in the Techfile IDE `.cdsenv` file to ignore undefined techParams.	

Constraints

`minIsolatedArea`	New
Specifies the minimum area required for shapes when no other shape is in a defined region.	
`minOppositeClearance (Two layers)`	New
(ICADVM18.1 Only–95511) Specifies the cut spacing between a cut of any cut class on a layer and two cuts of any cut class on a second layer.	
`pinConnectBlockage`	New
(ICADVM18.1 Only – 95511 and 95800) Specifies a blockage area when a wire is connected to any pins on a layer.	
`shieldMinLength`	New
Specifies whether to shield the route segment if the length of the segment is greater than or equal to the specified length.	
`shieldTerminal`	New

Specifies whether all terminal shapes of the shielded net are shielded while adding shield wires.	
voltageMode	New
Specifies the method of calculating the voltage of a net.	
endOfLineKeepout	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: exceptSameSideMetal and exceptNoOppSideMetal.	
maxCumAntenna	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: lowerLayer and upperLayer.	
maxViaArrayClusterSize	Enhanced
(ICADVM18.1 Only–95511) The following new parameters have been added: neighborDistanceRange, maxFacingEdgeNeighbors, and checkEachSideNeighborsSeparately.	
minAdjacentFourViaSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: mergedCuts, minCorner, minInnerCorner, maxInnerCorner, exceptOverlap, exceptMinLength, and cutWithin.	
minDirectionalSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: excludeCutEdge and minNumCuts.	
minEndOfLineEdgeExtension	Enhanced
The following new parameters have been added: horizontalEdge, verticalEdge, and allSides.	
minQuadrupleExtension	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameters have been added: otherMask1, otherMask2, otherMask3, and exactZero.	
minSideSpacing (One layer)	Enhanced
(ICADVM18.1 Only–95511) The following new parameters have been added: otherFacingEdgeRange, facingEdgeHorizontal, and facingEdgeVertical.	
minSpacing (One layer), minSpacing (Two layers), minWidth	Enhanced

(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: `length`.	
`minViaGroupSpacing`	Enhanced
(ICADVM18.1 Only–95511) The following new parameters have been added: `sameMask`, `vertical`, `horizontal`, and `cutPrl`.	
`minViaSpacing (One layer)`	Enhanced
(ICADVM18.1 Only–95511) The following new parameters have been added: `includedEdgeAligned`, `alignedHorizontal`, and `alignedVertical`.	
`onGridOnly`	Enhanced
(ICADVM18.1 Only–95511) The following new parameter has been added: `exceptWidth`.	
`tieShield`	Enhanced
The following new parameters have been added: `shieldTieFreq`, `shieldTieFreqValue`, and `shieldRedundantVia`.	
`viaGroup`	Enhanced
(ICADVM18.1 Only–95511) The following new parameter has been added: `sameMask`.	

SKILL Functions

`techID`	Enhanced
The following new attribute has been added: `needRefresh`.	
`techCreateWaveguideDef`	Enhanced
(ICADVM18.1 Photonics Only) The following new parameter has been added: `maxTaperAngle`.	

Technology File Sections

`waveguideTemplates`	Enhanced
(ICADVM18.1 Photonics Only) The following new argument has been added: `maxTaperAngle`.	

ICADVM18.1 ISR1

What's New and Enhanced

Features

Rule Editor Supports Statements	New
The Rule Editor in the Techfile IDE now helps you view and edit these statements: `techPurposes`, `techLayers`, `techLayerPurposePriorities`, and `techDisplays`.	
Monotonic Violation Checking in Spacing Tables	New
A new variable, `checkMonotonicViolations`, is now provided in the `.cdsenv` file to determine how spacing tables are checked for monotonic violations.	

Constraints

`implantGroup`	New
(ICADVM18.1 Only – 95511 and 95800) Specifies an implant group based on the layers specified.	
`mergedViaCornerToCornerSpacing`	New
(ICADVM18.1 Only–95511) Defines the corner-to-corner spacing between two merged vias with negative parallel run length.	
`voltageLayerMarkerMapping`	New
(ICADVM18.1 Only–95512) Specifies the LPP to be used for marking nets that transition in phase. The constraint applies on the layer on which the voltage is measured.	
`voltageLayerPairMarkerMapping`	New
(ICADVM18.1 Only–95512) Specifies the LPP to be used for marking nets that transition in phase. The constraint applies on the two layers between which the voltage is measured.	

SKILL Functions

`techCreateSnapPatternDef`	Enhanced
(ICADVM18.1 Only–95512) A new set of arguments has been added: `multiTrackCenter` and `singleTrackCenter`.	

vrfLowerPriority	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) Lowers the priority of a specified dynamic shape so that it is one unit less than the lowest priority currently set for dynamic shapes on the layer.	
vrfRaisePriority	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) Raises the priority of a specified dynamic shape so that it is one unit higher than the maximum priority currently set for dynamic shapes on the layer.	

Technology File Sections

snapPatternDefs	Enhanced
(ICADVM18.1 Only–95512) A new argument has been added: `snappingMode`.	

ICADVM18.1 Base Release

What's New and Enhanced

Features

The Techfile IDE	New
A new IDE is now included to make it easier to understand, write, and modify ASCII technology files.	
New Purpose and Layer Names	New
New reserved layer names have been added: `assemblyBoundary`, `beginGenericLayer`, `drill`, `endGenericLayer`, `internalGenericLayer`, `solderMaskBottom`, `solderMaskTop`, `wirebond`, and `wirebondFingerGuide`. A new reserved purpose name has been added: `dynamic`. The tables listing the reserved values for purpose and layers, respectively, in the Technology File Attributes section of *Virtuoso Design Environment SKILL Reference* have been updated.	
New Function Names	New
New function names have been added: `drill`, `diestack`, and `wirebond`. These have been included in the list of function names. See functions in *Virtuoso Technology Data ASCII Files Reference*.	

Constraints

Packaging Constraints	New

(ICADVM18.1 Only – Virtuoso MultiTech Framework) The following constraints pertain to various aspects of packaging: ■ pkgMinAperture ■ pkgMinApertureDefault ■ pkgMinArea ■ pkgMinAreaDefault ■ pkgMaxBondwireLength ■ pkgMinBondwireLength ■ pkgMinBondwireSpacing ■ pkgMinBondwireWidth ■ pkgMinHoleSpacing ■ pkgMaxLineWidth ■ pkgMinLineWidth ■ pkgMinSpacing	
minCornerVoltageSpacing	Enhanced
(ICADVM18.1 Only–95511) A new parameter has been added: endOfLineWidth	
trimShape	Enhanced
(ICADVM18.1 Only–95511) A new parameter has been added: keepoutRegion	

SKILL Functions

Fabric Type Functions	New
The following functions help you set and get the fabric type of a technology database: ■ techGetFabricType ■ techSetFabricType	
Analysis Attributes Functions	New
The following functions help you work with analysis attributes in the case of layers that support such attributes: ■ techSupportsLayerAnalysisAttributes ■ techGetLayerAnalysisAttribute ■ techHasLayerAnalysisAttribute ■ techSetLayerAnalysisAttribute	
SiP Object Functions	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) The following functions help you set and get values for attributes of SiP objects: ■ vrfSipSet ■ vrfSipGet	
Wire Profile Functions	New

(ICADVM18.1 Only – Virtuoso MultiTech Framework) The following functions help you create, find, and delete wire profiles and wire profile groups:

- techCreateWireProfile
- techCreateWireProfileGroup
- techDeleteWireProfile
- techDeleteWireProfileGroup
- techFindWireProfile
- techFindWireProfileGroup

(ICADVM18.1 Only – Virtuoso MultiTech Framework) The following functions help you import and export wire profile data:

- techImportWireProfileSet
- techExportWireProfileSet

Technology File Sections

fabricType	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) Specifies the fabric type of a technology database.	
analysisAttributes	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) Specifies the names and related attributes of the materials used to manufacture a layer.	
Technology File Packaging Definitions	New
(ICADVM18.1 Only – Virtuoso MultiTech Framework) A new section, packaging, has been added. It includes the new wireProfiles and wireProfileGroups classes.	

Removed Features

Removed Constructs
The following constructs have been removed per warnings issued in previous releases. Remove these from your technology files before you load the files. ■ `compactorRules` ■ `dlrRules` ■ `lasRules` ■ `streamLayers` ■ `cdsMos` ■ `cdsRes`
Deprecated Constructs
The following constructs have been deprecated and will be removed in an upcoming release. Cadence recommends that you remove these from your technology files. ■ `techPermission` ■ `routeSpecs` ■ `symEnhancementDevice` ■ `symDepletionDevice` ■ `symPinDevice` ■ `symRectPinDevice` ■ `viaLayers` ■ `lxRules` (`lxExractLayers`, `lxNoOverlapLayers`, `lxMPPTemplates`) ■ `prRules` (`prRoutingLayers`, `prViaTypes`, `prRoutingPitch`, `prRoutingOffset`, `prOverlayLayer`, `prStackVias`, `prMasStackVias`, `prMastersliceLayers`, `prViaRules`, `prGenViaRules`, `prNonDefaultRules`) ■ `layerRoutingGrid` ■ `electricalRules` (`characterizationRules`, `orderedCharacterizationRules`, `tableCharacterizationRules`) ■ `symContactDevice`

Technology Database Checker

No new features or enhancements have been introduced in Technology Database Checker in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Technology Database Checker User Guide*

Open Simulation System

This chapter provides a high-level overview of the new and enhanced features in Open Simulation System in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

■ *Open Simulation System Reference*

ICADVM18.1 ISR1

What’s New and Enhanced

SKILL Functions

hnlRegPreNetlistTrigger	New
Registers a trigger, which is a user-defined SKILL procedure that is called before netlist generation.	
hnlDeRegPreNetlistTrigger	New
Deregisters a trigger, which is a user-defined SKILL procedure that has been registered using `hnlRegPreNetlistTrigger`.	
hnlPreNetlistTriggerList	New
Returns the list of functions registered through `hnlRegPreNetlistTrigger`.	

Component Description Format

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

■ *Component Description Format User Guide*

ICADVM18.1 ISR9

What’s New and Enhanced

Environment Variable

defaultCDFType	New
Specifies the default value of the radio options in the CDF form.	

Virtuoso Text Editor

This chapter provides a high-level overview of the new features in the ICADVM20.1 and ICADVM18.1 releases.

- ICADVM18.1 ISR13
- ICADVM18.1 ISR12
- ICADVM18.1 ISR10

Related Documentation

- *Virtuoso Text Editor User Guide*

ICADVM18.1 ISR13

What’s New and Enhanced

Feature

Virtuoso HDL Package Setup Form	New
You can use the new HDL Package Setup form to: ■ Apply the unified HDL package text file setup for all Virtuoso tools. ■ Define the compilation order for the HDL package text files. ■ Search the HDL package text views in all `cds.lib` libraries. ■ Check HDL package setup and display dependencies. ■ Export HDL package setup as an xrunArgs file for reuse. The form ensures that handling HDL package files becomes more intuitive, requires less maintenance, and improves the task flow for SystemVerilog, SystemVerilog-AMS, VHDL, and VHDL-AMS package files.	

ICADVM18.1 ISR12

What’s New and Enhanced

Feature

showLineNumbersInSideBar	Enhanced
The default value of this environment variable has been changed to `t` to improve the WebEx productivity.	

ICADVM18.1 ISR10

What's New and Enhanced

Feature

Specifying the License Checkout Order for Check and Save in Text Editor	New
Use the new AMSEnvLicenseCheckoutOrder license to specify whether to use the AMS_Environment or an ADE license to run the *Check and Save* feature of Text Editor for verilogams and SystemVerilog views. The default license check out order is AMS, ADE, but you can customize this order or use only an ADE license for this command. Default value: AMS, ADE **Note:** AMSEnvLicenseCheckoutOrder replaces skipAMSEnvironmentLicCheck, which will be removed from a future release.	

Virtuoso Import Tools

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

■ *Virtuoso Import Tools User Guide*

ICADVM18.1 ISR9

What’s New and Enhanced

Faster Verilog Import	New
Verilog netlists for very large designs can be imported more quickly by using the new `cdsTextTo5x` command. For example, Cadence benchmarks indicate that a large design netlist featuring up to 20M pins and 10M instances can be imported within an hour, representing a 10x performance improvement. Additionally, you can use the command-line option `-BLACKBOX` to import a cell as a blackbox when you do not require its hierarchy to be imported.	

Part 3: Schematic Design

Virtuoso Schematic Editor

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 and releases.

Related Documentation

- *Virtuoso Schematic Editor User Guide*
- *Virtuoso Schematic Editor SKILL Reference*

ICADVM20.1 Base Release

What's New and Enhanced

Features

Enhanced Connectivity Extractor	Enhanced
The schematic extractor has been updated to facilitate improvements in performance, memory usage, and connectivity efficiency. Previously, each time you ran a connectivity check, the schematic extractor would destroy and rebuild all the connectivity in a design. Now, the connectivity is updated such that only the necessary changes are made to the database connectivity. As a result, there are now two connectivity modes available, batch and incremental. The default is incremental mode, which means that the connectivity of your design is updated automatically as you create and edit the design (see below for more details). In batch mode, the connectivity of a cellview is created in the usual way by you explicitly requesting an update by running the *File – Check and Save*, *Check – Current Cellview*, or *Check – Hierarchy* commands. If you introduce issues that prevent the connectivity from being made when updating your design, error and warning messages and markers are generated and placed in the canvas, the Annotation Browser, and the CIW to make you aware of the problem. For information on how to resolve these issues, see Troubleshooting Schematic Connectivity.	
Incremental Connectivity Extraction	New
You can now run connectivity extraction in incremental mode, which means that the connectivity in your design is updated automatically as you create and edit the design. You control incremental connectivity extraction by using the *Incrementally update connectivity while editing* option on the Editor Options form. The default behavior is controlled by the `incrementalConn` environment variable. Additionally, the `incrementalConnTimeout` environment variable lets you control how long an operation can take before incremental extraction is automatically paused.	

Creating Split Symbols	New

You can now create splits of a large symbol in Virtuoso Schematic Editor. Splitting a large symbol master allows the schematic to be more readable and have a legible presentation of the pin symbols.

Note: This feature is most useful for PCB and package schematics.

The following SKILL APIs are also available to create split symbols:

- schCreateSplitPrimarySymbol
- schGetSplitInstances
- schGetSplitInstTerms
- schGetSplitPrimaryInst
- schGetSplitPrimaryInstTerm
- schIsSplitInst
- schIsSplitPrimaryInst
- schIsSplitPrimarySymbol
- schIsSplitSymbol
- schIsUsingSplitFeature
- schSetSplitPrimaryInst
- schSetSplitSymbol
- schSplitNumber

Queries in Navigator Assistant	New

You can use queries in the Navigator assistant in many ways:

- To group design objects, such as nets to be routed together, into categories.
- To check design work stages, such as unplaced instances.
- To create a group of objects to be included in an operation, such as a set of clock nets when defining net priorities.
- To maintain ‘To Do’ lists of objects that require an operation to be carried out, such as clock nets without net priority when defining net priorities.

SKILL Functions

schGetShapeStyle	New
(ICADVM20.1 Photonics Only) Returns a disembodied property list (DPL) containing the style attributes of a given shape.	
schSetShapeStyle	New
(ICADVM20.1 Photonics Only) Specifies style attributes of a shape to override the defaults provided by the Display Resource file.	

ICADVM18.1 ISR13

What’s New and Enhanced

Features

Distinguish Optical Pins from Electrical Pins Using Color	New
(ICADVM18.1 Photonics Only) You can now set the color of optical pins to distinguish them from electrical pins using the *Set Optical Pin Color* field on the Schematic Check Options form. The following environment variables can be used: ■ `opticalPinColor` ■ `opticalPinColoring` ■ `opticalPinFillStyle` ■ `opticalPinLineStyle`	

ICADVM18.1 ISR12

What's New and Enhanced

Environment Variables

showSummaryPage	New
Shows the *Summary* pane or the last selected set in the Navigator when a cellview is opened for the first time in a Virtuoso session.	
Environment Variables Documentation	Enhanced
Documentation is now available for the following environment variables that were publicly available in previous releases: ■ createLabelAutoJustify: Specifies whether the placement of a new wire label automatically snaps into position or requires to be manually positioned. ■ editPropChangeAllConfirm: Specifies the object type groups that require confirmation before applying the same property modification to multiple objects of the type selected on the Edit Object Properties form. ■ editPropReadOnlyNetName: Controls whether the Edit Object Properties form allows you to edit the net name of a selected wire for read-only cellviews. Editing a net name changes all wire labels associated with the net across the entire schematic; the actual net is changed only when you save a design using *Check and Save*.	

ICADVM18.1 ISR10

What’s New and Enhanced

Environment Variables

autoEquivalentConnectivityTime	New
Specifies whether cosmetic symbol changes can be automatically signed off as having no impact on connectivity. The default value is `nil`. For more details, see Automatically Signing Off Cosmetic Changes to Symbols.	

ICADVM18.1 ISR8

What's New and Enhanced

Features

Recategorization of Symbols within the Basic Library	New
The symbol categories for the `basic` library have been recategorized. The `basic` library is a default reference library provided within the Virtuoso `cds.lib` file. Each symbol within the library is now categorized and a new category, *Obsolete*, has been created to contain obsolete symbols. The *Obsolete* symbols are preserved for existing designs, but should not be used for new designs. Documentation is also now provided containing information about each symbol within the `basic` library, *Virtuoso Basic Library Reference*.	

ICADVM18.1 ISR7

What’s New and Enhanced

Features

Mixed Signal Packaged Check	Enhanced
A new packaged check, *Mixed Signal*, is available on the Schematic Rule Checks Setup form. Selecting this packaged check sets the following in a single selection: ■ enables the *Normal* packaged checks ■ enables the *Signal Type* checks ■ sets the *Verilog HDL Syntax* (*Name* tab), *Sparse Index Bus* (*Name* tab), and *Pin Direction Collision* (*Logical* tab) checks to *warning*.	

ICADVM18.1 ISR5

What's New and Enhanced

Environment Variables

Environment Variables Documentation	Enhanced

Documentation is now available for the following environment variables which were available in previous releases:

- `autoResizeColumns` controls the automatic resizing of columns in the Navigator. If set, the columns resize automatically. The default is `t`.
- `mapCvMappingSchViewName` lets you specify the mapping schematic view name when creating a mapping schematic on the Create a Mapping Schematic form. The default is `schematic` and it can be changed as required.
- `mapCvTermDirectionCheck` controls how direction mismatches of mapped nets are reported when creating a mapping schematic on the Create a Mapping Schematic form. This lets you control the emphasis of the direction mismatch required between upper and lower level terminal direction. Messages will be displayed in the CIW. The default is `error`.
- `schDynamicNetHilightIslands` lets you specify whether arcs are drawn between islands of connectivity to show the connection between wires of the same net. The default is `t`.
- `schDynamicNetHilightMemNetIslands` lets you specify whether arcs are drawn for highlighted nets and member nets. This is useful for buses and bundles where the net contains multiple nets. For bus `A<0:1>` it will also highlight `A<0>` and `A<1>`. The default is `t`.

ICADVM18.1 ISR4

What’s New and Enhanced

Feature

Include Design Intent	New
The Copy form now has an option to *Include Design Intent* which lets you specify whether the design intent associated with an object is also copied.	

Environment Variable

copyDesignIntent	New
Specifies whether the design intent associated with an object is included when using the *Copy* command. The default is `nil`.	

ICADVM18.1 ISR3

What's New and Enhanced

Environment Variables

Default settings for pin labels	New
These pin label environment variables define the default settings for pin labels. Each has an equivalent variable that controls whether the default settings for pin labels can be changed: ■ pinLabelDrafting specifies that labels are never drawn upside-down or mirrored, even when rotated. The default is t. To change this setting the pinLabelSetDrafting environment variable must first be set to t (default). ■ pinLabelFontHeight defines the font height of pin labels. The default is 0.0625. To change this setting the pinLabelSetFontHeight environment variable must first be set to t (the default is nil). ■ pinLabelFontStyle specifies the font style for the pin label. The default is stick. To change this setting the pinLabelSetFontStyle environment variable must first be set to t (the default is nil). ■ pinLabelJustify specifies where the label appears in relation to its origin (placement location). The default is lowerCenter. To change this setting the pinLabelSetJustify environment variable must first be set to t (the default is nil). ■ pinLabelOverbar specifies whether overbars are displayed over pin labels. The default is nil. To change this setting the pinLabelSetOverbar environment variable must first be set to t (the default is nil).	
Environment Variables Documentation	Enhanced

Documentation is now available for the following environment variables which were publicly available in previous releases:

- `bundleDisplayMode` lets you specify whether each member of a bundle net name is stacked vertically or horizontally when a wire label containing a comma separated bundle name is added. The default is `horizontal`.
- `srcFixedGlobalsCheck` specifies the severity of the schematic rules checker (SRC) Plain Globals check. Controls the check for any globals that are not the default for a net expression within a cellview. The default setting is `ignored`.
- `srcInhImplicitTerm` specifies the severity of the SRC Implicit Inherited Terminals check. Controls the check for inherited signals' connections. The default setting is `ignored`.
- `srcInhInstPinPinConn` specifies the severity of the SRC Inherited Instance Pin/Pin Connections check. Controls the check for inherited instance terminals with connections to nets with terminals. The default for this option is `ignored`.
- `srcMatchedParameters` specifies the severity of the SRC for the Mismatched Parameters check. Controls the check of mismatched parameter values for devices that are members of the Matched Parameters constraint. The default setting is `warning`.

SKILL Functions

`schClearConn`	New
Removes the schematic connectivity from a specified cellview to restart extraction using the `schExtractConn` or `schCheck` functions.	

Documentation Updates

Updating Pins From Another View of the Same Cell	New
Documentation is now available covering the Update Pins form. This form lets you update a schematic or symbol's pins to match those in another view of the same cell. It is accessed using the *Edit – Update Pins From View* command.	

ICADVM18.1 ISR2

What’s New and Enhanced

SKILL functions

tsgConnectorMasters	New
A pin master can now be associated with a list of signal types. If you have the Virtuoso_Photonics_Option license, you can also specify a mixture of optical and electrical pins during symbol generation.	

Documentation Updates

Support for Optical Signal Types	New
(ICADVM18.1 Photonics Only) Documentation is now available covering the optical signal types functionality on the Schematic Check Options and the Schematic Rules Checks Setup forms.	

ICADVM18.1 ISR1

What’s New and Enhanced

SKILL functions

schAttachLibToPackageTech	New
Attaches a package technology to the given library.	

ICADVM18.1 Base Release

What's New and Enhanced

Feature

Improved functionality on forms accessed via the Create menu	New
The *Create* menu forms (*Create - Net Expression*, *Create - Block*, *Create - Note - Text* and *Create - Note - Shape*) have been renamed and the layout of each has been improved to provide better usability: ■ Create Net Expression Form, formerly the Add Net Expression form ■ Create Block Form, formerly the Add Block Form ■ Create Note Text Form, formerly the Add Note Text Form ■ Create Note Shape Form, formerly the Add Note Shape Form	

Environment Variables

`srcPinNetCollision`	New
Controls the check for pins connected to nets with wire names that do not match the pin. The default has been changed from `ignored` to `warning`.	
`descendTarget`	Enhanced
The documentation for this environment variable has been enhanced to describe more about how the `use viewNameList` option is used.	

SKILL functions

`schTraceNet`	New
Traces a schematic hierarchical net through the design hierarchy and calls a user-defined callback at each level of the hierarchy where the net is present.	
`schGetCellViewListInSearchScope`	Enhanced

This function returns a list of cellviews in the search scope specified. It is necessary for the function to open the cellview which increments the reference count. To ensure the cellviews returned are correctly released, the call `dbClose()` is to be returned on each cellview after use.

Documentation Updates

Virtuoso Schematic Editor User Guide now covers Schematics L and XL	New
The user guides for Virtuoso Schematic Editor L and Virtuoso Schematic Editor XL have been brought together into a single user guide, Virtuoso Schematic Editor.	

Virtuoso Design Intent

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Unified Custom Constraints User Guide*
- *Virtuoso Unified Custom Constraints SKILL Reference*
- *Virtuoso Design Intent User Guide*
- *Virtuoso Layout Suite XL: Basic Editing User Guide*
- *Virtuoso Layout Suite XL: Connectivity Driven Editing*
- *Virtuoso Schematic Editor User Guide*

ICADVM18.1 ISR12

What’s New and Enhanced

Features

Defining a Reference Pin as Single Bit or Multiple Bit	Enhanced
When creating a HighCurrent design intent for a multiple-bit pin connected to multiple-bit instance terminals, you can use the new Select Reference Pin(s) form to specify whether the reference pin for the design intent is to be a bundled multiple-bit pin or a single-bit pin.	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

Defining Scopes for Profile Properties	New
You can now control the visibility of the property settings for the current profile using scopes. Defining scopes for a profile allows different designers working together on a design to filter the profile properties so that only the properties in which they are interested are visible on the design intent forms. When more than one scope is defined, a combo box appears alongside the *Profile* field on the Create Design Intent and Edit Design Intent forms from which the scope required for the profile can be selected. You can use the following environment variables to customize the scopes for a profile: ■ `diPropDefaultScopes` ■ `diPropScopesFilter`	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

Defining Currents to Equal Zero	New
For HighCurrent design intents that use the profile *Current*, the *Comply with Kirchhoff's Current Law* check box has been added in the Inst Terms dialog box. Enabling this check box lets you specify whether the total current entering or exiting a node must equal zero.	

SKILL Functions

ciDiMinMaxVPropertyCallback	New
Runs a callback to check that customized property definitions using minimum and maximum voltage property values are set correctly so that *Min Voltage* is less than or equal to *Max Voltage*. When these properties are set incorrectly in the Create Design Intent form or Edit Design Intent form, the properties are highlighted and the form is prevented from closing until the values are corrected.	
ciDiPostTransferHighCurrent	New
Runs a callback to ensure that when a customized HighCurrent design intent is transferred from Schematics XL to Layout XL, the current is split evenly between the mfactored members in Layout XL. This function is specified using the `postTransferCallback` for the HighCurrent design intent properties.	
ciDiPostTransferMinMaxVoltage	New
Runs a callback to propagate the design intent properties Min Voltage, Max Voltage, Signal Type, Power Sensitivity, and Ground Sensitivity onto the associated design intent objects after transfer from schematic to layout.	
ciDiReplaceOrAddPropertyGroupDef	New
Adds or replaces customized design intent property group definitions.	
ciSetDIPropertyGroupDefs	New
Sets the design intent property group definitions to the specified definitions. All existing property group definitions are deleted.	
ciUpdateObjPropsFromDI	New

Runs a callback to propagate design intent properties onto the associated design intent objects. The function can be called when the Create Design Intent form or Edit Design Intent form are submitted.

ICADVM18.1 ISR9

What’s New and Enhanced

Feature

Design Intents for Global Nets	New
You can now create design intents on global nets using the Create Design Intent form. For more details see Creating a Design Intent.	

ICADVM18.1 ISR7

What’s New and Enhanced

Feature

Sync All Selected Design Intent	New
You can now sync the design intents between the schematic and layout cellviews for the selected objects only. A new command *Sync All Selected Design Intent* is available on the *Design Intent* context-sensitive menu. This command is available when the objects selected in the canvas or Navigator have multiple design intents associated that require to be synced. For more details, see the context-sensitive menu option on the Canvas and Navigator.	

ICADVM18.1 ISR6

What’s New and Enhanced

Feature

Sync button added to the Design Intent toolbar	Enhanced
The *Sync* button is enabled on the Design Intent toolbar if there are design intent changes to be synced. Clicking this button synchronizes all the design intent changes in the schematic and layout views. For more details, see Syncing Design Intent.	

ICADVM18.1 ISR5

What’s New and Enhanced

Features

Syncing a Single Design Intent	Enhanced
You now have the flexibility to sync an individual design intent where previously you had to sync all the design intent within a design at the same time. When a large design update impacts multiple design intents, you can now review and sync each design intent, one at a time. For more details, see Syncing Design Intent.	
New Design Intent Category now available in Navigator	New
A new category called *Design Intent* is now available in the *Summary* pane of the Navigator in Schematics XL and Layout XL. This new category lists each design intent within the design and when expanded lists each object associated with that design intent. Selecting a design intent halos its associated objects on the canvas across both applications.	

ICADVM18.1 ISR4

What’s New and Enhanced

Features

Multiple Cell Design Intents	New
You are now able to create multiple cell design intents within a cell using the Create Design Intent form. This lets you specify design instructions for an entire cellview rather than just for individual objects. For more details, see Creating a Cell Design Intent for a Cellview.	
Design Intent Reports	Enhanced
The high-level design intent reports generated from Schematics XL or Layout XL have been improved. When the *Report* option on the *Design Intent* toolbar is selected, you now have the option to include hierarchical information via the Configuration: Design Intent Report dialog box.	

ICADVM18.1 ISR2

What's New and Enhanced

SKILL Functions

New SKILL functions to create and modify design intents	New
The following new functions have been added to allow design intents to be created and modified using SKILL. For more details, see the individual documentation for each function in the Virtuoso Unified Custom Constraints SKILL reference.	
`ciDevGroupBoxIterator`	New
Finds all device groups within a cellview that are enclosed by a text layer box.	
`ciTemplateChangeDIProfile`	New
Replaces the property profile currently selected for a design intent template.	
`ciTemplateCreateDI`	New
Creates a design intent template for the specified template type.	
`ciTemplateDIProfileName`	New
Returns the name of the profile associated with the specified design intent template.	
`ciTemplateDIPropDef`	New
For the specified design intent template, returns the definition of the named property in the form of a DPL.	
`ciTemplateDIPropGroupDef`	New
For the specified design intent template, returns the definition of all the properties in the form of a DPL.	
`ciTemplateDIPropValue`	New
For the specified design intent template, returns the value of a named property.	
`ciTemplateIsKindOfDI`	New
Confirms if the specified constraint template is a design intent template.	
`ciTemplateListDIProps`	New
For the specified design intent template, returns a list of the property names, types, and values.	

ciTemplateUpdateDIProps	New
For the specified design intent template, updates the value of the named property.	

ICADVM18.1 Base Release

What’s New and Enhanced

Virtuoso Design Intent	New
Virtuoso Design Intent complements the Virtuoso Schematic Editor XL and Virtuoso Layout Suite XL applications by providing a method for the schematic designer to capture and communicate their design goals on existing objects within a design and to provide the layout designer the freedom to decide how those goals are implemented. The design goals can be defined and discussed, implementation restrictions resolved, and decisions agreed and recorded to avoid duplication of effort during design reuse. For details on the license required for Virtuoso Design Intent, see License Consumption for Products in *Virtuoso Software Licensing and Configuration User Guide*.	

Virtuoso Unified Custom Constraints

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 base release and subsequent ISR releases.

Related Documentation

- *Virtuoso Unified Custom Constraints Configuration Guide*
- *Virtuoso Unified Custom Constraints Getting Started Guide*
- *Virtuoso Unified Custom Constraints User Guide*
- *Virtuoso Unified Custom Constraints SKILL Reference*

ICADVM20.1 Base Release

What’s New and Enhanced

SKILL Functions

ciAPRCascodeIterator	New
(ICADVM20.1 EXL Only) Iterates over the specified cellview and returns objects that match a given finder expression.	
ciDiReportGenReport	New
Generates a report summarizing the design intent in the current design based on the specified criteria and optionally opens the report in the browser.	

ICADVM18.1 ISR12

What’s New and Enhanced

SKILL Functions

ciCacheTransfer	New
Transfers constraints from the source cache to the specified target cache. The transfer can be restricted to constraints of a specified type. The function cannot be used to transfer constraints between two schematics or two layouts.	

Environment Variables

highlightCPResults	New
Controls whether finder results are highlighted in the Circuit Prospector.	

ICADVM18.1 ISR11

What's New and Enhanced

SKILL Functions

ciAPRXYInstSymmetricIterator	New
(ICADVM18.1 EXL Only) Evaluates the `finderExpr` with the current symmetric pair of objects with common source or drain terminal names that are assigned to `L` and `R` local variables. This iterator is customized for the Auto-Device Placement and Routing flow and is used by the *Circuit Prospector* Analog Design Structures finders to iterate over all symmetric design instance pairs with common source or drain terminal names, collecting them into symmetric pairs if the result of evaluating the passed expression (`finderExpr`) is not `nil`.	
ciConUpdateMemberParams	New
Updates the specified constraint member parameters with the given values without impacting other member parameters.	

ICADVM18.1 ISR6

What's New and Enhanced

Features

Creating New Versions of a Custom Constraint Type	New
You can now define multiple versions of a custom constraint type in `config.xml` by assigning a unique constraint type name to different constraint types that share the same base type name. The following SKILL functions have been added to enable this: ■ `ciTypeDefBaseType` Returns the base type of the passed custom constraint type. ■ `ciTypeDefVersion` Returns the version number of the passed constraint type. The default version number is `1`. Custom constraint types defined through `config.xml` can specify the version number of the constraint through a *Version* tag. See Customizing Constraint Types Using a Configuration File in the *Virtuoso Unified Custom Constraints Configuration Guide* for more information.	

ICADVM18.1 ISR5

What's New and Enhanced

SKILL Functions

ciCommonGateIterator	New
Returns a list of corresponding devices that match common gate structures in the cellview. If the finder match net expression evaluates to `nil`, the structures are ignored. This SKILL function was available in ISR4.	
ciGetCPSelectedResults	New
Returns a list of finder results selected in Circuit Prospector. Each item in the list is a finder result which contains the list of devices and the finder name.	
ciGetMappedDeviceNames	New
Returns all default device names (fet, nfet, pfet, and BJT) that have been registered using ciRegisterDevice.	
ciInstsNetsPinsFromSelSet	New
Returns a list of instances, nets, pins, or instTerms (`InstsNetsPins`) for the current selection.	

ICADVM18.1 ISR4

What's New and Enhanced

SKILL Function

New Circuit Prospector Iterators	New
The following iterators are now available for the Circuit Prospector: `ciActiveSameCellAndSizeIterator` returns a list of devices where each sub-list corresponds to two or more active devices that have the same cell and size in the cellview. `ciCascodeSeriesCurrentMirrorIterator` iterates over all cascode series current mirror structures in the cellview and returns a list of corresponding devices. `ciCommonGateAndSourceIterator` iterates over all common gate and source structures in the cellview and returns a list of corresponding devices. `ciCommonSourceIterator` iterates over all common source structures in the cellview and returns a list of corresponding devices. `ciHierarchicalSeriesIterator` iterates over all the series structures in the cellview and returns a list of corresponding devices. If the finder match net expression evaluates to `nil` for these iterators, the structures are ignored.	

ICADVM18.1 ISR3

What’s New and Enhanced

SKILL Function

ciRegisterDevice	Enhanced
You can now use an optional match expression string for the argument *l_deviceNameMapList* when registering a group of devices to be used by ciIsDevice.	

ICADVM18.1 ISR2

What’s New and Enhanced

Feature

Allowed Width Ranges	New
Multiple width parameter values from a defined range of valid widths associated to a net LPP can now be selected for the net type constraints: *Bus*, *Diff Pair*, *Matched Length*, *Net Class*, *Process Rule Overrides*, and *Symmetry*. **Note:** This feature requires the constraint `allowedWidthRanges` to have been defined for the layers. The width parameter values are selected and deselected on the new Allowed Width Ranges form which is accessed from the *Constraint Parameter Editor*. For more details, refer to Widths Minimum Parameter in the *Virtuoso Unified Custom Constraints User guide*.	

ICADVM18.1 ISR1

What’s New and Enhanced

Feature

New Constraint Type	New
The new constraint type, Voltage Synced Net, is used to check the spacing between two synced nets. If the Voltage Synced Net constraint fails, a spacing violation marker is generated and a voltage synced layer is created in the design	

ICADVM18.1 Base Release

What's New and Enhanced

Feature

Co-managed Schematic and Constraint Views	New
You can now specify that a number of Schematics XL commands operate on both schematic and constraint views simultaneously, saving you from having to manage the views separately. The commands are: ■ *Save, Save a Copy, Discard Edits, Check and Save,* and *Change Edit Mode*, *all in the File* menu. ■ *Check - Current Cellview* This feature is enabled using the `comanageConstraintView` environment variable. The default is `t`.	

SKILL Functions

`cstFindCutClassConstraintByName`	New
Returns the database ID of the cut class constraint with a given cut class name, layer, and constraint group.	
`cstFindCutClassConstraintBySize`	New
Searches the specified layer in the specified constraint group and returns the first cut class constraint with the specified width and length. If *`g_checkOrientation`* is set to `t`, the orientation of the cut class will also be included as part of the search.	

NC-Verilog Environment

No new features or enhancements have been introduced in NC-Verilog Environment in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso NC-Verilog Environment User Guide*

Virtuoso SystemVerilog Netlister

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso SystemVerilog Netlister User Guide*

ICADVM20.1 Base Release

What’s New and Enhanced

Features

Packed and Unpacked Array Recognition	New
A new check box, *Enable Array Type propagation*, is added to the Netlister tab of the SystemVerilog Netlister Options form. Using this option, you can enable the recognition of packed and unpacked arrays and propagate the unpacked property to the top level.	
Stop View Support	New
A new flag, `hnlUserStopCVList`, can now be added in the `.simrc` file to provide a list of user specified cellviews, which are treated as stop views while netlisting the design.	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

Importing Package Files Automatically	New
A new check box, *Enable auto package importing*, is added to the Netlister tab of the SystemVerilog Netlister Options form. Using this option, you can automatically search and find the pre-compiled SystemVerilog package files on which the `systemverilog` modules in the design depend.	
Specifying Additional Arguments	New
A new button, *Additional Arguments*, is added to the *Miscellaneous* tab of the SystemVerilog Netlister Options form. Clicking this button opens the Additional Arguments form, which allows you to specify additional `xrun` arguments.	

ICADVM18.1 ISR8

What’s New and Enhanced

Virtuoso SystemVerilog Netlister	New
Virtuoso® SystemVerilog Netlister is a new utility that has been introduced in this release. Use this utility to generate netlists for digital SystemVerilog designs. This utility imports configuration views of digital designs for netlist generation, directly parses and accesses SystemVerilog and Verilog text models, and creates LRM-compliant SystemVerilog configurations to generate compatible netlists.	

SystemVerilog Integration Environment

This chapter provides a high-level overview of the new features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

■ *Virtuoso Verilog Environment for SystemVerilog Integration User Guide*

ICADVM18.1 ISR11

What's New and Enhanced

Features

Specifying Port Properties for Instances	New
You can now modify the local values of the `dataType` and `portKind` properties for an instance on a terminal. In addition: ■ When `ignoreDataType` is set to `t`, the master and local values of the `datatype` and `portkind` properties are ignored. ■ When `ignoreDataType` is set to `nil`, you can specify customized values in the *Local Value* field of the `dataType` and `portKind` properties. If the local value of these properties is not specified, the master value is used.	

Virtuoso VHDL Toolbox

No new features or enhancements have been introduced in Virtuoso VHDL Toolbox in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso VHDL Toolbox User Guide*

Simulation Environment

No new features or enhancements have been introduced in Simulation Environment in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Simulation Environment Help*

Verilog In

No new features or enhancements have been introduced in Verilog In in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Verilog In for Virtuoso Design Environment User Guide*

VHDL In

No new features or enhancements have been introduced in VHDL In in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

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

Connectivity-to-Schematic

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

■ *Connectivity-to-Schematic User Guide*

ICADVM20.1 Base Release

What’s New and Enhanced

Features

Support to Connect Floating Ports and Nets to A Cellview	New
A new group box, *Cellview to Connect Floating Ports/Nets* is now available in the *Schematic Generation Options* tab of the Connectivity-to-Schematic window. Using the fields in this group box, you can connect the floating nets and ports in the schematic with any cellview.	

Virtuoso Power Manager

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Power Manager User Guide*

ICADVM20.1 Base Release

What’s New

Feature

Virtuoso Power Manager	Enhanced
Virtuoso Power Manager provides an interface to specify, import, and export low power intent for designs. It provides the capability to perform static low power verification on designs by using Conformal Low Power (CLP) integration with Virtuoso and supports analog, digital, and mixed-signal implementations. Power Manager, by using the import flow, can annotate and stitch the top-down power connectivity for a hierarchical design by using the connectivity model for inherited connections. Using the export flow, you can extract the power intent of a fully connected design schematic. You can also generate a Liberty-based macro cell for custom blocks using Power Manager for full-chip power intent verification. When using special power-saving techniques for efficient power consumption, you often implement stringent power schemes throughout the design that add to the overall complexity while verifying the design. In addition, there are design scenarios that should be verified for the correct functioning of a circuit, for example, the correct biasing of MOS devices. To reduce the complexity, you should be able to perform a few basic-to advanced-level circuit design checks on schematic designs. The In-Design Checks feature in Power Manager provides such static checks that do not require circuit simulation.	

ICADVM18.1 ISR6

What’s New

Feature

New Power Manager Setup Form	New
Now, you can register the power-related setup information across the design hierarchy, interactively by using the Power Manager Setup form. Earlier, you could add the registration information only through a Setup file template.	

ICADVM18.1 ISR3

What’s New

Feature

Virtuoso Power Manager	New
Virtuoso Power Manager is a product for automated extraction of power intent from custom designs in Virtuoso Schematic Editor. It simplifies the design process for defining and checking the power intent based on the IEEE 1801 format. This format is an industry standard format that is used to express the power intent for both IP and SoC designs. Power Manager provides the functionality to import and export the low power intent for designs expressed in IEEE 1801 or Liberty format. It also provides the capability to perform static low power verification of designs by using Conformal Low Power (CLP) integration with Virtuoso. *Important* Virtuoso Power Manager is available for use in your production setup starting from ICADVM18.1 ISR3, but only on an on-demand basis. For more information, contact your Cadence representative, who will be able to guide you through the required steps to get started.	

Part 4: Analog Design

Virtuoso ADE Explorer and Virtuoso ADE Assembler

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso ADE Explorer User Guide*
- *Virtuoso ADE Explorer and Assembler SKILL Reference*
- *Virtuoso ADE Assembler User Guide*

- *Virtuoso Variation Option User Guide*
- *Virtuoso ADE Explorer and Assembler SKILL Reference*

ICADVM20.1 Base Release

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Distributing Simulation Jobs using LSCS	Enhanced
LSCS is now the default job control mode in the *Job Policy Setup* form for the newly created maestro views. LSCS implements a new architecture to run simulations with better resource utilization. It also enables future scalability requirements, mainly cloud simulations. The job control mode for the existing views remains unchanged.	
Filtering the Points Table in Run Preview	Enhanced
You can view specific data points in the *Run Preview* tab by using the filters available in each column header.	
Setting Up High-Performance Simulation Options	Enhanced
In the ICADVM20.1 release version, APS is the default simulation performance mode.	
Viewing the Run Statistics	New
You can now view the run statistics information in the *Results* tab by using the *Run Statistics* command in the *Configure what is shown in the table* list on the toolbar of the *Results* tab.	
Saving Snapshots Periodically	Enhanced
You can now specify a percentage value in the *Strobe Time* field of the *Netlist and Run Options* form. The time interval between saving snapshots is then calculated as the specified percentage of the value of time given in the *Stop Time* field. The value for this field can be specified in the *Choosing Analyses* form for transient analysis.	
ADE Assembler Only	
Merging Setup and Results from Multiple Histories	New
You can now merge simulation results from multiple histories. This helps you in bringing together all relevant results together without running simulations again. The feature provides the flexibility to merge selected results or add new points and simulate those while merging existing results.	

Automatic Fault Dropping Flow	New
You can now create a run plan to run multiple fault simulations in a sequence where each subsequent run in the plan drops the faults found in the previous run and uses a reduced list of faults that were not detected in the previous run.	
Separate History Management	New
Earlier, the setup details for histories were saved within the `maestro.sdb` file that contains the information about the active setup. Starting from this release, ADE Assembler saves the history setup for each history in a separate `.zip` file in the `history` subdirectory of the maestro cellview. In addition, a list of all available histories is saved in the `history.sdb` file. This makes the maestro cellview lighter and reduces their saving and loading time. By default, this feature is enabled in ICADVM20.1, but disabled in IC6.1.8. You can change the default status by using the useSeparateHistoryFileManagement environment variable.	
Monte Carlo Simulation Setup in Setup Library Assistant	Enhanced
The Setup Library Assistant now lets you add a *Monte Carlo Simulation Setup*. By dragging and dropping this setup to the Data View assistant, multiple designers can include the Monte Carlo Setup into their designs.	
Checking the Size of a History	New
You can now check the size of a history on the *History* tab of the Data View assistant.	
Importing Histories from Other Cellviews	New
You can now use the *File – Import History* command to import histories from other cellviews into the current maestro cellview. Related SKILL Function: maeImportHistory	
Copying Settings from One Run to Another in a Run Plan	New
You can now copy tests or settings from one run to another in the run plan by using the drag-and-drop operation.You can now drag and drop tests or settings from one run to another in a run plan.	

Environment Variables

ADE Explorer and ADE Assembler

maestro.results defaultRunStatisticItems	New
Specifies a space- or comma-separated list of information items to be displayed when the *Run Statistics* option is selected in the *Configure what is shown in the table* list in the *Results* tab. Default value: "memory, threads, jobId, host, nodes, startTime, endTime, elapsedTime"	
maestro.results enableRunStatisticInfoDisplay	New
Adds the *Run Statistics* option to the *Configure what is shown in the table* list on the toolbar of the *Results* tab. Default value: nil	
maestro.gui showRedundantPointsinDT	New
Displays the redundant data points in the *Detail-Transpose* results view. Default value: nil	
maestro.results waiveViolationsByCorner	New
Allows to waive corner-specific violations. To know more, see Waiving Violations. Default value: nil	
ADE Assembler Only	
maestro.faultsimulation handleErrorsAs	New
Specifies the status to be used for all simulation and fault expression evaluation errors in fault simulation results. Default value: "undetected"	
maestro.gui enableManualRefresh	New
Controls the enabled or disabled state of the *Manual Refresh* check box on the Run Options form. Default value: nil	

SKILL Functions

ADE Explorer and ADE Assembler	
asiGetEMIROptionVal	New
Returns the value of the specified EMIR option.	

asiSetEMIROptionVal	New
Sets the given value for the specified EMIR option.	
maeHasTestJobPolicy	New
Checks if the given test has a test-specific job policy setup.	
nlGetScratchInstance	New
Returns the database ID of the scratch instance to be used for Pcell evaluations.	
nlGetSwitchMaster	New
Returns the database ID of the switch master.	
nlIsPcellInstance	New
Checks if the specified instance name represents a schematic Pcell instance.	
nlIsPcellParam	New
Checks if the specified device parameter represents a Pcell parameter.	
ADE Assembler Only	
Functions for Fault Simulations	New

You can use the following new functions to work with run plans in fault simulations:

- maeGetCurrentRunPlanName
- maeGetEnabledRuns
- maeGetHistoryNameForCurrentRunInRunPlan
- maeGetNumberOfExecutedRuns
- maeGetNumberOfUndetectedFaultsFromHistory
- maeIsFinalRunCompleted
- maeIsFirstRunInRunPlan
- maeMergeFaultHistories
- maePrintFaultDroppingStatistics
- maeRunFaultSimulationWithFaultDroppingForActiveTests
- maeSaveFaultsRunCount
- maeSwitchActiveFaultGroupForCurrentRun

ICADVM18.1 ISR13

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Capacitance Results View	New
The *Results* tab now contains a new view named *Capacitance*. In this view, the results table displays the capacitance values that are saved when you enable the *allcap*, *netcap*, or *moscap* parameters in the *Save circuit information analysis* table of the Save Options form for Spectre. Related environment variable: `maestro.gui_showCapacitanceView`	
Format of Default Test Names	Enhanced
The format of default test names created in ADE Explorer and ADE Assembler has been changed to use an underscore as a separator instead of a colon. The new format for the test names is *`libName_cellName_sequenceNumber`*. If you rename a test to use the colon separator, it is automatically changed to an underscore.	
EMIR Analysis Setup Form	Enhanced
The *Spectre EMIR/Voltus-Fi XL Analysis* form has been renamed to *EMIR Analysis Setup.* For improved usability, the fields have been reorganized into different tabs. Use this form to run checks earlier in the cycle and reduce the number of iterations in the complete EM/IR flow. New environment variables have also been added to set the default values for the fields in this form. For details of the environment variables, see spectre.emirOpts.	
Viewing Results of Device Checks	Enhanced
A new column, *ViolatingRatio*, appears in the Checks/Asserts results pane when you select *Device Checks* or a dynamic mosv check from the drop-down menu. This column displays the percentage of violation detected above or below the specified minimum or maximum value in the checker definition.	
Editing Model Files for Corners	New
You can now view or edit the model files from the Corners Setup form by using the *View Model File* command in the context-sensitive menu of a corner.	

ADE Assembler Only	
Checking the Confidence Interval for Fault Simulation Results	Enhanced
You can now specify a confidence level for fault simulations that are run with samples generated using the *randomuniform* or *randomweighted* sampling methods. After the results are displayed in the *Faults* results view, you can modify the confidence level and see its impact on the confidence interval for each output.	
Multi-Test Editor	Enhanced
The Multi-Test Editor now shows the *adeInfo* element that includes the design information, simulator name, and the path to the project directory. It helps you compare the design and simulator for each test.	
Electrothermal Analysis in Legato Reliability	Enhanced
You can use the new *Save Channel Only* option on the *Electrothermal* tab of the Transient Options form to specify the layers for which you need to save the Trise data. When this option is set to `Yes`, the data is saved for the channel layer only. Otherwise, the data is saved for all layers.	
Global Variable in Run Plan	Enhanced
The parametric sets for global variables in the active setup are copied when you use the active setup to create a run plan. You can also create new parametric sets for global variables in a run plan.	

SKILL Functions

ADE Explorer and ADE Assembler	
maeSuspendSimulation	New
Suspends the simulation run for a maestro session.	
maeOpenLogViewer	New
Opens the Log Viewer window where you can view the messages loaded from a database.	
maeSetCurrentRunMode	New
Updates the current run mode in ADE Assembler.	
ADE Assembler Only	
Functions for Fault Simulations	New

You can use the following new SKILL functions to run fault simulations using SKILL:

- maeAddFaultRule
- maeAddFaultsToFaultGroup
- maeClearExistingFaultsForRevalidation
- maeCreateOrRenameFaultGroup
- maeDeleteFaultGroup
- maeDeleteFaultRule
- maeEditFaultRule
- maeGetFaultGroups
- maeGetFaultGroupToRun
- maeGetFaultRule
- maeGetFaultRules
- maeGetFaultRunModeOptions
- maeGetFaults
- maeGetFaultSamplingOptions
- maeGetGlobalFaultOptions
- maeSetFaultAnalysisType
- maeSetFaultDFARunModeOptions
- maeSetFaultGroupToRun
- maeSetFaultSamplingOptions
- maeSetFaultTFARunModeOptions
- maeSetGlobalFaultOptions

Environment Variables

ADE Explorer and ADE Assembler

adexl.monte enableMonteCarloOverStatisticalCorner	New
Enables you to run Monte Carlo analysis (mismatch variation) over parameter-based statistical corners, consisting of process variation. Default value: `nil`	

ICADVM18.1 ISR12

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Enhanced Resource Estimation for LSCS Job Control Mode	Enhanced
The following enhancements have been made for resource estimation in LSCS job control mode: ■ Earlier, resource estimation was run only for the nominal corner. Starting from this release, you can specify a resource corner in the Corners Setup form. To override a resource corner for a specific test, use the *Resource Corner* command in the context-sensitive menu of the test or add the `Resource_Corner` string in the note for that test. ■ The CPU resource estimation now reads the lower and upper bound of the estimate returned by the `+query=all` Spectre option and returns a range of cores required to run the simulation. ADE Explorer or ADE Assembler adds the lower bound of the range to the %CPU_EST template in the resource requirement string. ■ Spectre currently supports resource estimation for only hb and tran analysis. If the setup contains an enabled analysis other than these two, the feature reuses values from an available history. In case a history is not available, it uses the default memory and CPU value specified on the *Resources* tab of the Job Policy Setup form. Earlier, simulations were not started when the setup contained unsupported analyses. ■ While running resource estimation based on a history, the tool now uses the maximum memory and CPU requirements found across all points. ■ For a run mode that does not support resource estimation, ADE Assembler reuses values from an available history or the default values.	
Starting Point for Optimization and Sizing Run Modes	Enhanced
For the Local Optimization, Global Optimization, Improve Yield, Size Over Corners, and Feasibility Analysis run modes, you can now use the new `Automatic` starting state to use the schematic values of parameters and global variables for the starting point. ■ Parameters with range values are replaced with their values in the design schematic. Parameters with scalar values or a reference to other parameters remain unchanged. ■ Global variables with range values are replaced with the mid-point in the given range.	

ADE Explorer Only

Backannotating Variables and Parameters from RTT	Enhanced

After tuning the design through the *Real Time Tuning* assistant, you can now send the updated values to the Setup assistant or schematic. To do this, you can select one of the following options:

- *Explorer*: Updates the design variable and parameter values in the Setup assistant.
- *Back Annotate*: Updates the parameter values in the schematic.
- *Explorer and Back Annotate*: Updates the design variable and parameter values in the Setup assistant while also updating the parameter values in the schematic.

The related environment variable is `saveToMode`.

Process-Based Save Restart Flow	Enhanced

You can now select the *Enable Process Based Save Restart* check box in the *Netlist and Run Options form* to use the process-based save restart flow. This allows you to save the entire simulation setup, and facilitates a smoother restart operation, so that the future simulations can continue from that point. On saving the simulation setup, a new snapshot is created. To restart the simulation from that point, you can load the saved snapshot.

On selecting this check box, you see a new option, *HDL save* in the *Save snapshot* section of the form. This option represents the system function `$save` that you can use in the Verilog, Verilog-AMS and SystemVerilog modules to set up the process-based save restart operation in relation to the design or testbench events specified in the Verilog code.

ADE Assembler Only

Running Fault Simulation	Enhanced

- You can now use the new *Weight Expression* and *Weight Factor* fields on the Fault Rules form to apply weights to faults depending on their likelihood or importance. This helps you view a better coverage in the fault reports. If the faults with higher weight are reported, the report shows a better coverage.
- Fault simulations now support the sampling methods provided by Spectre. Use the Sampling Options form to choose a sampling method according to your requirements.

Running Incremental Simulations with Reference Histories	Enhanced
■ The settings on the Reference Histories form have been reorganized to separate the two use cases, one to use a reference history and the other to use a reference netlist. When you select *Use Reference Netlist*, only the *Reference Histories* group box is enabled. Other options are enabled only when you select *Use Reference History*. ■ A new option, *Carry Over Errors*, has been added for the use case that allows you to use a reference history. Use this option to specify whether to retain the result status of the points that returned errors in the previous history or to rerun simulations for those.	

SKILL Functions

ADE Explorer and ADE Assembler	
maeGetTestOutputs	New
Returns the list of outputs for the specified test in the current session or a specific maestro session.	
maeSetDesign	New
Sets a design for the given test. Use this function to change the design associated to a test	

Environment Variables

ADE Explorer and ADE Assembler	
`maestro.rtt_saveToMode`	New
Controls the default option to backannotate variables and parameters from the RTT assistant to ADE Explorer and schematic from the *Save back* drop-down menu. Default value: "`explorer`"	
`ams.envOpts_ipAddIpLabelAsPrefix`	New
Adds the label name specified in the *IP Label* field of the *AMS IP Export* form as a prefix to the Virtuoso library name. Default value: `nil`	

ICADVM18.1 ISR11

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Distributing Simulations using LSCS Job Control Mode	Enhanced
You can now use the following new environment variables to customize the message logs saved for LSCS job control mode: ■ `maestro.logging databaseDir`: Specifies the file path to be used to save the message database. ■ `maestro.logging level`: Specifies the severity level of messages to be saved. By limiting the type of messages to be saved, you can control the database size. ■ `maestro.logging maxNumberSimulatorNoteWarn`: Specifies the maximum number of information (note) and warning messages to be saved for each simulator.	
Setting Up AMS Simulation Options	Enhanced
A new check box*, Use symbols for blackboxes bound to external HDL*, has been added to the AMS Options form. Select it when you want the netlister to use a symbol view to generate an instance line for a blackbox in an extracted view. The blackbox must have the external HDL properties set in Virtuoso Hierarchy Editor. Related environment variable: `netlistBlackboxWithExtHdl`	
Setting Up High-Performance Simulation Options	Enhanced
In the *#Threads* field of the *High-Performance Simulation Options* form, you can now specify the computer farm management tool, such as sge, lsf, loadleveler, or farm, to be used to run simulation.	

Customizing Reliabiltiy Options Form	New
You can now customize the Reliability Options form by using the following new SKILL functions: ■ relxAddSetupRelxOption ■ relxCreateCustomizedTab ■ relxCustomizeDisplayOrEnableStatus ■ relxDisplayDiscField ■ relxEnableDiscField ■ relxEnableFormTab ■ relxGetCustomTabName	
ADE Assembler Only	
Running Incremental Simulations with Reference Histories	Enhanced
■ When the active setup in ADE Assembler uses reference histories for an incremental run, the color of the *Run Simulation* command on the run toolbar is changed to blue. ■ The tooltip of a history saved for a simulation run using reference histories now shows its time stamp, reference history names, and the status of the *Ignore Setup Changes* check box on the *Reference Histories* form.	

SKILL Functions

ADE Explorer and ADE Assembler	
maeExportOutputView	Enhanced
Three new arguments, `?testName`, `?filterName`, and `?clearAllFilters`, have been added to this function to control the specific information to be exported for checks/asserts and fault simulations.	
maeSensDeleteModel	New
Deletes the specified model from the setup for Sensitivity Analysis in the current or the specified maestro session.	
maeSensDeleteModelGroup	New
Deletes all the model groups specified in the *Model Groups* setting on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensDeleteParameter	New
Deletes the specified parameter from the setup for Sensitivity Analysis in the current or the specified maestro session.	
maeSensDeleteVar	New
Deletes the specified variable from the setup for Sensitivity Analysis in the current or the specified maestro session. You can also use this function to delete the value for *Temperature*.	
maeSensEnableDesignVariation	New
Selects or clears the *Enable Design and PVT Variation* check box on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensEnableStatVariation	New
Selects or clears the *Enable Variation of Statistical Parameters* check box on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensGetModel	New
Returns a list of valid values and the nominal value for the specified model from the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensGetModelGroup	New
Returns a list of values specified in the *Model Groups* setting on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensGetModels	New

Returns a list containing the names of all enabled models on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensGetParameter	New
Returns a list of valid values and the nominal value for the specified parameter from the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensGetParameters	New
Returns a list of all enabled parameters on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensGetVar	New
Returns a list of valid values and the nominal value for the specified global variable from the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensGetVars	New
Returns a list containing the names of all enabled global variables, including *Temperature*, on the Sensitivity form for the Sensitivity Analysis run mode in the current or the specified maestro session.	
maeSensSetMethod	New
Sets the value in the *Method* field on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensSetModel	New
Adds a model on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensSetModelGroup	New
Adds a model group on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session.	
maeSensSetParameter	New
Adds a parameter on the Sensitivity form for Sensitivity Analysis in the current or the specified maestro session. If a parameter already exists with the given name, its value is updated.	
maeSensSetVar	New
Adds a global variable to the run options for the Sensitivity Analysis run mode in the current or the specified maestro session. If a global variable already exists with the specified name, its value is updated. You can also use this function to change the value of *Temperature*.	

relxHideAgeCalculationApproachField	New
Hides the *Age Calculation Approach* field from the *Modeling* tab in the Reliability Options form.	
relxInitOptionsInCdsenv	New
Creates environment variables for the customized options added to the Reliability Options form.	
asiFormatSpecialParameterForRel	New
Netlists the new options added to the customized tab of the Reliability Options form.	

Environment Variables

ADE Explorer and ADE Assembler	
`checkOutLicenseDuringNetlistAndRun`	New
If set to `t`, the license `32760` Virtuoso Analog HSPICE Interface Option remains checked out only during netlisting and simulation. After these tasks are completed, the license is checked in. Default value: `nil`	
`ignoreAnalysisCheck`	New
Specifies that a check for existence of analyses is not required before running a simulation. By default, before running a simulation, the tool runs a check to ensure that at least one analysis is defined. However, if the requirement is to run a simulation without any analysis, for example, when running a digital simulation, you can set this variable to `t` to ignore this check. Additionally, if a simulation is run when the dynamic parameters from the Transient Analysis setup form are similar to the global variables, design variables, or corner variables, a message is displayed. To override this setting, set this environment variable to `t`. Default value: `t`	
`ipExportHDLFiles`	New
Selects or deselects of the *Export HDL Files* check box on the AMS IP Export form. Default value: `nil`	

allowLocalCornersInRunPlan	New
Specifies whether to allow creation of local corners in a run plan.. Default value: `t`	
author	New
Specifies the author name to be printed in the footer of the datasheet. Default value: ""	
maxNumSnapShots	New
Specifies the maximum number of snapshots to be saved during the specified period for the periodic snapshots. Default value: `4`	
saveAllSnapShots	New
Specifies whether to save all snapshots taken during the time window specified for periodic snapshots. Default value: `nil`	
snapShotBaseName	New
Specifies the base name of the snapshot. The base name must be alphanumeric with no spaces or special characters. Default value: `"mySnapShot"`	
snapShotNameTimeUnit	New
Specifies the time unit to be used in the snapshot name. Default value: `"us"`	
snapShotSaveMode	New
Specifies the mode for saving the snapshots. The two available modes are `Time points` and `Periodic snapshot`. Default value: `"Time points"`	
strobeTime	New
Specifies the time interval between saving snapshots, followed by the time unit. Default value: "".	

ICADVM18.1 ISR10

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Saving Snapshots Periodically	New
You can now save snapshots periodically after a specific time interval. Earlier, you could save snapshots only at specific time points.	
Reusing Stress Files from Other Reliability Setups	New
You can reuse the stress files from one reliability setup to another. A new form, *Configure Stress File*, has been introduced to support this enhancement. You can use this form to specify the reliability setup, history, run, corner, and design points from which you want to reuse the stress file.	
Setting up Simulator Environment Options	Enhanced
A new option, *Preserve Subckt Terminal by Names*, has been added to the *Environment Options* form. You can use this option to write the instance ports in the name format into the netlist file. By default, the instance names are written in the index format.	
Setting up Signal Outputs	Enhanced
If a signal in the output setup is not found in the schematic design hierarchy, it is considered invalid and highlighted in yellow in the Outputs Setup pane.	
Viewing Results in the Detail-Transpose View	Enhanced
If the results contain details of instance parameters, the `Detail - Transpose` result view shows the instance path in the tooltips of parameter names. This helps in uniquely identifying the parameters when multiple instances have parameters with same names.	
ADE Assembler Only	
Editing Corners in the Run Plan	Enhanced
You can now edit the values of local corners in a run plan. Any changes made in the local corner of a run are not reflected in the corners setup in the Data View assistant.	

Running Sensitivity Analysis with Hammersly Method	Enhanced
You can now use the Hammersly method for sensitivity analysis without specifying a nominal value.	
Running Fault Simulations	Enhanced
■ You can use the new *Use Netlist Syntax* check box on the Faults Rules setup form to specify the instance names in the netlist format. By default, this check box is not selected and you can specify the instance names in the schematic syntax. In this case, you can browse the design schematic to select instances, but you can specify only one instance in the *Instance* field. ■ A new *Open Fault in Design* command has been added to the context-sensitive menu of the Fault Group Preview form. You can use this to probe the fault points in the schematic view of your design. ■ `calcVal` expressions are now supported in the measurements for a fault simulation.	

SKILL Function

ADE Explorer and ADE Assembler	
maeGetStressFile	New
Returns the path of the stress file to be reused from the specified reliability setup.	

ICADVM18.1 ISR9

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Distributing Simulation Jobs	New
You can now use a new job control mode, LSCS (Large-Scale Cloud Simulations), to handle thousands of netlisting and simulation jobs in parallel. LSCS implements a new architecture to run simulations with better resource utilization. It also enables future scalability requirements, mainly cloud simulations.	
Viewing Netlist	Enhanced
The netlist files that are shown by using the *Netlist – Display* command in the context-sensitive menu of a test or the *View Netlist* command in the context-sensitive menu of a result value now contain hyperlinks to the referenced files. The file types for which you can find hyperlinks are: `scs`, `.va`, `.mod`, `.lib`, `.dspf`, `.fl`, `.vcd`, `.evcd`, `.mod`, `.def`, `.info`, `.log`, `.vec`, `.v`, `netlist`, `pspice`, `SPEF`, and `DSPF` files.	
Checks and Asserts Analysis	Enhanced
■ Two new fields, *Max* and *Min*, have been added to the results table for device checks. These fields show the maximum and minimum limits for a device check. ■ A new command, *Send to Column Filter*, has been added to the *Checks/Asserts* results view. Use this command to send a particular result value to the *Filter* cell of the same column in which the value appears to create a filtering criteria.	
Noise Summary	Enhanced
You can now use the following check boxes added to the *OPTIONS* section in the *Noise Summary* form to customize the way the noise summary data is printed: ■ *Consolidate iterated instances*: Combines the noise contributions of iterated instances to one contributor. ■ *Suffix notation*: Adds a notation for the printed information. This value overrides any global format value set with the `setup` command.	

Running Post-Layout Simulation Using DSPF Files	Enhanced
In the previous releases, if DSPF files were specified in the corners setup, only the first corner would specify whether the simulations will use DSPF files or not. Starting this release, the use of DSPF file is checked for each corner individually. For a particular subcircuit, some corners can use a DSPF cellview and others can use schematic.	

ADE Assembler Only

Running Fault Simulation	Enhanced

- A new workspace, *Fault Simulation*, has been added. It opens the Fault Setup assistant. You can set this for the maestro cellviews containing the setup for fault simulation analysis.
- The following enhancements are done in the Fault Rules setup form:
  - The fields in the *Design Hierarchy Filter* and *Fault Insertion Restrictions* group boxes have been enhanced to accept a modified set of values.
  - Added a new field, *Enable IEEE 2427 Mode*, to enable or disable the IEEE 2427 mode for Spectre info analysis.
- You can now edit fault groups to remove undesired faults.
- You can now import fault rules from other cellviews.
- You can now merge fault simulation results from multiple histories.

Working with Run Plans	Enhanced

- By default, the corners copied from the active setup to the run plan are synchronized with the active setup to reflect any change. You can now use the new *Change to Local Corner* command in the context-sensitive menu of corners in a run to make them local corners so that they are modified to reflect any change in the source corner in the *Data View* assistant.
- You can now copy run plans from one cellview to another cellview by using the *Run Plan* check box on the Load Setup State form or the Import Setup form.

Referencing Histories for Incremental Simulations	Enhanced

You can now select multiple reference histories to be used for an incremental simulation run. The following enhancements have been done to support this change:

- The Reference History form has been renamed as Reference Histories.
- New list boxes, *Histories* and *Selected Histories*, have been added to this form to allow you to select multiple histories.
- The following new check boxes have been added to this form:
  - *Run New Simulations*: By default, this check box is selected and simulations are run for the new points in the active setup. For the points common in the reference histories and the active setup, the simulations are not run and the results are copied from the reference history. Deselect this check box if you do not want to run simulations for the new point right after setting up reference histories. In this case, all new points are shown as canceled and the results for the common points are copied from the referenced history.
  - *Ignore Setup Changes*: Select this to run an incremental simulation even if there are any test-level changes in the active setup. By default, this check box is deselected.

Environment Variable

ADE Explorer and ADE Assembler	
`maestro.results sumDeviceFingerCurrents`	New
Enables or disables dumping of the sum total of terminal currents for all fingers of a multi-fingered device in the `save` statement in netlist. Default value: `t`	
`maestro.simulation behaviorOnPcellParamChangedByCallback`	New
Specifies whether to show an error or warning message when the user-specified Pcell parameter value is out of the valid range returned by the validating CDF callback. By default, the variable is set to `"none"` and incorrect values of Pcell parameters are not reported. They are replaced with valid values during netlisting. Possible values: `"error", "warning", "none"`	

maestro.distribute maxNetlistingJobsForLocal	New
Specifies the maximum number jobs allowed to be set for the netlisting services to be used when the distribution method is set to `Local` or `Remote-Host`. This variable is used only for the `LSCS` job control mode. Default value: `50`	
maestro.simulate exprEvalMaxTries	New
Specifies the maximum number of tries to launch a new expression evaluator process when the process that was already evaluating an expression stops abruptly. This variable is used only for the `LSCS` job control mode. Default value: `3`	

SKILL Functions

ADE Explorer and ADE Assembler	
maeConvertViewForReferencedAndLocalRunPlanCorners	New
Converts all local and referenced corners created in cellviews saved using IC6.1.8 ISR9 or ICADVM18.1 ISR9 to referenced corners supported in earlier versions.	
maeClearAllTestJobPolicies	New
Clears the test-level job policy setup for each test in the setup.	
maeClearTestJobPolicy	New
Clears the job setup of a given test and applies the global job setup.	
maeIsEvaluatorProcess	New
Returns `t` if the code is currently running in an expression evaluator process for ADE Assembler or ADE Explorer. **Note:** This function is used only for the LSCS mode.	
maeIsNetlistProcess	New
Returns `t` if the code is currently running in a netlister service process for ADE Assembler or ADE Explorer. **Note:** This function is used only for the LSCS mode.	
maeGetAllJobPolicies	New
Returns all job policies available in the given session.	

maeGetJobControlMode	New
Returns the job control mode currently set in the given session.	
maeSetJobControlMode	New
Sets the job control mode in the given session.	
maeStopJob	New
Stops the job for the specified job ID regardless of its state.	
maeStopAllJobs	New
Stops all the simulation or netlisting jobs you started during the current session regardless of their state.	
maeStmGenerateWaveforms	Enhanced
Three new arguments have been added to this function: ■ `?testName`: Specifies the name of the test to which the stimuli is associated. ■ `?reportFile`: Specifies the name of the report file in which the status message for each stimuli waveform is printed. ■ `?force`: Forces the waveform generation for each stimuli.	

ICADVM18.1 ISR8

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Simulation Files Setup	Enhanced
You can now specify multiple VCD and EVCD files by adding more rows in the *VCD Files* and *EVCD Files* trees on the *Vector Files* tab.	
Migrating ADE XL Cellviews to ADE Assembler	Enhanced
While using the Migrate ADE XL View form to migrate ADE XL cellviews to ADE Assembler, you can now select the *Exclude History* check box if you do not want to copy histories to the new cellview.	
ADE Assembler Only	
Fault Simulation	Enhanced
■ Faults are now injected by running Spectre info analysis based on the given fault rules. Earlier, ADE Assembler used to create schematic-based faults. ■ Three new fields have been added to the Fault Rules form: ❑ *Exclude Instances* to specify the instances to be excluded during fault injection ❑ *Exclude Subcircuits* to specify the subcircuits to be excluded during fault injection. ❑ *Extra Options* to specify additional options for the Spectre info analysis that is run to injects faults based on the specified criteria. ■ You can use any non-transient analysis for the direct fault analysis mode. ■ The format of the functional safety report now prints the fault mode ID, fault mode, and detection matrix summary at the top. The detailed fault matrix is printed after that.	

Viewing the Library, Cell, and View Information in Sensitivity Analysis	Enhanced
You can now view the complete path (*<Library>/<Cell>/<View>/<parameter_name>*) to the device parameters in the Sensitivity form and the Sensitivity Analysis window. For more information, refer to the following sections: ■ Specifying Variables, Parameters, and Model Files to be Varied for Sensitivity Analysis ■ Viewing the library, cell, and view in Sensitivity Analysis Form	

Environment Variable

ADE Explorer and ADE Assembler	
`asimenv.netlist displayPointNetlistTimeStamp`	New
Displays the time stamp in the header of the point netlist generated. Default value: `t`	

SKILL Functions

ADE Explorer and ADE Assembler	
maeCloseViolationDb	New
Closes an open checks and asserts database using the database ID returned by maeOpenViolationDB.	
maeOpenViolationDb	New
Opens a connection to the checks and asserts database for the given maestro cellview and returns a unique ID for the connection.	
maeWaiveViolation	New
Adds to the waivers SQL database a rule to waive a check and an assert for the given object.	
maeUnWaiveViolation	New
Removes from the waivers SQL database a waive rule matching the given criteria.	

ICADVM18.1 ISR7

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Printing Noise Summary Information	Enhanced
The following enhancements have been made to the Noise Summary form: ■ You can use the new *hierarchy level* drop-down list added to the *FILTER* section to filter the noise contribution results by hierarchy levels. Noise contribution results up to the specified hierarchy levels are displayed in the *Results Display Window*. ■ You can also use the wildcard character `*` in the *include instances* and *exclude instances* fields to include or exclude all the devices under the specified instances.	
Plotting Terminal Voltage using Direct Plot Main form	Enhanced
You can plot terminal voltage for the tran, ac, dc, pss, hb, pac, hbac, qpss, envlp, and qpac analyses using Direct Plot Main form. To do this choose `Terminal` from the *Select* drop-down list in the Direct Plot Main form.	

Environment Variables

ADE Explorer and ADE Assembler	
`maestro.gui_textColorForSpecNearInResults`	New
Specifies a text color, color name, or a color value in hexadecimal format, to be used for the *Pass/Fail*, *Min*, and *Max* columns for the result values that are near the specification. Default value: `"#D2691E"`	
`maestro.gui_textColorForSpecPassInResults`	New
Specifies a text color, color name, or a color value in hexadecimal format, to be used for the *Pass/Fail*, *Min*, and *Max* columns for the result values that meet the specification. Default value: `"#FF0000"`	

maestro.gui textColorForSpecFailInResults	New
Specifies a text color, color name, or a color value in hexadecimal format, to be used for the *Pass/Fail*, *Min*, and *Max* columns for the result values that fail to meet the specification. Default value: `"#008000"`	
maestro.gui nameDisplayWidthInDataView	Enhanced
Earlier, you could use this variable to set the default width of columns in the Setup assistant in ADE Explorer or the Data View assistant in ADE Assembler according to the column names. You can now use this variable to adjust the default column width according to the length of values as well as the column names. For this, you can set this variable to `"UseVariableNameWidth"`. **Note:** You can still drag the column separators and readjust the width. Default value: `"Interactive"`	

SKILL Functions

ADE Explorer and ADE Assembler	
calcValForRel	New
Retrieves the value of an output expression used in a reliability setup. You can use the value returned by this function in another output expression.	
vtimeterm	New
Returns the voltage of terminal at a specified time point or at all time points in the time domain.	
vhterm	New
Returns the voltage on terminal at the specified harmonic or at all the harmonics in the frequency domain.	
vfreqterm	New
Returns the voltage of terminal at a specified frequency or at all frequencies in the frequency domain.	
rmsTerminalVoltage	New
Calculates the root-mean-square voltage between two terminals for fast and regular envelop analysis.	

OCEAN Command

ADE Explorer and ADE Assembler	
ocnxlSetRelxEnabledForPreRun	New
Updates the reliability setup based on the corresponding setup from the current environment. Use this command in the pre-run script to print the reliability setup in the pre-run netlist.	

ICADVM18.1 ISR6

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Running Simulations with DSPF Files	Enhanced
The netlist creation for designs using DSPF files has been enhanced to traverse inside blackboxes and use the schematic name corresponding to each object to write the save statement in the netlist. This helps in probing the device terminals or objects inside them. The total current for a fingered device is now calculated as the sum of the current in all fingers.	
Managing Implicit Signals for Output Expressions	New
If an expression in the *Outputs Setup* tab uses a signal or net that does not already exist in the outputs table, a new row for that signal is added to the table. The type of this output is set to `Implicit`. By default, implicit signals are not visible in the outputs table or saved in the simulation results. You can choose to show, hide, or save these outputs.	
High-Performance Simulation Options form	Enhanced
A new simulation performance mode, *Spectre X*, is now available. The *Spectre X* mode allows you to massively distribute simulation workload, enabling greater speed and capacity.	
Analyzing Monte Carlo Results in Yield View	Enhanced
You can now view the mean ±K sigma results in the *Yield* results view for Monte Carlo Sampling simulations. Two new columns, *Mean -3Sigma* and *Mean +3Sigma* have been added to the results table in this view. By default, you can view the ±3 sigma results, but you can change the sigma value by using the *Set k Sigma* command in the context-sensitive menu for these columns. You can specify any integer between 0 and 100000 for the sigma value.	
Reliability Analysis with Gradual Aging	Enhanced
You can now save aging results for intermediate age points by selecting an option in the new *Save Intermediate Results* group box added for *Aging Options* on the *Basic* tab of the Reliability Options form.	
ADE Assembler Only	
Reliability Analysis in Run Plan	New

You can now set up reliability analysis in a run plan.	
Re-evaluating Expressions and Specifications	Enhanced
You can now select either *All* or *Partial Data* to re-evaluate both expressions and specifications after running a simulation, on the *Outputs Setup* tab of the *Outputs* pane. The other options are removed from the drop-down list.	

Environment Variables

ADE Explorer and ADE Assembler	
`maestro.explorer displayImplicitSignals`	New
Controls whether to show or hide the implicit signals in ADE Explorer. Default value: `nil`	
`maestro.gui saveimplicitSignalsDuringRun`	New
Specifies whether to regenerate implicit signals in ADE Explorer when you switch from *all* to *selected* in the *Select signals to output (save)* option in the Save Options form. Default value: `nil`	
`maestro.gui hideShowMultiProcess`	New
Controls whether to show or hide the option, *Multi-processing,* in the General section of High-Performance Simulation Options form, when Simulation performance mode is set to *Spectre X*. Default value: `nil`	
`maestro.gui showUnitsInResults`	New
Controls whether to hide or show the unit values in the measured results for the expressions in the Results tab. Default value: `t`	

SKILL Functions

ADE Explorer and ADE Assembler	
maeGetMTSBlock	New
Indicates whether the MTS options are enabled for the specified test.	

maeGetMTSMode	New
Enables the MTS options for the specified test.	
maeSetMTSBlock	New
Returns the requested MTS-related information about the specified library cell (block) or the instance.	
maeSetMTSMode	New
Enables local scoping for the specified library cell (block) or instance and sets the specified MTS options–models files to be scoped locally and the process parameters, such as scale and temp to be included in the simulation locally.	
maeUpdateImplicitSignals	New
Updates the implicit signals for the tests in the specified maestro session.	
relxGetRelxStage	New
Returns the currently running stage of the reliability simulation in the given session.	
relxGetStressFileDir	New
Returns the directory of the stress file.	

ICADVM18.1 ISR5

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Exporting Results to Excel	New
You can now export results in the `Excel File (.xlsx)` or `Ready for Pivot Excel File (.pivot.xlsx)` format that you can directly open in Excel. This feature is useful in the creation of PivotTables and PivotCharts for the analysis of simulation results.	
Choosing the Type of Data to be Saved for Signals	Enhanced
You can now specify the type of data, current, voltage, or both, by using the *Terminal Selection Type* drop-down list on the ADE Assembler Plotting/Printing Options form. This setting overrides the value of the `terminalSelectionType` environment variable.	
ADE Assembler Only	
Editing Multiple Tests	Enhanced
The support to copy settings from the reference test to other tests has been extended from Spectre only to other simulators, AMS, HSpice, and UltraSim, too.	

Environment Variables

ADE Explorer and ADE Assembler	
`adexl.results exportFormat`	New
Specifies the format to be used for the numeric values in the results exported from ADE Explorer or ADE Assembler. You can choose to export the result values without units, values with units in separate columns, or values with units in the same column as displayed on the *Results* tab. Default value: `"numbersOnly"`	

auCore.selection alwaysNameTerminalOutputs	New
Specifies whether to always add the `_I` suffix for the current type of signal outputs when the `terminalSelectionType` environment variable or the *Terminal Selection Type* drop-down list on the ADE Explorer Printing/Plotting Options form is set to `"Current"`. Default value: `nil`	
ADE Assembler Only	
maestro.gui detailTransposeViewShowTest	New
Controls whether to show the *Detail - Transpose with Test* result view type in the *Select the results view* drop-down list on the *Results* tab in ADE Assembler. By default, this result view type is hidden. You can display it when you need to export results to Excel for the purpose of PivotTable creation. Default value: `nil`	

SKILL Functions

ADE Explorer and ADE Assembler	
maeGetMappingForJobAndPoint	New
Returns a list containing the mapping of job IDs with the point IDs allocated to them.	
maeStmGenerateWaveforms	New
Generates preview stimuli waveforms for the stimuli added for a maestro cellview that contains stimuli definitions added through the Stimuli Assignment form.	
axlStmImportOasisStimulus	Removed
This function is not required now because the stimuli information that was saved by the Stimuli form in a previous release is now automatically read and assigned to the pins and global signals in the maestro view.	

OCEAN Command

ADE Explorer and ADE Assembler	
ocnxlLocalParametricSet	New
Adds a parametric set for local variables in a maestro view.	

ICADVM18.1 ISR4

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Enabling Local Scoping at the Instance Level	New
By using the MTS feature, you can now specify locally scoped models and simulator options at the instance level. To do so, you must first enable local scoping at the instance level. You can then specify the model libraries and various process parameters that you want to include in the simulation locally.	
ADE Assembler Only	
Working with Run Plans	Enhanced
■ **Reordering runs in the run plan:** You can now use the drag-and-drop feature to change the order in which runs are arranged in the Run Plan assistant. ■ **Copying simulation options:** You can use the new *Apply Simulation Options* command in the context-sensitive menu of run mode of a run to copy the options to other runs in the run plan.	

Environment Variable

ADE Explorer and ADE Assembler	
`maestro.simulation maxRDBSyncWait`	New
Specifies the maximum wait time (in seconds) for which ADE Assembler needs to wait for the results data to be used for the evaluation of `calcVal` expressions. This variable is particularly useful when the simulations are running over NFS and there could be a delay in retrieving the results. Default value: `60`	

maestro.gui_defaultAgingPlacementType	New
Determines whether to print the aging-specific model files before or after the standard model files in the netlist. Default value: `"After Standard Model"`	
maestro.distribute_jobCleanupTimeoutBeforeKill	New
Specifies the time for which ADE Explorer or ADE Assembler should wait before sending the `bkill` command to kill an ICRP job when a connection with an ICRP is lost or after 20 seconds have elapsed since it sent a command to an ICRP to clean up and exit. Default value: `0`	
ADE Explorer Only	
maestro.explorer_defaultOutputColumns	New
Specifies the default list of columns to be shown on the *Outputs Setup* tab in ADE Explorer. The specified columns remain visible for a new maestro view opened in Explorer or for an ADE Assembler to Explorer transition. Default value: `"\"Name\" \"Type\" \"Details\" \"Value\" \"Plot\" \"Save\" \"Spec\""`	

SKILL Functions

ADE Explorer and ADE Assembler	
maeGetAllPlottingTemplates	New
Returns a list of all plotting templates saved in the given session.	
maePlotWithPlottingTemplate	New
Plots the results for the given history using the specified plotting template.	
maeSaveImagesUsingPlottingTemplate	New
Plots and saves the results for the given history using the specified plotting template.	
axlToolSetOpPointInfo	New
Adds the specified `oppoint` type item to the Output Setup table in a test setup.	

ICADVM18.1 ISR3

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Running simulations for DSPF Files or DSPF Cellviews	New

The following enhancements have been done to support running simulations for DSPF files or DSPF cellviews:

- For DSPF files, you can create a `.scs` file containing a `dspf_include` statement for each `.dspf` file, and then sweep the `.scs` file as done for the model files.
- For DSPF cellview, you can bind it to the config view in the Hierarchy Editor. For multiple cellviews, you can sweep multiple DSPF cellviews using the config sweep feature.
- At times, the syntax used to save nets and instances in the netlist is different from the syntax used in DSPF. The following variables have been provided to modify the save statement in the netlist to match the syntax in DSPF:
  - spfFileNameMappingFormat
  - spfFileTermDelimiter
  - spfFileHierDelimiter
  - spfFileTerminalMapping
  - spfFileFingerDelimiter
  - spfFileDevicePrefixForMacroModels
  - spfFileDeviceDefaultPrefix
  - spfFileDevicePrefixForTermCurrent
  - spfFileDevicePrefixForInstCurrent
  - spfFileNetMapping
  - spfFileDevicePrefixToOverrideCDF

Reliability Options Form	Enhanced

The Reliability Options form has been revamped for improved usability. The options have been regrouped in four tabs to keep the related options together.

Stimuli Assignment Form	Enhanced

The Setup Analog Stimuli form has been renamed as Stimuli Assignment. This form has also been redesigned to allow previewing the stimuli waveforms without creating a testbench while running a simulation. The form contains two sections:

- *Stimuli Authoring* in which you can create and edit stimuli
- *Pin Assignments* in which you can assign stimuli to pins and global nets in your design

You can click *Setup Library Task Assistant* on this form to access quick help that provides task-based information to get you started with this form.

Related environment variables:

- `maestro.stimuliTool stopTime`
- `maestro.stimuliTool maxStopTime`

Checks and Asserts Analysis	Enhanced

- The new Summary check box on the Checks/Asserts results shows the count for each check or assert violation across all instances for each combination of corner and analysis name.
- The result values of checks and asserts now correctly indicate the completion status of checks as listed below:
  - `undefined` - To indicate that the *dochecklimit* device checking option on the *Checks* tab of the Simulator Options form is disabled; or the check is disabled in the constraint view
  - `disabled` - To indicate that no checks or asserts are run, and no SQL database is generated; or there is an error in the checker definition; or the checker is not run

Defining Locally Scoped Models and Options	Enhanced

The *Multi-Technology Mode* check box has been removed from the Choosing Simulator form.

You can now enable MTS options by selecting the *Enable locally scoped models and options (MTS)* check box on the MTS Options form. When you edit the MTS settings, changes are saved in the state files.

Save Options Form	Enhanced

The Save Options form includes a new tab, *Save By Subckt*, which you can use to customize options to save subcircuit instances and Pcells.	
Creating a Datasheet for a Checkpoint	Enhanced
A new option, *Replace Using Template*, has been added to the Create Datasheet form. Using this option, you can include the graphs in the datasheet from a plotting template.	
ADE Assembler Only	
Editing Multiple Tests	New
You can use the new Multi-Test Editor to view and compare the set up of multiple tests in a common tabular format. While comparing, you can focus on differences in any specific setting, or copy a particular setting from a reference test to other tests in the setup. Related environment variable: `maestro.commonTestSetup defaultComponents`	
Legato Reliability Solution	Enhanced
■ Electro-thermal Analysis: You can now enable the electro-thermal analysis on the *Electro-Thermal* tab of the Transient Options form. This analysis considers all power sources on a whole chip and the thermal interactions between devices. It uses the thermal technology to create a thermal model of the die based on the chip structure and thermal properties of the die stack. ■ N+N Monte Carlo: You can use the drop-down list in the *Reliability Mode* group on the Monte Carlo run options form to enable the N+N flow that runs both stress and aging simulations with Monte Carlo analysis. As compared to the 1+N flow, the stress results are shown for all stress points in this flow. Related environment variable: `maestro.monte reliabilitymode`	

Environment Variables

ADE Explorer and ADE Assembler	
auCore.selection terminalSelectionType	New
Specifies the type of data–current, voltage, or both–to be saved or plotted for an output.	
ADE Assembler Only	
maestro.gui manualResultsViewRefresh	New
Enables or disables the feature that allows you to control the refresh of the *Results* tab while a simulation is in progress. When the feature is enabled, the *Manual Refresh* check box is added to the Run Options form. Default value: `t`	
maestro.assembler defaultOutputColumns	New
Specifies the default list of columns to be shown on the *Outputs Setup* tab in ADE Assembler. Default value: `"\"Test\" \"Name\" \"Type\" \"Details\" \"EvalType\" \"Plot\" \"Save\" \"Spec\" \"Weight\" \"Units\" \"Digits\" \"Notation\" \"Suffix\""`	

SKILL Function

ADE Explorer and ADE Assembler	
maeDeleteSimulationData	New
Deletes the simulation results data for the given history.	
asiGetPageCallBack	New
Returns the callback function name when changing tabs in a multi-tab form.	
asiSetPageCallBack	New
Returns the status (successful or failed) of the specified callback function, when changing tabs in a multi-tab form.	
densityEstimateWaveform	New
Returns the density estimator waveform of a histogram.	
asiStmSupportWaveformGeneration	New
Determines whether the Stimuli Assignment form supports waveform preview functionality or not.	

asiStmGenerateNetlist	New
Generates netlist, as part of preview waveform generation for a stimuli and saves it at the specified psf directory path.	
asiStmRunSimulation	New
Runs a simulation by using the netlist generated by SKILL function `asiStmGenerateNetlist`, and writes the psf waveform data at the specified path.	
asiStmGenerateWaveform	New
Generates preview waveforms for the specified stimuli under the given directory path.	

ICADVM18.1 ISR2

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Enable/Disable Multiple Corners	New
You can enable or disable multiple corners from the *Setup* Assistant in ADE Explorer or *Data View* Assistant in ADE Assembler.	
Exporting Results of Mismatch Contribution Analysis	Enhanced
You can apply filters to view only the relevant mismatch contribution data on the *Mismatch Contribution* tab. The *Export to CSV* feature is enhanced to retain the filters applied to view a selected set of results. When the results of the mismatch contribution analysis are exported to a CSV file, they are saved in the same format as displayed in the *Mismatch Contribution* table in the Sensitivity Analysis window.	

Environment Variable

ADE Explorer and ADE Assembler	
`maestro.cpupdtr keepHistoryData`	New
Specifies whether or not to copy the histories while copying a maestro cellview. If you do not want to copy the histories with maestro cellviews and improve performance, set this variable to `nil`. Default value: `t`	

SKILL Function

ADE Explorer and ADE Assembler	
asiGetAnalysisFormObj	New
Returns the current analysis selected in the *Choosing Analysis* form for the simulation session.	

ICADVM18.1 ISR1

What's New and Enhanced

Features

ADE Explorer and ADE Assembler	
Backannotating the Estimated Thread Count from Job Policy to the High-Performance Simulation Options form	New
On the High-Performance Simulation Options form, when the *Multi-Threading* setting for the APS simulation performance mode is set to *Manual*, the tooltip of the *#Threads* field shows the estimated thread count. This count is the CPU estimate value backannotated from the Job Policy Setup form.	
Sweeping Schematic Pcell Parameters	New
You can now sweep Pcell parameters in the same way as done for other device parameters. The sweep values are used to create multiple design points for simulations. During netlist creation, ADE Explorer and ADE Assembler creates subcircuits for each swept Pcell parameter value, and then runs simulation. This feature is helpful for advanced node designs that contain Pcells with parameters to adjust circuit topology changes, such as the number of stacked components represented by a device. You can run local or global optimization to identify the best value for the Pcell parameters to optimize your designs.	
Exporting Results	Enhanced
Earlier, you could export the results by using the *Export Data in CSV or HTML format* command on the toolbar or the *Results* tab. You can now also use the `Ctrl+C` keyboard shortcut to copy a selected subset of results from the results table and then the `Ctrl+V` shortcut to paste them in any other application.	
ADE Assembler Only	
Improved Display of Overridden Specifications in Results	New
When the `showOverriddenValueForSpec` variable is set to `t`, if the corners in the setup override the global specification value for an output, the *Spec* column in the results view now shows the global specification and the overridden value.	

Environment Variables

ADE Explorer and ADE Assembler	
maestro.distribute_spectreMTNumberUsed	New
Specifies the behavior of the `+mt` option for Spectre APS when on the High-Performance Simulation Options form, the *Manual* multi-threading mode option is selected for APS Spectre simulation performance. Default value: `"Estimated"`	
adexl.distribute_isLSFMemSwapHostLimit	New
Specifies the default value for the *Memory Host Limit* check box in the *Job Policy Setup* form, when the *Distribution Method* is set to `LBS`. Default value: `nil`	
ADE Assembler Only	
maestro.gui_showOverriddenValueForSpec	New
Controls the display of the global specification as well as the overridden specification value for outputs in the simulations results. Default value: `nil`	

SKILL Functions

ADE Explorer and ADE Assembler	
maeGetOverallSpecStatus	New
Returns the overall specification status for the current history.	
maeWriteDatasheet	Enhanced
You can now use this function to write datasheets for the results that contain a group of histories. For example, now you can use this function to print results of a single or all runs in the run plan.	

Removed Features

ADE Explorer and ADE Assembler	
AMS simulations using AMS plugin in Virtuoso Hierarchy Editor	Removed
The functionality of running AMS simulations by using the AMS plugin in Virtuoso Hierarchy Editor has been discontinued from this release. It is recommended to *run AMS simulations* from Virtuoso ADE Explorer or ADE Assembler. For this, before running simulations, set the simulator on the Choosing Simulator form as `ams` and specify Spectre models in the Add Model Files form. ADE Explorer or ADE Assembler automatically runs UNL to create a netlist.	

Documentation Updates

Documents related to AMS Designer Environment	Removed
The following documents are no longer available from this release onward: ■ Virtuoso AMS Designer Environment User Guide ■ Virtuoso AMS Designer Environment SKILL Reference ■ Virtuoso AMS Designer Environment Tutorials	

ICADVM18.1 Base Release

What's New and Enhanced

Features

ADE Assembler Only	
Simulating Analog Defects	New
As a part of Legato Reliability Solution, ADE Assembler supports the fault analysis, which provides the capability to run defect-oriented tests on analog IC designs. These tests allow you to evaluate the ability to eliminate a die with manufacturing defects and resulting test escapes that cause field failures. You can also use these tests to optimize wafer tests by reducing the number of tests required to achieve the target defect coverage.	
Simulation Planning and Coverage	New
An important requirement for project sign off is to ensure that all the design simulations in ADE Assembler are run using the efficient (or predefined) sets of operating conditions (corners, sweeps and model files) in accordance to the project. ADE Assembler and ADE Verifier now allow you to plan and create project-specific master setups that you can then use to create setups for individual tests in ADE Assembler. This enables quick and efficient setup creation at the test level. Verifying the top-to-down progress of your design using ADE Verifier, you can compare the implementation histories with defined operating conditions from master setup and review the progress of your design based on Analog Coverage percentage reported by ADE Verifier. To support this feature, a new assistant, Setup Library, has been provided in ADE Assembler and ADE Verifier. For more information, see Working with the Setup Library in *Virtuoso ADE Assembler User Guide* and Verifying the Design Against the Specified Setup in *Virtuoso ADE Verifier User Guide*.	
Task Assistant – A Quick Help Functionality	New
A new task-based quick Help functionality, Task Assistant, has been introduced in the Setup Library Assistant in ADE Assembler and ADE Verifier. You can access this quick Help by clicking the *Setup Library Task Assistant* button available on the toolbar in Setup Library assistant. The Task Assistant will help you get started with the Setup Library assistant in a task-oriented way.	

Spectre Interactive Environment	New
Spectre Interactive Environment is a new debugging tool introduced to debug Spectre simulations interactively by using events, triggers, and breakpoints. You can run this tool in the standalone mode or from the results of a simulation run in ADE Assembler.	
Support for Worst Case Corners in Run Plan	New
You can now use the `Worst Case Corners` run mode in the run plan to identify the corner conditions at which you want to measure the performance of your design. After this, you can run a subsequent simulation with any other run mode by using the identified worst case corners.	
Advanced Aging Analysis	Enhanced
The aging analysis for reliability has been enhanced to include other effects, such as process variation, that contribute to device degradation, besides electrical stress. To account for process variation, you can combine advanced aging analysis with Monte Carlo run mode. **Note:** Currently, ADE Assembler supports only 1 + N flow for aging analysis. That means, a nominal stress analysis is run first. Next, Monte Carlo analysis is performed on the aging analysis. No variations are performed on the stress analysis.	
ADE Explorer and ADE Assembler	
Creating Plotting Templates	New
You can now create plotting templates in ADE Assembler to plot resultant waveforms, signals, or scalar values in a specified format in the Virtuoso Visualization and Analysis XL window. All the formatting, markers, layout changes, and other interface changes will be retained as specified in the plotting template.	
Tooltips for Simulator Options and Analysis Options Forms	New
The Simulator Options form and the various analysis options forms have been enhanced to show tooltips for the form fields. This helps in identifying the purpose and the possible values for each field. Related Environment Variable: `spectre.envOpts displayToolTip` **Note:** This feature is available with MMSIM16.1 ISR13, SPECTRE17.1 ISR3 or higher versions.	
Smart View	New

Starting from IC6.1.8 release, Parasitic Aware Design also supports Smart View, which is a new version of the extracted view, that you can use to simulate and analyze post-layout designs.	
Save by Subckt Instances Assistant Wildcard Support	Enhanced
The Save by Subckt Instances Assistant now supports the ability for you to enter any string, including wildcards, for saving instance information.	
OSS- and Cellview-based Netlisting in AMS Simulations	Removed
The OSS- and Cellview-based netlisting in AMS simulations have been discontinued from this release. It is recommended to run the AMS Unified Netlister (AMS UNL), the default AMS netlister.	

Documentation Updates

Environment Variables for Spectre Simulator Options	New
The descriptions for environment variables corresponding to fields in the Simulator Options are now available in the chapter Environment Variables for Spectre Simulator Options Form.	

Virtuoso ADE Verifier

This chapter provides a high-level overview of the new features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso ADE Verifier User Guide*
- *Virtuoso ADE Verifier SKILL Reference*
- *Tutorial: Virtuoso ADE Verifier Workflow*

ICADVM20.1 Base Release

What’s New and Enhanced

Features

Hierarchy Support in Implementation Sets	Enhancement
Implementation sets now help you to define dependencies for implementations. You can organize and group implementations and their dependencies, and subsequently, their runs, in a hierarchical structure.	
Usability Enhancements in the Preferences Form	Enhancement
Two new tabs, *Planning* and *Distribution*, have been added to the Preferences form for a better usability experience. The existing preference options have been redistributed accordingly.	

ICADVM18.1 ISR13

What’s New and Enhanced

Features

Support for Calculations through Custom Implementations	New
ADE Verifier now supports custom implementations, which you can use to perform calculations on output expressions within an ADE Verifier cellview. **Environment Variables**: `customLib`, `customCell`, `customView`, `customHistory`	

ICADVM18.1 ISR12

What's New and Enhanced

Features

Support for Reliability Verification	New
ADE Verifier now supports verification and evaluation of reliability analysis in implementations. You can control the display of reliability outputs using settings in the new group box, *Reliability Outputs*, that has been added to the *General* tab of the Preferences form. **Environment Variables**: displayFresh, displayStress, displayAge, displayRelOutputAges	
Improvements in the Information Assistant	Enhancement
The Information Assistant has been enhanced to display details of the selections in a hierarchically organized format. Additionally, the assistant has been enhanced with the *Setup*, *Run*, *Results*, and *Snapshots* buttons to help you navigate to the corresponding page.	
Loading Partial Results from Incomplete Histories	New
In ADE Verifier, you can now load partial results of running or incomplete simulations. The *Run* page shows the status, and the *Results* page shows the partial results. **Environment Variable**: allowLoadingIncompleteHistory	
Modifications in the Environment Setup to Run Simulations	New
A new group box, *Environment Variables*, has been added to the *Run* tab of the Preferences form. This group box includes a table, using which you can now modify the Verifier environment to run simulations. **Environment Variable**: envVarConfigFile	
Mapping Status	New
A new column, *Mapped*, has been added to the *Requirements* pane of the *Setup* page. The column shows a comma-separated list of mappable items that are mapped to the current requirement. A mappable item can be an implementation, run, reliability setup, test or an output. The values in this column allow you to quickly validate if all the requirements are mapped to the correct implementation.	

Filtering Results by ReadOnly Status	New
The *Results* page now has a check box in the *ReadOnly* column. Selecting this check box allows you to filter the read-only requirements from the results.	

SKILL Functions

New Setup Library Assistant Functions	New
Several new SKILL functions, with the prefix `sla`, have been added to help you work with the Setup Library Assistant in the `nograph` mode.	
Functions for opening and saving setup library views	New
slaOpenOrCreateView: In the edit mode, opens the specified view in the edit mode if the view exists. Otherwise, it creates a new view. In the read-only mode, the function opens the provided view in read-only mode. In both cases, invoking this function checks out the ADE Verifier license. **slaSaveAndCloseView**: Saves all the changes in the setupdb after adding or removing components in the setup library assistant. Invoking the `slaSaveAndCloseView` function releases or checks in the ADE Verifier license.	
Functions for creating top-level components	New
slaCreateCornerSetup: Creates a corner setup in setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. **slaCreateSweepSetup**: Creates a sweep setup in a setup library view that is opened the `slaSaveAndCloseView` function in edit mode. **slaCreateVerificationSpace**: Creates a verification space in the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.	

Functions for adding corners, variables, or model files	New
slaImportCorners: Imports corners into the setup library corner setup opened using the `slaSaveAndCloseView` function in edit mode. The file must be in CSV or SDB format. **slaImportSweeps**: Imports sweeps into the corner setup in a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. The supported file extensions are `csv` and `sdb`. **slaAddSweepVariable**: Adds a sweep variable to a sweep setup in the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. **slaAddCornerVariable**: Adds a corner variable to a corner of a corner setup in the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. The function creates the specified corner if it does not exist in the corner setup. **slaAddCornerModelFile**: Adds a model file to a corner within a corner setup in the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. The function creates the specified corner if it does not exist in the corner setup. **slaAddDocument**: Adds a document to a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.	

Functions for reading setups from the setup library view	New

slaGetAllDocuments: Retrieves a list of documents from the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

slaGetCornerModels: Retrieves a list of corner models of corner setup from the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

slaGetCornerSetupCorners: Retrieves a list of corners for a specified corner setup from the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

slaGetCornerSetups: Retrieves a list of corner setups from the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

slaGetCornerVars: Retrieves a list of corner variables of the specified corner setup from the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

slaGetDocumentAbsolutePath: Retrieves the absolute path of a document in the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

slaGetSweepSetupVars: Retrieves a list of sweep setup variables from the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

slaGetSweepSetups: Retrieves a list of sweep setups from the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

slaGetVerificationSpaces: Retrieves a list of verification spaces from the setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.

Functions for removing setups from the setup library view	New
slaRemoveCorner: Removes a corner from the corner setup in a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. **slaRemoveCornerModel**: Removes a corner model file from the corner setup in a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. **slaRemoveCornerSetup**: Removes a corner setup from a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. **slaRemoveCornerVariable**: Removes a corner variable from corner setup of a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. **slaRemoveSweepSetup**: Removes a sweep setup from a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. **slaRemoveSweepVariable**: Removes a sweep variable from corner setup of a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode. **slaRemoveVerificationSpace**: Removes a verification space from a setup library view that is opened using the `slaSaveAndCloseView` function in edit mode.	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

Saving and Loading the Run Summary Data of Read-Only Cellviews	New
The run summary data file is now saved to a location that is determined by the read-only or editable state of the cellview.	
Including Outputs That are Disabled in maestro Cellviews	Enhancement
While adding implementations in Verifier, you can include the outputs that have been disabled in maestro cellviews. As a result, the preference option *Include disabled run or test for implementation* has been renamed to *Include disabled run, test or output for implementation*.	
Filtering Implementations Based on Visible Requirements	Enhancement
A new context menu command, *Show – Filter On Requirements*, has been added to the *Implementations* pane of the *Setup* tab. This command selectively displays only those outputs in an implementation that are mapped to the requirements visible in the *Requirements* pane.	
Supporting Merged Cells when Importing Excel Files	Enhancement
Earlier, when you imported a Microsoft Excel file, any merged cells would be split into individual cells, and only the first cell of these individual cells would contain the value from the merged cell. The remaining cells would be empty. Now, when you import Excel files with merged cells, the merged cell is split into individual cells after the import and all the individual cells contain the same values as the merged cell.	
Options Removed from the General page of the Preferences form	Removed
The preference option *Show 'Verification Space' on Setup tab and 'Coverage' column on Results tab by default* is no longer required and has been has been removed.	

Environment Variables

`reqColumnOrder`	Enhancement
The default values *Verification Space* and *Cellview Holder* have been renamed to *VerificationSpace* and *CellviewHolder*, respectively.	
`reqHiddenColumns`	Enhancement
The default value *Cellview Holder* has been renamed to *CellviewHolder*.	
enableCoverage	Removed
When set to "YES", Verifier calculates coverage for verification spaces.	

SKILL Functions

verifExportJson	New
Exports the Verifier session as a `json` file. The `json` contains details about the session, including the requirements, implementations, mappings, and simulation results.	
verifEvaluateResults	New
Evaluates all requirements and helps in a force update of the results in special cases.	
verifIsSessionModified	New
Checks whether the setup has been modified after the last time it was saved in the given session.	
verifIsValidSession	New
Confirms if the given session is a valid ADE Verifier session.	
verifIsBatchRunProcess	New
Determines if Virtuoso has been launched for an ADE Verifier batch run. Returns `t` if the remote child process is currently running. You can use this function in your `.cdsinit` file or in custom SKILL code.	
verifCopyAndUpdateResultsFromUserDefinedDirectory	New
Copies the run summary data files from the user-defined directory to the `results` directory of the current cellview. Additionally, updates the current session setup to restore the run summary data file into the `results` directory of current cellview.	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

Reading Histories from Multiple Project Directories	New
You can now add additional project directories for every implementation when you have the required read-access permissions. These additional project directories allow you to access read-only histories owned by other users from the ADE Assembler setup. **Environment Variable**: **additionalProjectDirectory**	
Selecting Columns to Display in Results Tab	Enhancement
In addition to the existing columns, the *Results* tab now shows the *VerificationSpace*, *CellviewHolder*, *Scope*, *Domain*, *Description*, and the *ReadOnly* columns. By default, these columns are hidden and replicate the details shown on the *Setup* tab.	
Viewing the Status of Running Simulations in LSCS Mode	Enhancement
When you run an implementation that uses the LSCS mode for netlist generation, the *Status* column in the *Run* tab displays the progress of the simulation and netlist generation.	

ICADVM18.1 ISR9

What's New and Enhanced

Features

Adding Custom Context Menus	New
You can now define menu actions for context menus and add these custom menus to the *Requirements* and *Implementations* pane in the *Setup* tab, the run table in the *Run* tab, and the results table in the *Results* tab. **SKILL Function**: **verifGetMappableType**	
Running Simulations in ICRP or LSCS Mode	New
You can now choose the appropriate job policy for all the implementation runs. To support this, the following lists are added to the *Run* tab of the Preferences form: ■ *Override job control mode* ■ *Override simulation job policy* ■ *Override netlisting job policy*	

Environment Variable

`simulationTimeout`	New
Specifies the simulation as finished and displays an error message if an update is not received from ADE Assembler within the specified `simulationTimeout` seconds.	

SKILL Functions

verifEnableDebug	New
Enables additional debug logging of various predefined categories.	
verifDisableDebug	New
Disables one or more categories for debug logging.	
verifGetDebug	New

Retrieves the names of enabled debug categories or all possible categories.

ICADVM18.1 ISR8

What’s New and Enhanced

Feature

Authenticating the vManager Server	Enhancement
You can now establish a secure connection with the vManager server by using an encrypted *Token*.	

ICADVM18.1 ISR7

What's New and Enhanced

Features

Reporting Values Based on Design Points Defined in Verification Spaces	Enhancement
Now, selecting *Overall Coverage* on the Verifier toolbar updates the values of *MinValue*, *MaxValue*, and *OverallStatus* on the *Results* tab. These values are based only on the design points that are defined in verification spaces. **Form**: *Preferences*	
Editing Disabled Tests or Runs	Enhancement
A new check box, *Allow editing the enabled or disabled state for implementation run or test*, is added to the *Run* tab of the Preferences form. You can select this to edit the enabled or disabled state of all tests or runs in an implementation. **Environment Variable**: `allowEditingDisabledState`	

Environment Variable

`includeRequirementId`	New
Shows the requirement ID in the *Result Information* section of the HTML report.	
`maxFailures`	New
Controls the maximum number of test or output failures to be reported in the Information assistant.	

SKILL Functions

verifGetImpSetPreRunScript	New
Retrieves the pre-run script filename for an implementation set.	
verifSetImpSetPreRunScript	New
Sets the pre-run script filename for an implementation set.	

ICADVM18.1 ISR6

What's New and Enhanced

Feature

Using Snapshots for Verification	New
You can now save time-stamped snapshots of your verification project to back up data, view filtered results, make comparisons, or restore previous versions of your setup. To support this, the following have been added in ADE Verifier: ■ *Snapshots* tab ■ *Snapshots* toolbar ■ *Snapshots Editor* assistant ■ *Snapshots* workspace	

Environment Variables

`estRunTimeDisplayFormat`	New
Sets the *Display Estimated Run Time as* option to *Seconds* or *Formatted Time*.	
`historyNamePrefix`	New
Adds the specified prefix to the ADE Assembler history name.	
`appendDate`	New
Adds the current date as a suffix to the history name.	
`appendTime`	New
Adds the current time as a suffix to the history name.	
`toleranceProperties`	New
Controls the properties to which the tolerance values are applied.	

SKILL Functions

verifAreSnapshotsEnabled	New
Checks if snapshots are enabled in the specified Verifier session.	
verifCreateSnapshot	New
Creates a new snapshot `'*.snapz'` that contains the following: An archive of the current requirements, mapping, and results as shown in the *Results* tab. This archive is stored in `.json` format. A copy of the Verifier setup (`settings.v3`). A copy of the results directory.	
verifCreateSnapshotConfiguration	New
Creates a snapshot configuration that contains the setup for *Abs Tolerance*, *Rel Tolerance* and *Show* filters on the *Snapshots* tab.	
verifDeleteSnapshot	New
Deletes the specified snapshot from the specified Verifier session.	
verifDeleteSnapshotConfiguration	New
Deletes the specified snapshot configuration with the specified name.	
verifExportSnapshotsToExcel	New
Exports the snapshots to a Microsoft Excel file.	
verifGetReferenceSnapshot	New
Retrieves the snapshot from the specified Verifier session.	
verifGetSnapshot	New
Retrieves the disembodied property list (DPL) for the specified snapshot from the specified Verifier session.	
verifGetSnapshotAbsoluteTolerance	New
Returns the absolute tolerance value that is specified when comparing snapshot values.	
verifGetSnapshotComment	New
Returns the comment text for a snapshot.	
verifGetSnapshotRelativeTolerance	New
Returns the relative tolerance value used when comparing snapshot values.	

verifGetSnapshots	New
Retrieves the list of snapshots from the specified Verifier session.	
verifGetSnapshotsData	New
Retrieves the disembodied property list for snapshots from the specified Verifier session.	
verifIsSnapshotLocked	New
Returns the locked status of the specified snapshot.	
verifIsSnapshotVisible	New
Returns the visibility status for the specified snapshot.	
verifRenameSnapshot	New
Renames a snapshot with the specified name.	
verifRestoreFromSnapshot	New
Restores a Verifier session from a snapshot, along with any stored results.	
verifSetReferenceSnapshot	New
Sets the reference snapshot in the specified Verifier session.	
verifSetSnapshotAbsoluteTolerance	New
Sets the absolute tolerance for snapshot comparison in a Verifier session.	
verifSetSnapshotComment	New
Sets the comment for a snapshot in a Verifier session.	
verifSetSnapshotConfiguration	New
Saves the current *Show* list configuration with the specified name and filters the *Snapshots* tab items with the specified configuration.	
verifSetSnapshotLocked	New
Locks or unlocks the specified snapshots in the given Verifier session.	
verifSetSnapshotRelativeTolerance	New
Sets the relative tolerance for snapshot comparison in a Verifier session.	
verifSetSnapshotVisible	New
Sets the visibility of a snapshot in a Verifier session. A visible snapshot appears as a new column in the *Snapshots* tab.	

verifSetSnapshotsEnabled	New
Enables or disables snapshots in the specified Verifier session.	

ICADVM18.1 ISR5

What’s New and Enhanced

Feature

Showing Disabled Tests or Runs	New
A new check box *Include disabled run or test for implementation* is added to the *General* tab of the ADE Verifier Preferences form. You can select this setting to show all tests or runs, including disabled tests or runs, in an implementation.	

Environment Variables

`simMode`	New
Specifies the default run mode when you open a cellview. This run mode is saved or loaded to and from the settings.v3 file.	
`filterOptions`	Enhanced
Enables the default set of filters in case of new Verifier cellviews.	

ICADVM18.1 ISR4

What's New and Enhanced

Features

Running Simulations based on Time and Priority	Enhanced
You can now change the distribution of simulation jobs in ADE Verifier. To support this enhancement: ■ Two new columns, *Priority* and *EstRunTime* are added to the Implementations pane to allow changing the distribution of simulation jobs. ■ The*Run Controls* group of settings is added to the Run tab in the Preferences form. ■ The following new SKILL functions are now available: ❑ verifGetImpEstRunTime ❑ verifGetImpPriority ❑ verifSetImpEstRunTime ❑ verifSetImpPriority ❑ verifUpdateImpEstRunTime	
Using vPlanx Filenames	Enhanced
You can now use UNIX shell variables and or relative paths to specify the vPlan file name in the *vManager Setup* form.	

Environment Variable

`autoCreateDataSheet`	New
Controls the number of requirements that Verifier can load as one block from vManager during a query to retrieve requirements.	

SKILL Functions

verifCheck	New

Checks whether implementations changed since last run or reload by Verifier.	
verifCheckImp	New
Checks for changes of the specified implementation and loaded results.	
verifStop	Enhanced
Stops simulations for one or more implementations.	

ICADVM18.1 ISR3

What’s New and Enhanced

Features

Verification of Mixed-Signal Designs	New
You can now verify mixed-signal design projects by using the bi-directional connection between Virtuoso ADE Verifier and Cadence© vManager™ Metric-Driven Signoff Platform, a digital verification tool.	
Editing Multiple Requirements	Enhanced
You can edit multiple requirements in a verification plan using the *Requirements Editor* assistant. You can select these requirements and update them with the values that you specify in the assistant.	
Running Simulations on Referenced Cellviews	Enhanced
You can now run simulations on referenced cellviews by using *Run* from the toolbar.	
Enabling Mapping Mode	Enhanced
A new command *Enter Mapping Mode* has been added to the *Tools* menu and the toolbar to enable the Mapping Mode.	
Modifying Implementations	Enhanced
The *Replace Implementations* command is renamed to *Modify Implementations*. Using this command, you can select, replace, and modify multiple implementations.	
Enabling Simulation Runs for Multiple Implementations	Enhanced
A new *Run* check box is added to the *Modify Implementations* and *Add Implementation* forms. Selecting this check box enables simulation runs for single and multiple implementations.	

Environment Variables

itemLimit	New
Controls the number of requirements that Verifier can load as one block from vManager during a query to retrieve requirements.	
hostname	New
Specifies the hostname of the vManager connection.	
port	New
Specifies the port of the vManager connection.	
timeout	New
Controls the time after which the vManager connection is disconnected if the vManager server is unresponsive.	
username	New
Specifies the username for logging into the vManager server.	
password	New
Specifies the password for logging into the vManager server.	
project	New
Specifies the project to use on the vManager server.	
vPlan	New
Specifies the vPlan file associated with the Verifier cellview.	

SKILL Functions

vManager Functions	New
New SKILL functions have been added to let you manage the connection with vManager from Verifier.	

ICADVM18.1 ISR2

What's New and Enhanced

Features

Creating Outputs using Selected Requirements	New
You can add outputs into implementation cellviews by dragging requirements and dropping them on a test in an implementation. The newly added outputs are mapped to corresponding requirements automatically and added to the `maestro` cellviews in ADE Assembler.	
Distributed Processing of Simulation Runs	New
You can now specify the external program to use when running Verifier in the batch mode. You can also use `bsub` options to ensure that the launched Virtuoso sessions run on farm. Form: *Preferences* Environment Variable: `binaryName`	
Customizing the Displayed Columns	New
You can now show, hide, or reorder the columns of the *Setup*, *Run* and the *Results* tabs.	
New Title Naming Rules for Requirements	Enhanced
The naming rules for requirement titles have been updated to ensure that the titles contain unique hierarchical paths with slashes used as separators. The unique hierarchical paths make the verification plan compatible with `vplan` in vManager-based flows. The existing requirement titles are changed according to the new rules. See also Requirement Titles.	
Running Individual Implementations	Enhanced
The *Run* column now contains *Run* check boxes for individual implementations on the *Run* tab. This check box is synchronized with the *Run* check box on the *Setup* tab for every implementation. Consequently, a new column *Go* has been added to contain the *Run* buttons for individual implementations.	
Simulating Batch Runs	New
The run mode *External Run* is renamed to *Batch Run*.	

Environment Variables

reqColumnOrder	New
Controls the order in which columns in the *Requirements* pane appear in the *Setup* tab	
reqHiddenColumns	New
Specifies the columns that are hidden by default in the *Requirements* pane of the *Setup* tab.	
impColumnOrder	New
Specifies the order in which columns in the *Implementations* pane appear in the *Setup* tab.	
impHiddenColumns	New
Specifies the columns that are hidden by default in the *Implementations* pane of the *Setup* tab.	
runColumnOrder	New
Specifies the order in which columns appear in the *Run* tab.	
runHiddenColumns	New
Specifies the columns that are hidden by default in the *Run* tab.	
resColumnOrder	New
Specifies the order in which columns appear in the *Results* tab.	
resHiddenColumns	New
Specifies the columns that are hidden by default in the *Results* tab.	

ICADVM18.1 ISR1

What's New and Enhanced

Features

Mapping Mode	New
A new interactive mapping mode has been added. You can enter and exit this mode by pressing the 'm' key. In this mode, selection of requirements and implementations is automatically mapped.	
Exporting CSV and Excel files with Hierarchy Number	New
A new check box *Hierarchy* has been added to the *What to Save* section of the *Export – CSV* and *Export – Excel* forms. Enabling this check box saves the hierarchy number of the requirement in the exported file. This check box is not selected by default.	
Evaluating the Metric Prefix	New
A new check box *Evaluate Metric Prefix* has been added to the *General* tab of the *Preferences* form. When enabled, it considers the metric prefix of the required unit during the specification check and evaluation, and ensures that the overall minimum and maximum values for a requirement use the same metric prefix as the used specification. **Form**: *Preferences* **Environment Variable**: `evaluateMetricPrefix`	
New Bindkeys	New
New menu access keys have been added for the frequently needed commands.	

Environment Variables

`fields`	New
Specifies a set of predefined fields to be exported in the CSV and Excel files. `"verifier.export" "fields" 'string "Overall Status,Mapping,History,Result Data Age,Min Value,Typical Value,Max Value,Passed,Failed,No Results,Unmapped,Verification Space,Coverage"`	
`addSummaryInformation`	New

Controls the export of the 'Summary Information' field in the exported CSV and Excel files `"verifier.export" "addSummaryInformation" 'boolean t`	
evaluateMetricPrefix	New
Controls the evaluation of the metric prefix of the required unit during the specification check and evaluation. It ensures that the overall minimum and maximum values for the requirement use the same metric prefix as the used specification. `"verifier.preferences" "evaluateMetricPrefix" 'boolean nil`	
displayReqHierId	New
Controls the display of hierarchical numbers of requirements in the *Setup* tab. `"verifier.gui" "displayReqHierId" 'boolean t`	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Performance Improvements	New
Verifier has been fully updated for usability, performance and capacity, and as a part of these improvements, the menus, toolbar, assistants, forms, environment variables and SKILL functions have been updated consequently.	
Verifier in Virtuoso	New
Verifier is now a part of Virtuoso and no longer a standalone application. Because of this integration, a new *Launch* menu is displayed in the Verifier window and Verifier has acquired all benefits of the Virtuoso flow.	
External Reference to Cellviews	New
The new External Reference flow in Verifier supports unidirectional references to multiple `verifier` cellviews, providing a multi-level hierarchy, and allowing more flexibility to designers and the latest verification status to the project manager.	
Simulation Planning and Coverage	New
An important requirement for project sign-off is to ensure that all the design simulations in ADE Assembler are run using the efficient (or predefined) sets of operating conditions (corners, sweeps and model files) in accordance to the project. ADE Assembler and ADE Verifier now allow you to plan and create project-specific master setups that you can use to create setups for individual tests in ADE Assembler. This enables quick and efficient setup creation at the test level. Verifying the top-to-down progress of your design using ADE Verifier, you can compare the implementation histories with defined operating conditions from master setup and review the progress of your design based on Analog Coverage percentage reported by ADE Verifier. To support this feature, a new assistant, Setup Library, has been provided in ADE Assembler and ADE Verifier. For more information, see the Working with the Setup Library chapter in *Virtuoso ADE Assembler User Guide* and the Verifying the Design Against the Specified Setup chapter in *Virtuoso ADE Verifier User Guide*.	

Task Assistant – A Quick Help Functionality	New
A new task-based quick help functionality, Task Assistant, has been introduced in the Setup Library Assistant in ADE Assembler and ADE Verifier. You can access this quick help by clicking the *Setup Library Task Assistant* button available on the toolbar in Setup Library assistant. The Task Assistant will help you get started with the Setup Library assistant in a task-oriented way.	
Import and Export Flow	New
The export of requirements using CSV and Excel files now supports summary information. As a result, a *Summary Information* check box has been added to the export forms. See Exporting Requirements. The import flow for CSV and Excel files has been separated from the import of Verifier cellviews. See Importing Requirements.	
Filter Requirements and Implementations	New
The *Requirements* and *Implementations* panes now include search filters on the column headers that you can use to filter requirements or implementations based on specified search words or strings. See Filtering Requirements and Implementations.	
Mapping Requirements to Implementations	New
Verifier lets you export or import mappings between requirements and implementations using a CSV or Excel file format. As a result, a *Mapping File* sub-option has been added to *File — Export* and *File — Import*.	
Overriding Specifications	New
Verifier lets you override the specification set in an implementation with the specification set in the requirement mapped to that implementation. Similarly, you can override the requirement specification with the implementation specification. You can also override the specifications of multiple requirements and implementations.	
Toolbar	New
The new Verifier toolbar lets you access the frequently used actions using a single click.	

Assistants	New
The following new assistants have been added to the Verifier window: ■ Requirements Editor: Add, edit or view details of the selected requirement. ■ Information: View the information of the selected requirement, implementation, run or result. ■ Coverage: View the coverage summary and parameter conditions that are not covered for the requirement. ■ Setup Library: View the information associated with the current simulation run and the implementation run results.	
Forms	New
The user interface has retained the earlier theme. However, the following UI forms have been updated for usability improvements: ■ Preferences ■ Choose Verifier Cellview ■ Import ■ Import Verifier Cellview ■ Add Implementation ■ Add Implementation Cellviews ■ Change Referenced Cellview ■ Typical Value Setup ■ Sign Off ■ Set Requirement Owners ■ Managing Imported Files	

Environment Variables

Preferences	New
New environment variables have been introduced, that you can use to specify setup preferences in Verifier.	

SKILL Functions

Requirement Functions	New
New SKILL functions have been introduced to let you add, delete or setup requirements, extract requirement data, export and import requirements, manually sign-off requirements, and define custom fields.	
Verifier Session and Setup Functions	New
New SKILL functions have been introduced to let you start and exit Verifier, create batch scripts, and set session preferences.	
Implementation Functions	New
New SKILL functions have been introduced to let you add, delete or setup implementations, and extract implementation data.	
Mapping Functions	New
New SKILL functions have been introduced to let you extract, export and import mappings between requirements and implementations.	
Simulation Result Functions	New
New SKILL functions have been introduced to let you retrieve results on implementation sets and simulations.	
Report Functions	New
A new SKILL function has been introduced to let you publish HTML reports.	

Removed Features:

Features	
The following features are no longer supported in Verifier: ■ Undo and Redo ■ Search Feature ■ Monitor Mode ■ Managing References Form	
Environment Variables	
verifier.preferences	
expert	verbose
openxmlinbrowser	createlog
logfilename	createbatchtextreport
createbatchxmlreport	batchxmlreport
createbatchlogfile	batchlogfile
status	monitor
showonlymapped	showrunexternal
keepwidth	batchusereportidenticalhistory
impreadonly	imprunexplorer
palswhere	startup
verifier.triggers **Note:** The functionality provided by the following environment variables in this category is now supported by the verifRegisterCallback SKILL function.	
vstPostStart	vstPostOpen
vstPostSave	vstPostExit
vstPostStartRun	vstPostStopRun
vstPostFinishRun	vstFinishSingleRun
vstPostStartRunExternal	vstPostFinishRunExternal
vstPostChangeReqStatus	vstPostReload
vstPostVerifyChanges	vstPostCreateScript
verif_report_html_create	vstPostExport

verifier.run	
reuseAssembler	
verifier.implementation	
allowmatlabscript	defaultRunState
verifier.requirement	
idHierDelimiter	noteHierarchy
ownerCheck	
verifier.addImpCVsForm	
libDisplayFilter	cellDisplayFilter
viewDisplayFilter	
verifier.logerrorwarn	
interval	
verifier.font	
fontfamily	fontpointsize
SKILL Functions	
Starting from ICADVM18.1, some of the SKILL functions supported in the previous releases are no longer supported in Verifier. See **Removed SKILL Functions**.	

Spectre Interactive Environment

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Spectre Interactive Environment User Guide*
- *Spectre Interactive Environment Known Problems and Issues*

ICADVM18.1 Base Release

What’s New and Enhanced

Spectre Interactive Environment is a new debugging tool introduced to debug Spectre simulations interactively by using events, triggers, and breakpoints. You can run this tool in the standalone mode or from the results of a simulation run in ADE Assembler. For more details, refer to *Spectre Interactive Environment User Guide*.

Virtuoso Visualization and Analysis XL

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Visualization and Analysis XL User Guide*
- *Virtuoso Visualization and Analysis XL SKILL Reference*

ICADVM20.1 Base Release

What’s New and Enhanced

Feature

Working with Tabular Graphs	New
A new type of graph, tabular graph, is added in Virtuoso Visualization and Analysis XL. Tabular graphs offer a convenient way to analyze waveform data, especially for S-parameter data.	

ICADVM18.1 ISR13

What’s New and Enhanced

Feature

Direct Measurements Assistant	New
A new assistant, *Direct Measurements*, is added in Virtuoso Visualization and Analysis XL. This assistant contains the following tabs: *Amplitude*, *Time* and *Freq*. Using these tabs, you can quickly perform various amplitude, time-domain, and frequency-domain measurements without creating lengthy expressions. Related environment variable: `viva.graph enableDirectMeasurementAssistant`	

ICADVM18.1 ISR12

What's New and Enhanced

Feature

Digital Signal Plotting	Enhanced
The way digital signals are plotted has been enhanced: ■ More than one digital signal can now be plotted in a single strip. ■ Digital signals for the same corner are plotted in a single strip. ■ Similar to analog signals, digital signals can now be combined into a single strip. ■ Digital signals and buses are grouped together as a trace group object based on the corners. All such corner-grouped digital signals are plotted in a single strip. ■ Digital buses can be expanded or collapsed from the trace legend area. ■ The new *Configure Mnemonics* form lets you create mnemonic maps and add rules to define how different bit patterns and values of a bus are mapped with specific colors and formats. See Creating a Mnemonic Map for more information.	

Environment Variables

`viva.application confirmCloseWindow`	New
Controls whether to display a confirmation dialog box before deleting a subwindow.	
`viva.graph undoStackSize`	New
Specifies the maximum number of commands that can be undone in a graph window.	

ICADVM18.1 ISR10

SKILL Functions

mu	New
Returns the alternative stability factor that indicates the minimum distance between the origin of the unit Smith chart and the load unstable region. **Note:** This function requires four arguments.	
Mu	New
Returns the alternative stability factor that indicates the minimum distance between the origin of the unit Smith chart and the load unstable region. **Note:** This function requires a single argument.	
mu_prime	New
Returns the alternative stability factor that indicates the minimum distance between the center of the unit Smith chart and the source unstable region. **Note:** This function requires four arguments.	
Mu_prime	New
Returns the alternative stability factor that indicates the minimum distance between the center of the unit Smith chart and the source unstable region. **Note:** This function requires a single argument.	

ICADVM18.1 ISR9

What's New and Enhanced

Features

Support for PAM4 Eye Measurements	New
Earlier, you could calculate the eye measurements by using the Non-Return-to-Zero (NRZ) or PAM 2 (Pulse-Amplitude Modulation 2-Level) modulation scheme. But now, you can also choose the Pulse-Amplitude Modulation 4-Level (PAM4) scheme to calculate the eye measurements. It uses four voltage levels to represent four combinations of two-bit logic: 00, 01, 10, 11. The *Eye Measurements* tab of the Eye Diagram assistant has been enhanced to support this enhancement. A new option, Modulation Scheme, has been added to let you choose between the two modulation schemes.	
Using the Frequency Slider Bar for Circular Graphs	New
A slider bar is now available on top of every circular graph. You can use this slider bar as a frequency filter to set a range of frequencies. Data on the circular graph is filtered and selectively displayed based on the specified frequency range.	
Support for Immittance Smith Chart	New
In addition to Impedance and Admittance Smith Charts, you can now plot the Immittance Smith Charts too. To support this enhancement, a new option, *Immittance*, has been added to the *Graph Type* drop-down list on the Results Browser toolbar. In the Immittance Smith Chart, both Z Smith and Y Smith grids are plotted on the same chart. The Immittance Smith Chart has two grids: red and green. The red grid represents the Impedance or Z Smith grid, whereas, the green grid represents the Admittance or Y Smith grid. You can use the bindkey `O` to switch between the Impedance, Admittance, and Immittance Smith Charts.	

Environment Variables

`viva.depAxis labelAngle`	New

Specifies the angle through which the labels of the dependent axis are rotated.	
viva.indepAxis labelAngle	New
Specifies the angle through which the labels of the independent axis are rotated.	

ICADVM18.1 ISR7

What’s New and Enhanced

Features

Support for Reading Files with .snp Extensions	Enhanced
Virtuoso Visualization and Analysis XL now supports reading files with `.snp` extensions.	
Support for Reading RTSF Formats	Enhanced
Reading RTSF formats is now disabled by default in Virtuoso Visualization and Analysis XL. To enable it, set the shell environment variable, `SRR_ENABLE_RTSF`, to `1`. **Note:** RTSF support will be completely removed in future.	

ICADVM18.1 ISR6

What’s New and Enhanced

Features

Snapping Markers on Circular Graphs	Enhanced
Earlier, you could snap point markers and reference point markers by frequency, using the *Next POI* and *Previous POI* commands on the *Snap* toolbar. You can now use these commands from the context menus of the markers too. The Smith Reference Point Marker Properties and the Point Marker Properties forms have also been enhanced to enable you to snap the markers by frequency. A new field, *Frequency*, has been added to these forms, wherein you can specify the frequency to snap these markers on circular graphs.	

Environment Variable

`reloadSingleStyle`	New
Determines whether to apply a single style to all the traces that were added to a trace group after the graph is reloaded.	

ICADVM18.1 ISR5

What's New and Enhanced

Calculator Functions

triggeredDelay	New
Calculates the delay from the trigger point on the edge of a triggering signal to the next edge of the target signal. Related SKILL Function: triggeredDelay	
v	Enhanced
Earlier, you could use the v function to calculate the voltage of a net by specifying the *Net name* argument. You can now use this function to calculate the terminal voltages too. To support this enhancement, the following changes have been made: ■ The *Net name* argument has been renamed to the *Signal name*. You can use this argument to specify the signal whose voltage you want to calculate by selecting the signal from the schematic. ■ A new argument, *Signal Type*, has been added. You can use this argument to indicate whether the specified signal is a terminal voltage or a net.	

SKILL Functions

rfWrlsMeasContour	New
Returns the contours for the measurements of simulation results.	
rfWrlsCcdfValues	New
Plots the CCDF curve, or calculates the average or peak power from the results of an ENVLP wireless simulation.	

ICADVM18.1 ISR4

What’s New and Enhanced

Calculator Functions

eyeHeightAtXY	New
Calculates the height of an eye at the specified point (x,y) inside the eye diagram.	
eyeWidthAtXY	New
Calculates the width of an eye at the specified point (x,y) inside the eye diagram.	

SKILL Functions

eyeHeightAtXY	New
Calculates the height of an eye at the specified point (x,y) inside the eye diagram.	
eyeWidthAtXY	New
Calculates the width of an eye at the specified point (x,y) inside the eye diagram.	

ICADVM18.1 ISR3

What’s New and Enhanced

Features

Filtering Trace Legends	New
You can click the filter icon in the trace legend area to open the *Legend Filter* form. Use this form to apply filters to selectively view the traces that you want to work upon. With this form, you can save a significant space from the trace legend area to view the traces better in the active graph window or subwindow.	
Snapping Markers on Circular Graphs	New
You can now use *Frequency* as the snapping criteria (POI criteria) to snap point markers and reference point markers on circular graphs.	

ICADVM18.1 ISR2

What’s New and Enhanced

Features

Adding a Point Marker	Enhanced
While adding a point marker, you can now specify `%D` as the marker label format to show the absolute path to the results directory from where the signal is plotted. Default value `%D` is added to the defaultLabel environment variable to support this enhancement.	

ICADVM18.1 ISR1

What’s New and Enhanced

Features

Adding Custom Calculator Functions	Enhanced
Earlier, you could add a custom calculator function using SKILL (.il) files. With this enhancement, you can add a custom calculator function using SKILL++ (.ils) files too.	
Maestro Plotting Template Support for Digital Signals	Enhanced
Earlier, you could apply plotting templates only to analog signals. With this enhancement, you can now apply plotting templates to digital signals too.	

SKILL Functions

`rfJitter`	New
Calculates jitter from the result of Pnoise sample (jitter) analysis. It is used to calculate Jee, JDelay, and RMS Phase Noise.	
`rfJc`	New
Calculates cycle jitter from the result of Pnoise sample (jitter) analysis.	
`rfJcc`	New
Calculates cycle-to-cycle jitter from the result of Pnoise sample (jitter) analysis.	
`rfThresholdXing`	New
Calculates the threshold crossing value according to the jitter event time from the result of Pnoise or Hbnoise sample (jitter) analysis.	
`rfCimMcpValue`	New
Returns the main channel power (MCP) value when counter-intermodulation (CIM) is selected in LTE symbol.	
`rfWrlsCim3Value`	New
Returns the value of third-order counter-intermodulation (CIM3).	
`rfWrlsCim5Value`	New
Returns the value of fifth-order counter-intermodulation (CIM5).	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Displaying History, Test, and Design Point Information in Graphs	New
When you plot waveform results for a particular history in Virtuoso Visualization and Analysis XL, the related history, test, and design point information for the plotted waveform is displayed at the following locations: ■ Trace Legend ■ Trace Info Assistant ■ Status Bar ■ Markers	
Loading Vector Files in Virtuoso Visualization and Analysis XL	New
You can now load a VCD and a VEC file directly in Virtuoso Visualization and Analysis XL using the Select Waveform Database form. When a VEC file is loaded in the Results Browser, two folders, *digital* and *analogStimulus* are displayed. However, loading a VCD file displays only one folder, *digital*. The analog stimuli files are automatically generated when you load a VEC file, whereas for a VCD file, you need to perform an additional step to convert the digital signal to generate analog stimuli. You can also use the command-line convertor, `cdsConvertDigitalVector`, to convert a VEC or VCD file into SST2 and PSF XL database that can be loaded into Virtuoso Visualization and Analysis XL.	
Creating Plotting Templates	New
After the simulation results are plotted, you can now save a graph window as a plotting template in Virtuoso Visualization and Analysis XL. To do this, choose *File — Create Plotting Template*. You can then use the saved plotting template to plot the waveform outputs in the specified format for next simulation runs in ADE Explorer or ADE Assembler. All the formatting, markers, layout changes, and other interface changes are retained as specified in the plotting template.	
Subwindows Layout	Enhanced

The subwindows layout in Virtuoso Visualization and Analysis XL has been enhanced to provide more flexibility. You can now customize subwindows, drag and resize them, open blank subwindows for rectangular, polar, impedance, and admittance plots and then plot the signals into them.	
Virtuoso Visualization and Analysis XL Licensing Changes	Enhanced
The tool partition for environment variables, `VIVALicenseCheckoutOrder` and `VIVA_UseNextLicense`, which are used to control the licensing of Virtuoso Visualization and Analysis XL, has now been changed to `license`. However, the `viva.application` partition is still supported for them. It is recommended that you use the `license` tool partition for setting these variables in the `.cdsinit` file or CIW.	

Analog Library

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Analog Library Reference*

ICADVM20.1 Base Release

What’s New and Enhanced

Features

Support for hSpiceD View with the ibis_buffer Symbol	Enhanced
The `ibis_buffer` symbol now supports `hSpiceD` views.	

ICADVM18.1 ISR12

What’s New and Enhanced

Features

Enhanced rcwireload Symbol	Enhanced
The `rcwireload` symbol now supports the `auCdl`, `auLvs`, and `hSpice` views.	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

Redesigned vrefgnd Symbol	Enhanced
The redesigned `vrefgnd` symbol now aligns with its purpose of adding an options statement to the netlist.	

ICADVM18.1 ISR9

What’s New and Enhanced

Feature

Default Interpolation Method	Enhanced
A new interpolation method, *default*, is available for the `nport` component in the Edit Object Properties form. When you choose this method, Spectre uses the global default `auto_switch` as the interpolation method. If `nport_default_interp` is set to `auto_switch`, `nport` automatically switches to the default interpolation method for the analysis.	

ICADVM18.1 Base Release

What’s New and Enhanced

Feature

vrefgnd	New
A new global component, `vrefgnd`, has been added to the analog library. It can be applied in a design at the global level or at the subckt level as a local option.	

Virtuoso Parasitic Aware Design

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Parasitic Aware Design User Guide*
- *Virtuoso Parasitic Aware Design SKILL Reference*

ICADVM20.1 Base Release

What’s New and Enhanced

Features

Improvements in Smart View	Enhanced
The following improvements are available in a smart view: ■ QRC options are supported to help in preventing cells from a file or cells of a specific type from being included in a DSPF file. ■ The format of the default variable name assigned to multi-process corners (MPC) has been changed to `<ViewName_extractionCorner>`. ■ Resistor bounding boxes support trapezoid shapes. **Note:** These features depend on the design selection and consequent smart view extraction from Quantus QRC.	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

Enhanced Feature Support in the Smart View Flow	Enhanced
The following additional features are supported in a smart view: ■ Multi-process corner extraction (MPC flow) **Related environment variable:** `autoSyncMPC` ■ Hierarchical extraction (HRCX flow) ■ Resistor bounding boxes **Note:** These features depend on the design selection and consequent smart view extraction from Quantus QRC.	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

DSPF Files Are Saved Outside History	Enhanced
Earlier, DSPF files were copied and saved for each point in each history. Now, by default DSPF files are saved outside histories so that they can be shared by each run. **Related environment variable:** `smartViewDSPFDirectory`	
Transferring Estimates to Constraints	Enhanced
A new command, *Transfer estimates to constraints*, has been added in the Parasitics and Electrical Setup assistant toolbar to support the transfer of estimates to constraints.	

ICADVM18.1 ISR2

What’s New and Enhanced

Features

Clearing Parasitics	Enhanced
You can now clear the highlighted sets of devices and node names from the smart view by choosing the *Clear Parasitics* command in the *Smart-Parasitics* menu.	
Displaying Node Names	Enhanced
You can now toggle the display of node names on or off in a smart view by using the *Node Name* check box on *Smart-Parasitics – Display Parasitics*.	

Removed Features

Commands Removed From the Smart-Parasitics Menu	Removed
The following two commands have been removed from the *Smart-Parasitics* menu: ■ *Remove Probes* ■ *Probe Node*	

ICADVM18.1 ISR1

What's New and Enhanced

Feature

Displaying the Net Fragment Name of Parasitic Nodes	New
A new check box *Node Name* has been added to the *Smart Parasitics — Display Parasitics* form. It controls the display of the net fragment names for parasitic nodes, in the smart view. **Environment variable:** `svDisplayNodeName`	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Smart View	New
Prior to ICADVM18.1, Parasitic Aware Design supported extracted view for debugging circuits, and for post-layout verification and simulation. Starting from ICADVM18.1, the Parasitic Aware Design flow in ADE supports a new version of extracted view, called smart view, that is created using Quantus. The smart view provides the same functionality as provided by the extracted view, but has a highly efficient and scalable storage mechanism that helps in achieving a better performance in case of large and complex designs as it reduces its database and netlist size.	
Smart-Parasitics Menu	New
A new Smart-Parasitics menu has been introduced in Virtuoso Layout Suite XL and Virtuoso Layout Suite GXL to display, probe and analyze parasitics on the Smart View.	

Environment Variables

svDisplayResistance	New
Enables or disables the display of parasitic resistances in the `smart_view`.	
svDisplayCapacitance	New
Enables or disables the display of parasitic capacitances in the `smart_view`.	
svResistanceThresholdMin	New
Specifies the minimum threshold for resistance to control the display of parasitic resistances in `smart_view`.	
svResistanceThresholdMax	New
Specifies the maximum threshold for resistance to control the display of parasitic resistances in `smart_view`.	
svCapacitanceThresholdMin	New
Specifies the maximum threshold for capacitance to control the display of parasitic capacitances in `smart_view`.	

svCapacitanceThresholdMax	New
Specifies the maximum threshold for capacitance to control the display of parasitic capacitances in `smart_view`.	

Command-Line IP Selector

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

- ICADVM18.1 ISR11
- ICADVM18.1 Base Release

Related Documentation

- *Command-Line IP Selector User Guide*

ICADVM18.1 ISR11

What’s New and Enhanced

Feature

Configuring Global Options	Enhanced
A new option, *Liblist in the Verilog2001 Configuration File* has been added to the *Global Options* section on the *Settings and Options* form. You can use this field to add a new or modified list of libraries. If this field is empty, the default library list, `worklib` is used.	

ICADVM18.1 Base Release

What’s New and Enhanced

Feature

Reusing Testbenches and IP using CLIPS	New
CLIPS enables you to reuse an existing testbench/IP in other designs. It shows the design hierarchy of the specified testbench. You can now export an IP block from a remote Virtuoso config view, without the top-level SoC design access to a package directory. Multiple packages can be imported into an SoC top-level design to assemble them together. **Note:** This feature is available with XCELIUM 17.10 s007 or higher versions.	

Part 5: Physical Design

- Virtuoso Layout Viewer
- Virtuoso Layout Suite XL
- Virtuoso Layout Suite EXL
- Virtuoso Width Spacing Patterns
- Virtuoso Parameterized Cells
- Virtuoso Relative Object Design
- Virtuoso Design Rule Driven Editing
- Virtuoso Multi-Patterning Technology
- Virtuoso Abstract Generator
- Virtuoso Floorplanner
- Virtuoso Module Generator
- Virtuoso Custom Digital Placer
- Virtuoso Placer
- Voltus-Fi Custom Power Integrity Solution L
- Voltus-Fi Custom Power Integrity Solution XL
- Virtuoso Fluid Guard Ring
- Virtuoso Symbolic Placement of Devices
- Virtuoso Design Planner
- Virtuoso Concurrent Layout
- Virtuoso Photonics Solution

Virtuoso Layout Viewer

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

■ *Virtuoso Layout Viewer User Guide*

ICADVM18.1 Base Release

What’s New and Enhanced

New Virtuoso Layout Viewer	New
ICADVM18.1 introduces the Virtuoso Layout Viewer application that lets you open a layout cellview in view-only mode.	

Virtuoso Layout Suite XL

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Layout Suite XL: Basic Editing User Guide*
- *Virtuoso Layout Suite XL: Connectivity Driven Editing User Guide*
- *Virtuoso Layout Suite SKILL Reference*

ICADVM20.1 Base Release

What's New and Enhanced

Features

Creating Pins that are Schematic Aware	Enhanced
The Create Pin form now includes a *Schematic Aware* check box, which specifies whether pin creation must be schematic aware. **Environment variable:** createPinSchematicAware	

Environment Variables

`flattenExcludePcellListFile`	New
Specifies the name of the file that contains the list of Pcells that should not be flattened.	

SKILL Functions

`lxRegPreUpdateComponentsAndNetsFn`	New
Registers a pre-function hook for `lxUpdateComponentsAndNets` to allow replacing pins for which the layer and width characteristics were defined using the Design Intent tool.	
`lxRegPostUpdateComponentsAndNetsFn`	New
Registers a post-function hook for `lxUpdateComponentsAndNets` to allow replacing pins for which the layer and width characteristics were defined using the Design Intent tool.	

ICADVM18.1 ISR13

What’s New and Enhanced

SKILL Functions

leQuickAlignToggleMode	New
Allows you to toggle between quick align modes - move or stretch, and copy.	

ICADVM18.1 ISR12

What's New and Enhanced

Features

Controlling Logical Elaboration in CPH	New
Use the new *Set Stop Level* button in the CPH toolbar or the *Stop Level* field in the Update Connectivity Reference form to specify the level at which to stop the logical elaboration for a design hierarchy. **Environment variable:** cphStopLevel	

Environment Variables

`lxPositionUseInstPrB`	New
Controls whether the *Generate All From Source* and *Place As In Schematic* commands use instance master place and route boundary or instance bounding box to space instances during positioning.	

SKILL Functions

`leRegisterPPNetNameFn`	New
Registers a SKILL function that specifies the names of pseudo parallel subnets when called by the layout generation commands *Generate All From Source* and *Update Components and Nets*.	

ICADVM18.1 ISR11

What's New and Enhanced

Features

Ignoring Valid Jogs during Batch Checking	New
A new check box, *Ignore Valid Jogs*, has been introduced in the Batch Checker form to allow jogs to be ignored while running the WSP | SP Active batch checker. **Related environment variable**: `ignoreValidJogs`	
Setting a Hierarchical Force Descend in CPH	New
Use the *Set/remove hierarchical force descend* command to set a forced hierarchy traversal for the selected instances until the physical view specified in the *Physical stop library list* field is reached.	
Setting the Instance Stop Level in CPH	New
Use the new *Stop level* instance generation attribute in *CPH to* specify the hierarchy level that the selected instance must elaborate to in relation to the top level.	

Environment Variables

`renameConflictingLayoutInstances`	New
Controls whether the *Generate Selected From Source* command automatically renames existing instances in the layout with instance names that conflict with the schematic instances being transferred.	
`updateBindMfactorSplit`	New
Sets the *Split mfactored devices* option in CPH to `false` if hierarchical binding cannot take place using the specified schematic mfactor value.	
`updateBindPhysicalBinding`	New
Updates the physical bindings in CPH if the current physical binding does not match the bound layout instance.	
`hierMfactorNames`	New
Lists the names of schematic properties used to specify that hierarchical mfactors will be generated for transistors.	

useHierMfactorNames	New
Enables the hierMfactorNames environment variable.	

SKILL Functions

viaRegisterPostViaServerCallback	New
Registers the function name to be called after via server processing.	
viaUnregisterPostViaServerCallback	New
Unregisters the postViaServer procedure if it has been registered.	
leCopyTemplatesToCellView	New
Copies the row region templates and WSPDefs from the source cellview specified with the environment variable autoCopyTemplateCellViewList to the cellview specified in this function.	

ICADVM18.1 ISR10

What's New and Enhanced

Features

For use in Layout XL, Bindkeys Must be Defined in Layout XL	New
To use bindkeys in Layout XL, you must define them in Layout XL, if not already defined. Bindkeys registered earlier using the Layout Viewer are no longer supported in Layout XL.	
Interactive Folding using Folding Threshold	New
New *Use Threshold* and *Folding Threshold* options added to the Generate Folded Devices form to allow devices to be folded based on the *Folding Threshold* value.	
Create In Place During Interactive Folding	New
New *Create In Place* option added to the Generate Folded Devices form to allow device folds to be created at the same location in the layout canvas as the original device.	
Customizing Information Displayed on the Status Bar	Enhanced
You can customize the information that is displayed on the status toolbar using the environment variable, `statusToolbarFields`. You can display fields, such as cellsize, number of selected objects, environment variable values, shell environment variable values, window variable values.	

Environment Variables

`updateLayoutInstNames`	New
Controls whether the layout instance names are updated after an *Update Components And Nets* run to match their schematic counterparts.	
`upateLayoutNetNames`	New
Controls whether the layout net names are updated after an *Update Components And Nets* run to match their schematic counterparts.	
`unfoldUseLayoutWidths`	New
Unfolds a folded device to the total layout width.	
`checkMasterCompType`	New

Controls whether the cells found to have different masters are reported during a *Check Against Source* run.	
`schLayLibraryPair`	New
Specifies the schematic and layout library pairs to be used when launching Layout XL or Layout EXL to create a new library cellview or to open an existing one.	
`rulerSegmentModeType`	New
Specifies the type of ruler created by the measurement. The default value is `Single`.	
`viaIgnorePurposes`	New
Enables you to specify the list of purposes to be ignored for via creation in *Auto* and *Fast Edit* modes.	

ICADVM18.1 ISR9

What's New and Enhanced

Features

Retaining Labels on Shapes Using the Flatten Command	New
A new *Label* option on the Flatten form controls whether labels on shapes are retained when the Flatten command is run in Layout XL. The default for the related environment variable `flattenKeepLabels` has been changed from `nil` to `t`.	
Support for Must-Connect-All-Pins Pin Connectivity Model	New
You can now use the Pin Connectivity Model form to select the must-connect-all-pins model for terminals that have more than one pin. When this model is selected, routers connect to all pins of the terminal, the connectivity extractor flags all unconnected pins of the terminal, and tools such as the wire editor display flight lines to all pins of the terminal.	
Area Boundary Estimation using SKILL	New
Register a user-defined SKILL function to control the area enclosed by a top-level PR boundary or a virtual hierarchy block. See also Add Area Estimators form.	

Environment Variables

`snapRectToWSPGrid`	New
Snaps rectangles to the width spacing pattern grid.	
`checkStrictFingers`	New
Checks the number of fingers of an `nmos` or `pmos` layout device against the number of fingers in the source, and triggers the update of the associated schematic or layout devices, as appropriate.	

SKILL Function

`lxMakeDummy`	New
Creates a dummy instance of a device (instance or mosaic) by selecting one of the nets attached to the device as a reference point. The net connectivity is also created for the dummy device. For more details, see Creating Dummy Instances.	

lxUnfold	New
Deletes all the folds of the selected folded device and creates a single, unfolded device at the same coordinates in the layout as the fold of the original devices.	
leUpdateInstanceCDFParameterValues	New
Updates the list of CDF parameters name-value pairs for the specified cellview and instance.	

ICADVM18.1 ISR8

What’s New and Enhanced

Environment Variables

ignoreTermsWithNonRoutingLabels	New
Controls whether the following Layout XL commands ignore instances of cells with missing terminals that have labels on non-routing layers. ■ *Check Against Source* ■ *Update Binding* ■ *XL Compliance*	
xlComplianceFilter	New
Enables setting filters to control the display of messages issued after an *XL Compliance* run.	

ICADVM18.1 ISR7

What's New and Enhanced

Environment Variables

chainPartitionByCompType	New
Controls whether chaining partitions devices by component type and so only attempts to abut devices with the same component type. The default is `nil`.	
chainPartitionByNumFins	New
Controls whether chaining partitions devices by number of fins and so only attempts to abut devices with the same number of fins. The default is `nil`.	
displayConfigureTab	New
Controls the display of the *Configure* tab in the Connectivity form. The default is that the tab is displayed (`t`).	

ICADVM18.1 ISR6

What's New and Enhanced

Feature

Selecting Shapes Based on Nets	New
You can use the net-based selection feature to select shapes based on their nets.	

Environment Variable

saveViewSubType	New
Automatically opens a Layout cellview in the same application in which it was last saved (`t`). The default is `t`.	

ICADVM18.1 ISR5

What’s New and Enhanced

Environment Variable

flattenKeepLabels	New
Controls whether labels on shapes are retained when the *Flatten* command is run in Layout XL. The default is `nil`. For more information about Flattening instances in Layout XL, see Flatten.	

ICADVM18.1 ISR4

What's New and Enhanced

Feature

Folding a Multi-fingered Device to Create Split Fingers	New
You can now use the Generate Folded Devices form to split a multi-fingered device or transistor after it has been generated in layout by folding the multi-fingered device in order to create a separate device for each finger. You can specify how the width of each fold is calculated using the environment variable foldFingerSplitUseSchWidth. Folding creates a new device for each fold by setting the Fingers parameter to 1. If set to `t`, the width of the schematic instance is divided by the number of fingers and the calculated number is applied to the width of each new device created. If set to `nil`, the width of each new device relies on the fingers callback of the PDK to set the correct width. The default is `nil`.	

ICADVM18.1 ISR2

What's New and Enhanced

Feature

Pins below boundary	New
When using the Make Cell command you can choose to position the created pins outside the design's boundary using the environment variable `makeCellPinsBelowBoundary` or the *Pins below boundary* field in the Make Cell form.	

Environment Variable

`makeCellPinsBelowBoundary`	New
Specifies whether the pins of the made cell, created using the Make Cell command, are placed outside the design boundary (t).	

SKILL function

`leMakeCell`	Enhanced
A new argument `?pinsBelowBoundary` *`g_pinsBelowBoundary`* has been added which specifies whether the pins of the made cell are placed outside the design boundary.	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Tracing Nets	New
The Net Tracer is enhanced to support the following new features: ■ Logical trace creation. ■ Step trace creation for physical traces. ■ *Edit In Place* support for the selected trace. ■ New net search feature and enhanced trace display in the Trace Manager.	
Excluding Blockages during Batch Checking	New
You can now exclude blockages while running the WSP Active batch checker.	
Layout XL Options Form Renamed, New Tab Added	New
The erstwhile Layout XL Options form that could be accessed using the *Options – Layout XL* command has now been renamed to the Connectivity form. ■ To access the Connectivity form, use *Options – Connectivity*. ■ The Connectivity form now supports a new Configure tab that displays the values of the environment variables associated with the Preset options.	
Canvas Glyph Support for CAS Tab Markers in Annotation Browser	New
To make the CAS tab markers more prominent, the markers now display unique icons on the canvas that can be pinned to the canvas, moved around, and deleted. In addition, the information balloon associated with each marker is now more informative as it highlights the mismatches between the current and expected values.	
Remove lxStickyNet Property on Non-floating Shapes	Enhanced
Default changed to `t`.	
Dynamic Selection Assistant Enhancements for Designs with Color Data	Enhanced

For designs with color data, in addition to *LPP* and *Data*, *Color*, *State*, and *Hierarchy Depth* columns are displayed in the Dynamic Selection assistant.	
On-Canvas Editing of Modgen Rows and Columns	Enhanced
You can now edit rows and columns of a modgen using the Property Editor assistant.	
Saving Terminal Name of Flattened Pins	Enhanced
You can now preserve the terminal name of flattened pins using the *Term name* field in the Flatten form.	

Environment Variables

`casVisibleUnboundMarkers`	New
Displays unbound markers on the canvas.	
`lxDummyBackAnnotateMFactorName`	New
Allows specifying an override to the parameter name otherwise used for dummy backannotation by the `lxDummyBackAnnotateMFactor` environment variable.	

SKILL Functions

`cphGetTopCellName`	New
Returns the cell name of the top schematic cellview in the specified physical configuration view.	
`cphGetTopLibName`	New
Returns the library name of the top schematic cellview in the specified physical configuration view.	
`cphGetTopViewName`	New
Returns the view name of the top schematic cellview in the specified physical configuration view.	
`leCreateNetField`	New
Creates cyclic field for showing net names of the current cellview.	
`leHiIncrementalViolation`	New
Specifies the incremental command mode.	
`leHiIncrementalViolationUpdater`	New

Enables you to iterate between the violations of the command.	
lntGetCurrentStep	New
Returns the current step value when the Net Tracer is invoked in Step trace mode.	
lntGetTailVal	New
Returns the current value of the tail when the Net Tracer is invoked in Step trace mode.	
lntNextStep	New
Increments the current step value for a trace by one.	
lntNextToSetStep	New
Forwards the tracing of all Step mode traces to the newly specified step value.	
lntPrevStep	New
Decrements the current step value for a trace by one.	
lntSetCurrentStep	New
Sets the current step value for a trace when the Net Tracer is invoked in Step trace mode.	
lntSetTailVal	New
Sets the current tail value for all the traces when the Net Tracer is invoked in Step trace mode.	
lxHiAdjustAreaBoundary	New
Opens the Adjust Area Boundary form, which can be used to modify the area boundary of the selected virtual hierarchy.	
lxHiCreateVirtGroup	New
Creates a group with an area boundary around the selected instances that can be edited to allow automatic placement inside the group.	
lxHiDetach	New
Opens the Detach form, which can be used to select a virtual hierarchy from the original, top-cell layout to create a new layout cellview.	
lxHiEmbed	New
Opens the Embed form, which can be used to integrate a layout hierarchy that was realized outside of the design.	
lxHiRemaster	New
Opens the Remaster form, which can be used to replace the selected virtual hierarchy with an existing layout variant.	
lxHiVirtHierOptions	New

Opens the Virtual Hierarchy Options form, which can be used to specify whether the layout displays the name of the virtual hierarchy, the cell, or both. In addition, the form can be used to enable or disable symbol overlay and to specify whether the virtual hierarchy can be auto placed when the area boundary is modified.

Virtuoso Layout Suite EXL

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Layout Suite EXL Reference*

ICADVM18.1 ISR10

What’s New and Enhanced

Features

For use in Layout EXL, Bindkeys Must be Defined in Layout EXL	New
To use bindkeys in Layout EXL, you must define them in Layout EXL, if not already defined. Bindkeys registered earlier using the Layout Viewer are no longer supported in Layout EXL.	

ICADVM18.1 Base Release

What’s New and Enhanced

New Virtuoso Layout Suite EXL Product	New
ICADVM18.1 introduces the brand new Virtuoso Layout Suite EXL design cockpit, providing access to the industry’s first electrically and simulation-driven layout design environment. Layout EXL offers full access to all the functionality provided under Layout EAD in previous releases and to the newly introduced concurrent layout team design and interactive simulation driven routing features. It is also the required base platform for all 5nm design; for the Virtuoso RF solution; and for a new set of in-design technologies to facilitate advanced design planning and congestion analysis.	

Virtuoso Width Spacing Patterns

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Width Spacing Patterns User Guide*
- *Virtuoso Layout Viewer User Guide*
- *Virtuoso Layout Suite XL: Basic Editing User Guide*
- *Virtuoso Layout Suite XL: Connectivity Driven Editing Guide*
- *Virtuoso Technology Data SKILL Reference*
- *Virtuoso Design Environment SKILL Reference*
- *Virtuoso Layout Suite SKILL Reference*
- *Virtuoso Technology Data ASCII Files Reference*
- *Virtuoso Technology Data Constraints Reference*

ICADVM18.1 ISR13

What’s New and Enhanced

Feature

New Filter in theTrack Pattern Assistant	Enhanced
You can use the new *Global Active* filter in the *Track Pattern* assistant to set active wire types for all WSSPDefs in the cellview.	

SKILL Function

tpaSetFilterByName	New
Sets the related snap pattern (RSP) filter by name or by other filters specified in the *Track Pattern* assistant.	

ICADVM18.1 ISR8

What’s New and Enhanced

Feature

Track Pattern Task Assistant - A Quick Help	New
A new task-based quick help functionality, Track Pattern Task Assistant, has been introduced for the Track Pattern assistant. You can access this quick help by clicking the question mark button available in the Track Pattern assistant. The Task Assistant provides task-oriented information to get you started with the Track Pattern assistant.	

SKILL Function

wspGetLineEndGrids	New
Returns a list of line-end grid names for the specified layer.	

ICADVM18.1 ISR7

What’s New and Enhanced

Feature

Enhanced Track Pattern Assistant	Enhanced
You can use the *EIP Auto Pushdown Options* icon on the toolbar in the *Track Pattern* assistant to have pins in a subcell snap to the top level WSP tracks.	

ICADVM18.1 ISR5

What’s New and Enhanced

Feature

Enhanced WSP Region Snapping	Enhanced
You now have the flexibility to derive the WSP region snapping values from the WSP pitch, a predefined value, user units, the manufacturing grid, or window snap values. These options are available on the Create Region form.	

ICADVM18.1 ISR1

What’s New and Enhanced

Feature

Improved Track Pattern Assistant GUI	Enhanced
The user interface of the *Track Pattern* assistant has been enhanced. The key usability improvements are: ■ The commonly used commands are now available on the toolbar in the top section of the *Track Pattern* assistant. This allows more WSP commands, such as *WSP Manager* and *WSP Active Checker*, to be available, by default. ■ You can collapse or expand the *Filter* section. This lets you view more of the WSSPDef state information in the *Tracks* table. ■ Similar to Palette, the *Tracks* table toolbar now has buttons for setting all, none, active, and visible WSP tracks. ■ A new column, *A*, has been added to the *Tracks* table to identify the globally active WSP tracks.	

ICADVM18.1 Base Release

What’s New and Enhanced

Feature

Enhancements in the Tracks table of the WSP Manager	Enhanced
You can now view the display packet name for the WSP track in the *Tracks* table of the WSP Manager.	

SKILL Functions

wspMetalFillTrim	New
Fills metal shapes based on WSP tracks and inserts trim shapes as needed.	
wspDeleteMetalFill	New
Deletes the shapes created by the `wspMetalFillTrim()` function.	

Virtuoso Parameterized Cells

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Parameterized Cell Reference*
- *Virtuoso Parameterized Cell SKILL Reference*

ICADVM18.1 ISR9

What’s New and Enhanced

Promoting the Express Pcell feature	New
When you close a layout design that contains a high proportion of Pcell submasters, you now see a non-blocking pop-up message recommending that you use the Express Pcell feature to reduce the time taken to open such designs.	

ICADVM18.1 ISR4

What’s New and Enhanced

Maintaining the PDK Version in the Express Pcells Cache	Enhanced
The Express Pcells cache is now PDK version-aware and will notify you when the Pcells currently in the cache do not match those in the latest version of the PDK. This lets you keep the cache current whenever the Pcells in a PDK are updated during a particular design cycle. The changes are also fully backward compatible, allowing you to continue to use older cache content where required. For more information, including instructions on how to enable the feature for CAD engineers and PDK developers, follow the link above to the *Virtuoso Parameterized Cell Reference*.	
Enabling Express Pcells for Specific Libraries	Enhanced
The Express Pcells mechanism now lets you specify which libraries are included/excluded for cache operations. This allows you to disable cache operations for dynamic reference libraries containing Pcells that are edited and optimized regularly but for which supermasters are not recompiled.	

Virtuoso Relative Object Design

No new features or enhancements have been introduced in Virtuoso Relative Object Design in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Relative Object Design User Guide*
- *Virtuoso Relative Object Design SKILL Reference*

Virtuoso Design Rule Driven Editing

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Design Rule Driven Editing User Guide*
- *Virtuoso Layout Suite SKILL Reference*
- *Virtuoso Technology Data ASCII Files Reference*

ICADVM20.1 Base Release

What's New and Enhanced

Constraints

minEndOfLineToNotchSpacing	New
(ICADVM20.1 Only – 95512) Specifies the minimum spacing between an end-of-line edge and a notch whose length is less than the specified notch length.	
requiredEndOfLineShape	New
(ICADVM20.1 Only – 95511 and 95800) Specifies the spacing at which at least one end-to-end shape must be present on at least one end of a rectangular shape.	
allowedCutClass	Enhanced
(ICADVM20.1 Only – 95511) Support added for the following parameters: ■ minVoltage: The constraint applies only if the voltage on the shape is greater than or equal to this value. ■ maxVoltage: The constraint applies only if the voltage on the shape is less than this value.	
endOfLineKeepout	Enhanced
(ICADVM20.1 Only – 95511 and 95800) Support added for the following parameters: ■ eolSpacingRange: The constraint applies when the spacing of the line-end edge of the triggering EOL is in the range specified by the parameter. ■ otherEolSpacingRange: The constraint applies when a line-end edge of a neighboring shape intersects or is within the keepout region and the spacing of this line-end edge is in the range specified by otherEolSpacingRange. ■ jointAsEol: The two edges at the joint of two intersecting shapes are also treated as end of line. ■ checkOtherWidth: The constraint applies when the width of the other shape is also less than the width parameter. ■ equalRectWidth: The constraint applies only if the length of the end-of-line edge is equal to the width of the shape.	

minCornerToCornerDistance	Enhanced
(ICADVM20.1 Only – 95511 and 95800) The following new parameters have been added: ■ `extensionLength`: Specifies the length of the extension that needs to be created from the internal concave corners. ■ `oppositeDirection`: The constraint applies only when the extensions are in opposite directions.	
minEndOfLineEdgeExtension	Enhanced
(ICADVM20.1 Only – 95511 and 95800) Support added for the following parameters: ■ `minVoltage`: The constraint applies only if the voltage on the shape is greater than or equal to this value. ■ `maxVoltage`: The constraint applies only if the voltage on the shape is less than this value.	
minOppExtension	Enhanced
(ICADVM20.1 Only – 95511 and 95800) Support added for the following set of parameters: `widthHorizontal | widthVertical`. Only widths of metal shapes in the specified direction are considered for the corresponding width-based extension lookup.	
minPRBoundaryInteriorHalo	Enhanced
(ICADVM20.1 Only – 95511 and 95800) Support added for the following related parameters: `exceptLayer` and `exceptWidth`. The constraint applies only to shapes that overlap (fully or partially) a shape on the *`tx_exceptLayer`* layer whose width is less than or equal to *`f_exceptWidth`*.	
minQuadrupleExtension	Enhanced
(ICADVM20.1 Only – 95511 and 95800) Support added for the following parameter: `lineEndGapOnlyTable`. The constraint value corresponding to this table entry applies only if the `lineEndGap` constraint condition on this layer is met.	
minSideSpacing (One layer)	Enhanced
(ICADVM20.1 Only – 95511 and 95800) Support added for the following parameters: ■ `otherLayer`: The constraint applies to the specified cut layer that is inside the metal layer. ■ `cutClass`: The constraint applies only when *`tx_otherLayer`* is of the specified cut class size. ■ `cutWithin`: The constraint applies only if *`tx_otherLayer`* is within the specified distance.	

minStepEdgeLength	Enhanced
(ICADVM20.1 Only – 95511 and 95800) Support added for the following parameters: ■ adjEolSpacingRange: The constraint applies to an edge if any one of its adjacent edges is an end-of-line edge meeting the width parameter and this adjacent edge has another perfectly aligned end-to end-shape at a spacing specified by this parameter. ■ jogEdge: The constraint applies only to a jog edge (Z shape). ■ adjJogEdge: The constraint applies only to an edge adjacent to a jog edge. ■ adjNotchEdge: The constraint applies only to an edge adjacent to a notch edge.	
minViaSpacing (One layer)	Enhanced
(ICADVM20.1 Only – 95511 and 95800) Support added for the following parameters: ■ exceptGap: The constraint does not apply if there is a gap of exactly this value between the via cuts that overlap the shape on the overLayer layer. ■ exceptSameMetalOverlap: The constraint does not apply if the via cuts that overlap the shape on the overLayer layer are on the same metal.	

Environment Variables

drdEditSmartSnapAperture	New
Sets the aperture size to be used to snap the shape to the nearest valid width.	
drdValidLayersGrayCheck	New
Indicates that only the colorable layers specified in the validLayers constraint are checked for uncolored and unlocked shapes.	

ICADVM18.1 ISR13

What’s New and Enhanced

Constraints

preColoredLayers	New
(ICADVM18.1 Only – 95511) Lists precolored layers or layer-purpose pairs in the `VirtuosoMPTSetup` constraint group.	

ICADVM18.1 ISR12

What's New and Enhanced

Constraints

antennaRatio	New
Defines antenna ratios on a per-layer basis.	
cumPerLayerAntennaRatio	New
Defines cumulative antenna ratios on a per-layer basis.	
allowedSpacingRanges (One layer)	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following parameter: `exceptWidthEdge`. The constraint does not apply to any edges of a square shape or to edges whose span is considered as the width of the shape.	
maxViaArrayClusterSize	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following new parameter: `checkLengthOfAnyArray`. The constraint applies if the length of any one of the via arrays is within *`g_lengthRange`*.	
minStepEdgeLength	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following parameter: `width`. The constraint applies only on shapes with width equal to the given value.	
mergedViaCornerToCornerSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) ■ Support added for the following parameters: ❑ `lengthRange`: The constraint applies if the length of the first shape, which might be merged or unmerged, in the merge direction is in this range. ❑ `otherLengthRange`: The constraint applies if the length of the other shape, which might be merged or unmerged, is in this range. It depends on `lengthRange`. ■ The following parameter is now mandatory: `mergeSpacing`.	
minEndOfLinePerpSpacing	Enhanced

(ICADVM18.1 Only – 95511) Support added for the following parameter: `negativePRL`. The constraint applies only if the perpendicular edge has a parallel run length of less than zero with the end-of-line edge.	
`minNotchSpacing`	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following set of parameters: `horizontal | vertical`. These indicate the direction in which the constraint is applied.	
`minViaSpacing (One layer)`	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following parameters: ■ `prlRange`: The constraint applies only when the parallel run length between two shapes is in this range. ■ `viasOnSameNet | viasOnSameMetal | viasOnSameVia`: The constraint applies to vias of the specified connectivity type. ■ `exceptViasOnSameNet | exceptViasOnSameMetal | exceptViasOnSameVia`: The constraint does not apply to vias of the specified connectivity type.	
`minViaSpacing (Two layers)`	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following parameter: `prlRange`. The constraint applies only when the parallel run length between two shapes is in this range.	
`minViaGroupSpacing`, `viaGroup`	Enhanced
(ICADVM18.1 Only – 95511) The following parameter is now optional: `cutClass`.	

Environment Variables

`drdBatchReportCoveredNonPcellGrayShapes`	New
(ICADVM18.1 Only) Enables you to specify whether or not to report uncolored shapes inside non-Pcell instances in Batch mode even if these are covered with colored and locked shapes.	

ICADVM18.1 ISR11

What's New and Enhanced

Constraints

allowedSpanLengthRanges	New
Specifies all allowable span lengths on a routing layer.	
endOfLineKeepout	Enhanced
(ICADVM18.1 Only – 95511 and 95800) Support added for the following set of parameters: `horizontal | vertical`. These indicate the direction in which the constraint is applied to ends-of-lines.	
mergedViaCornerToCornerSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) Support added for the following parameters: ■ `tripletCenterViaSpacingX` and `tripletCenterViaSpacingY`: Three cuts at any of these values in the X and Y directions are exempted. ■ `tripletCenterViaLarger`: The area of the center via must be more than or equal to other vias for the diagonal cluster of three vias to be exempted.	
minClusterSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511 and 95800) Support added for the following parameters: ■ `enclosingLayer`: The constraint applies only for the specified group direction for shapes enclosed by this layer. ■ `viaEdgeType`: The constraint applies only when all via edges in the direction orthogonal to the group direction and in the enclosing layer are of this type.	

minViaSpacing (One layer)	Enhanced
Support added for the following sets of parameters. The presence or absence of one or more layers and related purposes determines whether the constraint applies. ■ insideLayers and insidePurposes ■ outsideLayers and outsidePurposes (ICADVM18.1 Only – 95512) Support added for the following parameters: ■ exceptSameNet: The constraint does not apply to via cuts on the same net. ■ exceptSameMetal: The constraint does not apply to via cuts enclosed by a contiguous same-metal shape. (ICADVM18.1 Only – 95511 and 95800) Support added for the following parameter: deltaVoltage. The constraint applies if both the cut shape and the metal shape have voltages and the delta voltage between the *layer1* shape and the *layer2* shape is greater than this value.	
minViaSpacing (Two layers)	Enhanced
(ICADVM18.1 Only – 95511 and 95800) The following new parameter has been added: deltaVoltage. The constraint applies if both the cut shape and the metal shape have voltages and the delta voltage between the *layer1* shape and the *layer2* shape is greater than this value.	

ICADVM18.1 ISR10

What’s New and Enhanced

Constraints

minViaSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511 and 95800) Support added for the following parameters: ■ `otherCutClass`: The constraint applies if one cut class is of the type `cutClass` and the other cut class is of the type `otherCutClass`. ■ `otherViaEdgeType`: The constraint applies between via edges where one via edge is of the type `viaEdgeType` and the other via edge is of the type `otherViaEdgeType`.	

ICADVM18.1 ISR9

What's New and Enhanced

Constraints

overlapViaGroup	New
(ICADVM18.1 Only – 95511 and 95800) Specifies the maximum number of cuts in a via group, where the spacing between cuts is within a specified range.	
minParallelWithinViaSpacing	Enhanced
(ICADVM18.1 Only – 95511 and 95800) Support added for the following parameter: `otherWidth`. The constraint applies if the metal shape (of the layer above the cut layer) is within the common projected parallel run length of cut shapes and has width less than this value.	
minSideSpacing (Two layers)	Enhanced
Support added for the following parameters: ■ `deltaVoltage`: (ICADVM18.1 Only – 95511) The constraint applies if both the cut shape and the metal shape have voltages and the delta voltage between the *`layer1`* shape and the *`layer2`* shape is greater than this value. ■ `prlRange`: (ICADVM18.1 Only – 95511) The constraint applies only if the parallel run length between the two sides falls in this range.	
minViaSpacing (One layer)	Enhanced
(ICADVM18.1 Only–95511) Support added for the following set of parameters that indicate the sides between which the spacing is measured: `shortSideToShortSide` | `shortSideToLongSide` | `shortSideToAnySide` | `longSideToLongSide` | `longSideToAnySide` | `anySideToAnySide`	
minViaSpacing (Two layers)	Enhanced
Support added for the following sets of parameters: ■ `insideLayers` and `insidePurposes` ■ `outsideLayers` and `outsidePurposes` The presence or absence of one or more layers and related purposes determines if the constraint applies.	

viaGroup	Enhanced
Support added for the following parameters: ■ `allowBend`: (ICADVM18.1 Only – 95511 and 95800) When *`g_numCuts`* is `3` and this parameter is specified, the constraint applies to a group of three via cuts that form a bend or notch pattern. ■ `sameMask`: (ICADVM18.1 Only – 95511) The constraint applies only to via cuts on the same mask.	

ICADVM18.1 ISR8

What's New and Enhanced

Constraints

minTouchingDirSpacing	New
Specifies the minimum spacing between *layer1* and *layer2* shapes in the direction perpendicular to the *layer1* edge that is abutted or overlapped by a shape on *layer3*. In Notify and Enforce modes, DRD does not check the minimum spacing between the shapes if *layer3* is moved.	
minCutRoutingSpacing (Two layers)	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following parameter: deltaVoltage. This parameter specifies that the constraint applies only if both the cut shape and the metal shape have voltages, and the delta voltage between *layer1* (the cut shape) and *layer2* (the metal shape) is greater than this value.	
minViaSpacing (Two layers)	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following parameters: spacing direction [horizontal | vertical], prl *f_prl*, and oppositeSides. ■ spacing direction [horizontal | vertical]: The direction in which spacing is measured can be specified as horizontal or vertical. ■ prl *f_prl*: When the spacing direction is specified, *f_prl* can also be specified. In this case, the constraint applies only if the parallel run length is equal to or greater than the f_prl value, which is measured orthogonally to the spacing direction. ■ oppositeSides: When *f_prl* is specified, the oppositeSides parameter can also be specified. In this case, the constraint applies only if the *layer1* shape is surrounded by the *layer2* shape on opposite sides.	

ICADVM18.1 ISR7

What’s New and Enhanced

Constraint

minSpanLengthSpacing	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following parameter: `diffMask`. This parameter specifies that the constraint applies only between shapes of different mask colors.	

ICADVM18.1 ISR4

What’s New and Enhanced

Constraints

sameMetalAlignedCuts	New
(ICADVM18.1 Only – 95511) Specifies the maximum number of centered cuts if the cuts share the same metal above or below the metal layer.	
minVoltageSpacing (One layer)	Enhanced
(ICADVM18.1 Only – 95511) Support added for the following set of parameters: `horizontal | vertical`. The direction in which spacing is measured. If direction is not specified, spacing is measured in any direction.	

ICADVM18.1 ISR3

What's New and Enhanced

Feature

Enhanced UI for Ignoring Marked Instances	Enhanced
The *DRD Add Ignore* command in the Instance context menu has been renamed to *DRD Ignore*, and a submenu has been added to improve the usability of this feature that is used to exclude marked instances from checking. For more information, see Excluding Instances from Checking.	

Constraints

minIsolatedArea	New
Specifies the minimum area required for shapes when no other shape is in a defined region.	
mergedViaCornerToCornerSpacing	Enhanced
(ICADVM18.1 Only–95511) Support added for the following parameters: ■ horizontalSpacingRange: The constraint applies only between shapes in this spacing range in the horizontal direction. ■ verticalSpacingRange: The constraint applies only between shapes in this spacing range in the vertical direction. ■ isolatedPairOtherSpacing: The constraint applies only if other shapes have at least the specified spacing from both the shapes.	
minEndOfLineAdjacentToStep	Enhanced
(ICADVM18.1 Only–95512) Support added for the following parameter: adjacentLength. The constraint applies only if the other adjacent edge of the minimum step edge has length greater than or equal to this value.	
minQuadrupleExtension	Enhanced

Support added for the following set of parameters: `horizontal | vertical`. The first extension pair is measured in the given direction; the second extension pair is measured in the perpendicular direction. If direction is not specified, the extension pairs are measured in any direction.	
`minVoltageExtension`	Enhanced
(ICADVM18.1 Only-95511) Support added for the following set of parameters: `horizontal | vertical`. The constraint applies only to the extensions measured in this direction. If direction is not specified, the extension is measured in any direction.	
`trimMinSpacing (One layer)`	Enhanced
(ICADVM18.1 Only–95511) Support added for the following set of parameters: `exactAligned` and `exceptDiffMaskAligned`. The end-to-end spacing must be greater than or equal to this value if the shapes are aligned. If *`prl`* is not specified, end-to-end spacing is applied in Euclidian measurement. If *`prl`* is specified, end-to-end spacing is applied in Manhattan measurement. If `sameMask` is also specified, `exceptDiffMaskAligned` can be used to indicate that the constraint does not apply to two aligned trim shapes on the same mask when there is another exactly aligned trim shape on a different mask between them.	
`trimShape`	Enhanced
(ICADVM18.1 Only–95511) Support added for the following parameter: `keepoutRegion`. It is applicable only when `extension` is used. The *`f_offset1`* value specifies the keepout offset in the metal direction. The *`f_offset2`* value specifies the keepout offset in the direction perpendicular to the metal layer.	

ICADVM18.1 ISR2

What's New and Enhanced

Features

DRD Configuration File	New
The DRD Options form offers the option to save settings from the form into a configuration file and load settings from the file into the form.	
Support Added to Consider Unlocked Shapes as Gray	New
(ICADVM18.1 Only–95511) DRD provides the *Consider Unlocked Shapes as Gray* option while performing color-related checks. See Performing Color Checks. You can also use these related environment variables to enable the option: ■ Interactive: `drdConsiderUnlockedShapesAsGray` ■ Batch: `drdBatchConsiderUnlockedShapesAsGray`	

Constraints

`maxViaArrayClusterSize`	Enhanced
(ICADVM18.1 Only–95511) Support added for the following parameters: ■ `neighborDistanceRange`: Neighbors are considered within this range in a direction orthogonal to the cluster direction. ■ `maxFacingEdgeNeighbors`: Indicates the maximum facing edge neighbors within the specified neighborDistanceRange for a cluster. ■ `checkEachSideNeighborsSeparately`: Indicates that neighbors on each side must be checked.	

SKILL Functions

`drdOptionsSet`	New
Sets the default states for fields in the configuration file for the DRD Options form.	
`drdOptionUpdateConstraint`	New

Sets the default states for specified constraints and modes in the configuration file for the Filter tab of the DRD Options form.	
drdOptionUpdateLayer	New
Sets the default states for specified layers in the configuration file for the Filter tab of the DRD Options form.	

ICADVM18.1 ISR1

What’s New and Enhanced

Constraints

mergedViaCornerToCornerSpacing	New
(ICADVM18.1 Only–95511) Defines a corner-to-corner spacing between two merged vias with negative parallel run length.	
minVoltageExtension	New
(ICADVM18.1 Only-95511) Specifies the minimum extension of a *layer1* shape past a *layer2* shape when the voltage on *layer1* is greater than or equal to the specified value.	
minViaSpacing (One layer), minViaSpacing (Two layers), minSpanLengthSpacing, minSideSpacing (One layer)	Enhanced
Support added for the Enforce mode.	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Consolidated DRD Interface	New
The new DRD toolbar offers access to the simplified and consolidated interface in just one click. This new interface houses the interactive and batch checking options in one place instead of two separate forms. For more information, see **Setting Up DRD Options** and **DRD Options**.	
Improved User Experience	New
The following features have been introduced to provide the user an improved experience during DRD editing: ■ **Constraint and Layer Filtering:** The new *Filters* tab of the DRD Options form lets you choose the constraints and layers for verifying a design. The filter feature gives you more flexibility at both the category level and individual constraint and layer levels. For more information, see Selecting Constraints for Verification and Selecting Layers for Verification. ■ **Incremental Display of Violations:** You can now see and analyze the DRC violations incrementally using the `Ctrl+Shift+m` bindkey. For details, see Incremental Violation Display. ■ **Violation Display Arrow:** The arrows, which are used to indicate the optimum spacing between objects, are now proportional to the size of the target shape. They are placed at the zoomed in portion of the shape. ■ **DRD Targets:** The DRD Targets options are now available in the context menu of the selected shape. ■ **Halo Display:** You can now control the display of halos generated during DRD editing using the `drdEditAutoTurnOffHalo` environment variable or the *Auto Halo Turn Off* field in the DRD Options form.	
Using the Deferred Post Edit Mode	New
You can now see the violation markers for a number of create or edit operations in one go.	
Using the Relaxed Enforce Mode	New

The Relaxed Enforce mode is a relaxed version of the Enforce mode that lets the pointer to move into the violation region.	
Verifying Nets	New
You can now check for DRC violations on a net or a set of nets.	

Constraints

minRectArea	Enhanced
(ICADVM18.1 Only-95511) Support added for the following parameters: layer: You can check if the rule applies to a shape only if one or both its ends overlap or touch shapes on this layer. overlapType: The rule applies only if the number of overlaps is equal to this value.	
minViaGroupSpacing	New
(ICADVM18.1 Only-95511) Specifies the spacing of a cut in a via group to a cut of a given cut class.	
viaGroup	New
(ICADVM18.1 Only-95511) Specifies the maximum number of cuts in a via group when the parallel run length is within the prlRangeOne range on one side and the prlRangeTwo range on the other side and the spacing is in the dualDistance range.	

Environment Variables

drdBatchColorability	New
Specifies whether to perform colorability check on the design in Batch mode.	
drdBatchColorShorts	New
Specifies whether to check the design for color shorts violations in Batch mode.	
drdBatchConsiderGrayShapes	New
Specifies whether you want gray shapes to be considered as colored in Batch mode.	
drdBatchScopeAreaLimit	New
Sets the area of check in DRD Batch mode.	
drdBatchScopeSpecifiedArea	New
Specifies the area of the design on which you want to run DRD in Batch mode.	
drdBatchUncoloredShapes	New
Specifies whether to check the design for uncolored shape violations in Batch mode.	
drdBatchUnlocked	New
Specifies whether to check the design for color-unlocked shape violations in Batch mode.	

drdEditAutoTurnOffHalo	New
Specifies whether to suppress the display of halos if the number of violations increases beyond a threshold.	
drdEditAutoTurnOffHaloLimit	New
Sets the threshold halo limit after which DRD automatically suppress the display of halos.	
drdEditBatchColorRules	New
Sets the MPT checks options in DRD Batch mode with which you can check a design for conformance to the color rules defined in the technology file.	
drdEditBatchHierDepth	New
Sets the number of hierarchy levels to be checked for violations in Batch mode.	
drdEditPaletteLayers	New
Specifies whether you want DRD to monitor only those layers for violations that are defined in the *Layers* panel of the *Palette* assistant.	

SKILL Function

drdVerifyObjects	Removed
The `drdVerifyObjects` function is now obsolete. To verify objects, use the `drdVerifySelSet` function instead.	

Virtuoso Multi-Patterning Technology

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Multi-Patterning Technology User Guide*
- *Virtuoso Layout Viewer User Guide*
- *Virtuoso Layout Suite XL: Basic Editing User Guide*
- *Virtuoso Layout Suite XL: Connectivity Driven Editing Guide*
- *Virtuoso Layout Suite EXL Reference*
- *Virtuoso Technology Data SKILL Reference*
- *Virtuoso Design Environment SKILL Reference*
- *Virtuoso Layout Suite SKILL Reference*

ICADVM18.1 ISR13

What’s New and Enhanced

SKILL Function

mptGetColorShiftingLayers	New
Returns a list of valid shifting layers set by the environment variable, colorShiftingLayers.	

Environment Variables

colorShiftingLayers	New
Specifies the layers that are color shiftable.	

ICADVM18.1 ISR10

What’s New and Enhanced

Environment Variables

updateColorOnActivate	New
Determines whether coloring will be updated when the color engine is turned on and the design contains outdated coloring data. The default is `never`.	

ICADVM18.1 ISR9

What’s New and Enhanced

SKILL Function

mptColorRemastering	New
Converts color-shifted abstract views to corresponding layout views or conversely.	
mptGetTrackPatternPattern	New
Returns a list of the color patterns for the specified track pattern.	
mptSetTrackPatternPattern	New
Specifies the color pattern for the track pattern.	

ICADVM18.1 ISR4

What’s New and Enhanced

Feature

Deleting Color on Shapes Connected to Selected Shape	Enhanced
You can delete the color on shapes connected to the selected shape using the *Delete color on connected shapes* option on the Multiple Patterning Options form.	

Environment Variable

deleteConnectedShapes	New
Specifies whether the color on the shapes connected to the selected shape must be deleted.	

ICADVM18.1 ISR1

What’s New and Enhanced

Feature

Generating Temporary Markers to Highlight Violating Shapes	Enhanced
The last operation status feature on the Multiple Patterning toolbar has been enhanced to display temporary markers on the violating shapes.	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Checking the Design	New
You can use the *Checks* option on the Multiple Patterning toolbar to perform color checks in the design. It has the *Hierarchical Color Locking Check*, *Violation Checks*, and *Methodology Compliance* options to let you check for potential color conflicts in the design.	
Pre-defined Setup Driven MPT Flows	New
You can now easily setup the MPT engine with pre-defined MPT flows. You can use SKILL functions, `mptGetFlowSettings`, `mptGetFlowNames`, `mptSetFlow`, `mptDefineFlow`, `mptCheckFlow`, and `mptReportCurrentSettings` to define the MPT flows.	
Viewing the Last Operation Status and Probing the Color Source	New
You can view the status of the last coloring operation based on the color indicated in the last operation status icon on the Multiple Patterning toolbar. You can also probe the color source and mark the color information using the *Probe Color Source* and *Mark Color Info* options on the Multiple Patterning toolbar.	
Verifying the Consistency of Color Assignments	New
You can ensure that the design constraints are followed in the design by verifying the consistency of color assignments between a schematic and layout. There are two methods to verify the consistency of color assignments - Virtuoso coloring check and Layout Versus Schematic (LVS) coloring check. **Related:** `colorConstFileColorAName`, `colorConstFileColorBName`, `colorCDFCheck`, and `colorConstFileName`.	

Environment Variable

`displayMaskColorMode`	New
Controls the granularity of the *Show/Hide Color* function on the Multiple Patterning toolbar.	

Virtuoso Abstract Generator

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Abstract Generator User Guide*

ICADVM18.1 ISR13

SKILL Functions

absExtract	Enhanced
A new option, `ExtractAntennaPinPathsOnly`, has been added to control whether all nodes or only pin nodes and pin-connected nodes are considered for antenna number generation.	

Documentation

Abstract Generator SKILL Functions Documentation Moved	Change
The Abstract Generator SKILL functions documentation has been moved from the *Virtuoso Layout Suite SKILL Reference* into the *Virtuoso Abstract Generator User Guide*.	

ICADVM18.1 ISR12

Features

Support for 32 Oxide Models	Enhanced
Abstract Generator now supports up to 32 antenna models. You can select these from the *Oxide* list in the Antenna tab of the Extract step. By default, Abstract Generator extracts antenna ratio factors by using the `antenna` constraint. To extract these factors by using the `cumPerLayerAntennaRatio` constraint, set the `ExtractAntennaExcludeCumPerLayerAntennaRatio` option of the `absExtract` function to `false`.	

ICADVM18.1 ISR10

Licensing Updates

Changes in Licensing	Enhanced
When a library is opened, if the technology file has data reflecting geometries of 20nm, 16nm, or 14nm, the `Virtuoso_Adv_Node_Opt_Lay_Std` license (95512) is checked out. If the technology file has data reflecting geometries of 10nm or smaller, the `Virtuoso_Adv_Node_Opt_Layout` license (95511) is checked out. Earlier, Abstract Generator checked out advance node licenses even when the technology file did not have advance node data.	

ICADVM18.1 ISR9

What’s New and Enhanced

Feature

Specifying Must-Join and Must-Connect Relations	Enhanced
The Extract page now includes a section to specify how Abstract Generator should treat terminals that have the must-connect attribute or the `LEF54_MUSTCONNECTPORTS` property. Additionally, the `abstract.options` file includes options to specify must-join and must-connect relations.	

SKILL Functions

`absExtract`	Enhanced
Two new options have been added to specify how Abstract Generator establishes must-connect relations between disjoint groups of extracted shapes: `ExtractMustConnectAllPinsAlways` and `ExtractMustConnectTerminals`. **Note:** The Extract page of Abstract Generator no longer offers the options to specify must-connect relations.	
`absPins`	Enhanced
A new option, `AbstractRemoveRedundantBlockages`, has been added to control whether Abstract Generator removes redundant blockages when the shapes of the same layer coincide with each other.	

Removed Feature

Support Removed for Some Symmetries for Cells	Removed
Abstract Generator no longer supports these symmetries for cells: X90, Y90, and R90. These OA symmetries were earlier included for backward compatibility and do not have any mapping with LEF symmetry values. Existing designs will not be impacted by this change.	

ICADVM18.1 ISR4

SKILL Function

iagGenAbstract	New
Runs Integrated Abstract Generator in the Virtuoso design environment.	

ICADVM18.1 ISR3

What’s New and Enhanced

Feature

Saving the Status File	New
Abstract Generator now saves the status file (`.abstract.status`) in the `.abstract/<library name>/cellStatus` directory in the current working directory. Information about each cell is stored in a separate status file.	

Removed Feature

Support for TLF Format	Removed
Starting from the ICADVM18.1 ISR3 release, Abstract Generator does not support the TLF format. Therefore you cannot import logical data from a TLF file.	

ICADVM18.1 ISR1

What’s New and Enhanced

Features

Generating Antenna Numbers for All Terminal Directions	Enhanced
Abstract Generator now generates antenna numbers for any terminal direction type of a pin, including `input`, `output`, `inout`, `tristate`, `unused`, `switch`, and `jumper`.	
Ignoring WSP Regions	Enhanced
Abstract Generator now automatically detects shapes placed in width spacing pattern (WSP) regions and ignores these shapes while creating pins and blockages.	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Saving and Loading the Options File	New
Abstract Generator now saves the options file (`.abstract.options`) in the `.abstract` directory in your current working directory and also loads it from there. A message notifies you about the new default location of the options file.	
Integrated Abstract Generator Is the Default Interface	New
The integrated Abstract Generator is now the default interface in the Virtuoso design environment. To access the integrated Abstract Generator, you no longer need to set the shell environment variable `USE_AGILE_UI` before launching Virtuoso.	
Creating Pins from Labels	New
Two new fields, *Create Pins From Labels* and *Label to Pin Map*, have been added to the *Pins* section of the integrated Abstract Generator interface. These fields help create pins by mapping overlapping geometries to text labels.	
Removal of Verify Step and Grid Analysis	Removed
Abstract Generator no longer supports the Verify step and the grid analysis features. As a result, all the Verify-related menus and the *Grids* tab on the Running step Abstract form have been removed from the standalone Abstract Generator interface.	

Virtuoso Floorplanner

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Floorplanner User Guide*
- *Virtuoso Layout Suite XL User Guide*
- *Virtuoso Layout Suite SKILL Reference*

ICADVM20.1 Base Release

What's New and Enhanced

Features

Specifying the Objects to Be Generated During Physical Hierarchy Generation	Enhanced

The Generate Physical Hierarchy form now includes a *Generate Options* section, which lets you select the objects to be generated when the physical hierarchy is generated. The options in this section are:

- *Create Instances*: Generates instances in the layout canvas.
- *Create Pins*: Generates pins below the PR boundary.
- *Create Boundary*: Generates the PR boundary.

New Options in the Push Into Blocks Form	Enhanced

New options have been added to the following sections in the Push Into Blocks form:

- *Options* section
  - *Push Whole Shape*: Pushes overlapping whole shapes into the selected soft block without clipping them.
  - *Push Selected*: Pushes only the selected nets and shapes inside the soft block.
- *Net Filter* section
  - *Float*: Pushes floating net shapes into blocks.
  - Push modes: Lets you set the push mode to *Auto Push*, *Create Net & Push with Feed Thro Pins*, or *Push as Blockage*.
- *LPP Filter* section: Lets you specify the layers that can be pushed into soft blocks. Specify the required layers in the *LPP* field or click *Sync with Palette* to filter LPPs based on their selectability settings in the Palette assistant.

Related Environment Variables:

`pibCellName`, `pibFloatPurposeNames`, `pibLayerNames`, `pibLibName`, `pibNetType`, `pibPurposeNames`, `pibPushFloats`, `pibPushMode`, `pibPushWhole`, `pibViewName`

Specifying Pin Connectivity when Creating Pins from Net Shapes	Enhanced
The Create Pin from Net Shapes form now includes a *Pin Connectivity* section that specifies routing preferences.	
Auto-Create Pins Command Support for Straddling Shapes	Enhanced
The *Auto-Create Pins* command has been enhanced to recognize straddling shapes and create corresponding boundary pins.	

ICADVM18.1 ISR13

What’s New and Enhanced

Features

Creating Pin Labels Automatically	New
(ICADVM18.1 EXL Only) The Auto-Generate Hierarchy form now includes the *Create Labels* option, which lets you specify whether labels are to be generated along with the soft block pins.	

Environment Variables

pibPushSelection	New
Determines whether the Push Pre-Routes command can be run on selected nets.	
deletePinCheckbox	New
Specifies whether the existing pins that were created by the Auto-Create Pins tool are to be deleted before generating new ones.	
autoPinCreateFigGroup	
Specifies whether the Auto-Create Pins tool needs to create pins as groups or figGroups.	
mustConnectAllPins	New
Determines whether the LEF58_MUSTJOINALLPORTS property is to be applied to all newly created pin terminals in the layout.	
createPinPathSpineAware	New
Determines whether pins created in the *Single* mode are path spine aware.	

ICADVM18.1 ISR12

What's New and Enhanced

Features

Adding Hierarchical Instance Names to Promoted Pin Names	Enhanced
The Promote Pins command now includes the *Prefix Instance Name* option, which lets you prefix the pin source paths to the promoted pin names.	
Aligning Unconnected Bus Pins	Enhanced
The pin aligner now supports the alignment of unconnected bus pins. To run this feature, you need to set the environment variable `ignoreConnectivityBusPinsAlign` to `t`.	

Environment Variables

`autoPinPathSpineAware`	New
Determines whether signal pins created in the *Single* mode are path spine aware.	

ICADVM18.1 ISR11

What's New and Enhanced

Features

New Options in the Auto Pin Form	Enhanced

The following new options have been added to the Auto Pin form:

- **Check For Duplicate Pins:** Checks for the presence of any similar pins in the design. If found, then duplicate pins are not created.
- **Enable Coloring:** Uses the multiple patterning coloring engine to assign colors to the new pins based on their positions relative to the WSP tracks. This option is available only when the `coloringEngineEnabled` environment variable is set to `t`.
- **Create Metal Shapes Under Pin:** Creates a metal shape under each auto-created level-1 (soft block) pin. The metal shapes are assigned to the same layers and have the same dimensions as the parent pin, but their purpose is set to `drawing`.

SKILL API: The `vfpAutoPin` SKILL function now supports the following arguments:

- `checkDuplicatePin`
- `enableColoring`
- `createMetalShape`
- `metalShapeType`

Running Pin Optimizer after Auto-Generating the Hierarchy	Enhanced

(ICADVM18.1 EXL Only) The Auto-Generate Hierarchy form now includes the *Run Pin Optimization* option, which lets you run Pin Optimizer hierarchically on the generated blocks. The supported modes for running Pin Optimizer are *Top Down* and *Bottom Up*.

Environment Variables: `autoGenHierRunPinOpt`, `autoGenHierPinOptMode`

Specifying Pin Layers while Pushing Top-Level Structures to the Block Level	Enhanced

The Push Into Blocks form now includes a new section, *Create Pin Layer Selection*, which lets you specify the layers on which pins are to be created.

Environment Variables: `pibCreatePinLPPChoice`, `pibPinPurposeName`, `pibPinLPPName`

Specifying Terminal Attributes in the Pin Connectivity Model Form	Enhanced
The context menu for terminal names in the Pin Connectivity Model form now includes new options to set the following terminal attributes: ■ The physical-only attribute ■ The route and pin connectivity methods ■ The signal type	

Documentation Updates

Electrically Aware Pins	New
(ICADVM18.1 EXL Only) Documentation is now available for the options in the Electrically Aware Pin Sizing form, which is part of the Pin Tool. The Pin Tool has been enhanced to use the capabilities of Layout EAD to derive the electrical requirements of a design. You can use the Electrically Aware Pin Sizing form to identify under-sized pins that do not meet the electrical requirements of the design and resize them to avoid potential electrical violations.	

ICADVM18.1 ISR10

What's New and Enhanced

Features

New Options in the Promote Pins Form	New
The following new options have been added to the Promote Pins form: ■ **Delete Promoted Pins from previous run(s):** Deletes pins that were promoted during the previous runs of the command. ■ **Promote Visible Shapes:** Promotes only those pins that are available on visible layers. ■ **Promote Pins considering connectivity:** Honors existing pin connectivity to top-level nets during the promotion of these pins. **Related Environment Variables:** promoteDeleteOldPins, promoteVisibleLPPs, promotePinConnectivity	
Merging Pins	New
You can now merge two or more overlapping rectangular or rectilinear pin shapes into a single polygonal pin shape. The Merge and Must Connect Pins form lets you merge pin shapes and apply the must-connect-all property to the merged pin shapes. **Related Environment Variables:** mergePinsMustConnectMode, mergePinsSelMode, mergePinsSuffix	
Updating the Layer Purpose Pairs of Labels	Enhanced
The Floorplan Global Options form now includes the *Label Layer Purpose Pair Update* section. This section provides options to control the layer and purpose assignments of pin labels when pins are re-layered. **Related Environment Variables:** labelLayerPurposePair, sameLabelPurpose	
Pin Planner Command in the Floorplan Menu	Enhanced
Earlier, the Pin Planner could be accessed by selecting *Floorplan – Pin Optimization* to open the Pin Placement form, and then clicking the *Pin Planner* tab. In addition to this, now, the *Pin Planner* command has been added to the *Floorplan* menu, which lets you directly open the *Pin Planner* tab.	

SKILL Functions

vfpAutoPin	Enhanced
The `vfpAutoPin` SKILL function now supports the following arguments: ■ `mode` ■ `boundaryMultipleMode` ■ `buriedMultipleMode` ■ `boundarySignalPinLayers` ■ `buriedSignalPinLayers`	

ICADVM18.1 ISR9

What’s New and Enhanced

Environment Variables

enableNoPinsBetweenBusPins	New
Prevents placing other pins between bus pins.	

ICADVM18.1 ISR8

What's New and Enhanced

Features

Congestion-Aware Pin Placement	New
The *Pin Optimization* tab of the Pin Placement form now includes a *Congestion Aware* option. With this option selected, Pin Optimizer internally uses the results from Global Router to find congestion-aware placement positions for pins and places the pins accordingly.	
Schematic Awareness while Promoting Pins	Enhanced
The Promote Pins form now includes a *Schematic Awareness* option to promote only those pins that have a corresponding schematic counterpart at the top level. **Related Environment Variable:** `promoteSchematicAware`	

SKILL Functions

`vfpLoadPhysicalView`	Enhanced
The `vfpLoadPhysicalView` SKILL function now supports the following arguments: ■ `updateInstanceOptions` ■ `updateDesignOptions` ■ `addGeometryOptions` ■ `updateBoundaryOptions` ■ `replaceObstructionOptions`	

ICADVM18.1 ISR7

What’s New and Enhanced

Features

Maintaining the Area of Top-Design PR Boundary during Stretch	Enhanced
The *Use Constant Area Stretch* option on the Stretch form can now be applied to Top-Design PR Boundary to keeps the area of the top-design PR boundary constant when the boundary is stretched.	

ICADVM18.1 ISR6

What’s New and Enhanced

Features

Creating Pins Automatically	New
The Pin Tool now includes the Auto Pin form, which lets you generate pins automatically. The form provides options to search for net shapes in a design, set the pin generation options and pin attributes, preview pins, and generate pins in the layout canvas. **Related SKILL Function:** `vfpAutoPin`	

ICADVM18.1 ISR5

What's New and Enhanced

Features

Promoting Pins	New
The Promote Pins form lets you promote pins from a lower level in a design hierarchy to a higher level, which can be the current level or lower than it. You can specify either the pins or the nets or instances that contain the required pins for promotion. The form also lets you specify various promotion options such as promotion of selective bus pins, the prefix and suffix for the promoted pin names, the layer to which pins must be promoted, and schematic awareness. **Environment Variables:** promoteCreateLabelsOnPins, promoteCustomPurpose, promoteCustomLPP, promoteLayerChoice, promoteOddEvenChoice, promotePinNetChoice, promotePinNetList, promotePinOnNetForSelInsts, promotePinOnNetPickLevel, promotePrefixChoice, promotePrefixString, promoteSchematicAware, promoteSelectiveBusBits, promoteSelectInCanvas, promoteSuffixChoice, promoteSuffixString, promoteToChoice, promoteTopLayerFig, promoteToLevel	
Automatically Generating Hierarchy	New
You can now use the Auto-Generate Hierarchy functionality to configure top blocks and soft blocks, specify bindings, and generate these blocks in the layout canvas. The Auto-Generate Hierarchy form is an integrated, simplified version of the Configure Physical Hierarchy (Soft Block mode) and Generate Physical Hierarchy forms.	
Aligning Level-1 Pins with Level-1 Routes	New
The Pin Planner now lets you align level-1 pins with level-1 routes that belong to the same net. Such alignment is supported both, in the presence and absence of guide pins.	
Creating Free Area Pin Group Guides	New
The *Guides* tab of the Pin Group Guide from includes a new option, *Free Area*, which lets you create free area pin group guides in the free area within the PR boundary. You can create both polygonal and multi-segment pin group guides. The Pin Optimizer places the pins accordingly, while maintaining design correctness.	
Snapping Pins to WSP Tracks	New

The Snap Pins form provides additional options to control snapping of pins to WSP tracks. You can control the constraint awareness, layer awareness, and pin resize behavior of pins during snapping.

ICADVM18.1 ISR2

What’s New and Enhanced

Features

Accessing the Floorplanner Menu in Layout EXL	New
In Layout EXL, the *Floorplan* menu has been moved under the *Plan* menu.	

ICADVM18.1 Base Release

What’s New and Enhanced

Features

Introducing the Pin Assistant	New
Virtuoso Floorplanner now includes the Pin Assistant, which is a unified interface for all pin-related tasks, such as creating, resizing, planning, and optimizing pins. The assistant also lets you edit the pin attributes and set their location constraints.	
Displaying Block Annotations	New
Virtuoso now provides an option to display block annotations while you floorplan a design. Block annotations provide concise information about each block. Any change you make to a block is updated dynamically in the annotation.	
Floorplan Global Options Form	New
The Layout GXL Options form is now renamed to Floorplan Global Options.	
Aligning Level-1 pins to Top-Level Routes	New
The Pin Planner now includes the *Top-Level Route* option in the *Edge* list, which lets you align level-1 instance pins with the top-level routes to which they are connected. You can also choose to resize the pins according to or to match the dimensions of the top-level routes.	
New Options in the Analyzing Connectivity Form	New
The following options have been added to the Analyze Connectivity form: *Width – User Defined*: Lets you specify the width of flight lines. *Filter Nets*: Lets you hide the specified type of net.	
Pushing Routes as Blockages	New
The Push Into Blocks form now includes the *Push All As Blockages* option that lets you push the routes that are connected to blocks as blockages, and not as routes.	

Documentation Updates

Reorganized Floorplanner Quick Start and Migration Content	New
The *Preview to Virtuoso Floorplanner Migration Guide* and *Virtuoso Quick Start: Virtuoso Floorplanner Flow* guides are no longer available. Information from these guides has been added to the *Virtuoso Floorplanner User Guide* in a new chapter, Migration from Preview to Floorplanner, which provides information about the differences in the functionality and use models of both tools.	

Virtuoso Module Generator

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Module Generator User Guide*
- *Virtuoso Layout Suite SKILL Reference*

ICADVM18.1 ISR7

SKILL Functions

gpeCreateAlignmentAndSpacing	New
Specifies the alignment and spacing settings for all the grid entries in the given Modgen sandbox object.	
gpeGetGridEntry	New
Returns a grid entry that matches the given argument for the given Modgen sandbox object.	
gpeInsertEmptyColumns	New
Inserts empty columns in the specified Modgen sandbox object in the specified direction, starting at the specified column.	
gpeInsertEmptyRows	New
Inserts empty rows in the specified Modgen sandbox object in the specified direction, starting at the specified row.	

ICADVM18.1 ISR5

SKILL Functions

gpeCancelSandbox	New
Discards the specified Modgen sandbox object. Changes made to the Modgen sandbox object are lost.	

ICADVM18.1 ISR4

What’s New and Enhanced

Features

Support for Stacked Devices	New
Modgens have been enhanced to recognize stacked devices. The devices in a properly arranged stack are abutted automatically. Each abutted stack is represented as a single symbol, as if it were a single device, in the GPE and GPM assistants. You can add dummies around stacks. **SKILL APIs:** gpeStacksCompress, gpeStacksAreCompressed, gpeStacksUncompress	
Support for Reusable Modgen Templates	New
Virtuoso supports a template-driven Modgen reuse solution to help improve layout productivity. A Modgen template comprises a set of Modgen parameters, such as the interdigitation pattern, abutment, dummy definitions, and spacing values, that can be reused to create a gridded layout of matching structures. You can use the Reuse Template Exaction form to generate a Modgen template file and the Apply Reuse Template form to reuse the template file on a set of devices to generate a Modgen. **SKILL APIs:** gpeTemplateExtractLaunchForm, gpeExtractTemplateFromMG, gpeLoadReuseTemplate	

ICADVM18.1 ISR3

What’s New and Enhanced

SKILL Functions

gpeExtractTemplateFromMG	New
Generates a reusable template for each Modgen constraint present in the specified constraint cache.	
gpeLoadTemplateLaunchForm	New
Invokes the Reuse Template form.	
gpeTemplateExtractLaunchForm	New
Invokes the Template Extract form.	

ICADVM18.1 ISR2

What’s New and Enhanced

Features

Adding Surround Dummies	New
The new Surround Modgen form provides greater control over adding surround dummies around Modgens. The Surround Dummies form lets you perform the following tasks: ■ Load default values from pre-registerd SKILL callback functions. ■ Select the sides for adding dummies. ■ Select the cellview to be used to create custom dummies. ■ Set dummy parameters. ■ Define connectivity. ■ Insert tap cells.	

SKILL Functions

`gpeSetMergeLayer`	New
Sets the `MergeLayer` parameter for the given Modgen sandbox object.	
`gpeCreateTrunkChain`	Enhanced
Supports a new optional boolean argument `?centerInChannel`, which specifies whether the trunk chain must be centered in the channel indicated by the `?anchorIndex` value.	

ICADVM18.1 ISR1

What’s New and Enhanced

SKILL Functions

gpeGetGridColCount	New
Returns the number of columns available in the grid of the given Modgen sandbox object.	
gpeGetGridRowCount	New
Returns the number of rows available in the grid of the given Modgen sandbox object.	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Converting a Mosaic to Modgen	New
You can now directly convert a mosaic in a layout design to a Modgen by using the *Convert—To Modgen* command in the *Edit* menu.	

SKILL Functions

`gpeStartSandbox`	New
Checks for a Modgen sandbox object that includes the specified instances. If one exists, it is opened for editing. If not, a new Modgen sandbox object is created.	

Virtuoso Custom Digital Placer

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Custom Digital Placer User Guide*
- *Virtuoso Layout Suite SKILL Reference*

ICADVM18.1 ISR12

What’s New and Enhanced

Features

Loading Boundary Cell and Tap Cell Information from Row Template	Enhanced
The Placement Planning form now honors the boundary cell and tap cell definitions in the selected row template form. Therefore, if a row template with boundary cell and tap cell definitions is selected in the *Row* tab, then these definitions are honored. The settings in the *Boundary Cell* and *Tap Cell* tabs are ignored.	

ICADVM18.1 ISR11

What’s New and Enhanced

Environment Variables

pinPlacementXGrid	New
Specifies the X-coordinate of the placement grid on which the Custom Digital placer places pins.	
pinPlacementYGrid	New
Specifies the Y-coordinate of the placement grid on which the Custom Digital placer places pins.	

ICADVM18.1 ISR7

What’s New and Enhanced

Features

Placement Planning Form	New
The Placement Planning form has been revamped for better usability. The options in the form are organized in the following four tabs, which control the various placement settings. ■ **Boundary Cell:** Add boundary cells around the core cells. ■ **Tap Cell:** Add single- and multi-height tap cells between standard cells. ■ **Row:** Generate rows in the placement area. ■ **Create:** Define the placement area and select the components to be generated.	

ICADVM18.1 ISR6

What’s New and Enhanced

SKILL Functions

vcpfeCreateRows	New
Estimates and creates rows in the layout canvas.	

ICADVM18.1 ISR5

What’s New and Enhanced

Environment Variables

hierVHPlace	New
Specifies whether the placer must descend into the hierarchy and place the instances inside it while placing virtual figGroups.	

Features

Updated Insert/Delete Filler Cells Form	Enhanced
The Insert/Delete Filler Cells form now provides a list of component types and cells, from which you can select the ones to be used as filler cells. Related: Environment Variables: ■ selectedFillerCells ■ selectFillerCellsFrom	

ICADVM18.1 ISR3

What’s New and Enhanced

Features

Support for Virtual Hierarchies	New
VCP has been enhanced to support virtual hierarchies in the Virtuoso® Design Planner environment. You can use the Placement Planning form to create rows in the PR boundary, cluster boundary, and virtual hierarchy boundary (area boundary).	

ICADVM18.1 ISR1

What’s New and Enhanced

SKILL Functions

vcpfePlaceTapCells	New
Inserts tap cells in the empty spaces between standard cells.	

Virtuoso Placer

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Placer User Guide*

ICADVM18.1 ISR5

What’s New and Enhanced

Features

Specifying RowRegionSpec Type Attributes for Row Templates	New
The Row Template Manager now includes the *Region Type* option that specifies the RowRegionSpec (RRS) type attribute. Virtuoso can automatically query, derive, and run the predefined actions that are specified in the RRS type attribute.	

Removed Features

Features

Running Virtuoso Placer in the Analog Refine Mode	New
The Analog Refine option has been removed from the Automatic Placement mode. Use the Auto P&R assistant to initialize, generate, place, and route layout designs automatically, as per your requirements. For more information about the Auto P&R assistant, see Virtuoso Automated Device-Level Layout Flow Guide.	

ICADVM18.1 ISR3

What’s New and Enhanced

Features

Automatically Computing Row Offsets	New
The Row Template Manager now includes the *Auto-compute row offset* option, which lets you choose whether the tool must automatically calculate and update the row offsets whenever rows are modified.	
Checking for Context-Aware Placement	New
The *Placement* options in the Batch Checker form now includes a *Context Aware* option, which checks for context-aware placement of standard cells. It also verifies whether the edge type and spacing constraints of tap cells have been honored during placement.	

ICADVM18.1 Base Release

What’s New and Enhanced

Features

Row-Based Snapping of Multiple Instances	New
Virtuoso Placer now supports movement and row-based snapping of multiple instances simultaneously. Each instance is snapped to rows in real time, while honoring the row attributes.	
WSP and CST-Aware Snapping of Pins	New
Virtuoso Placer now supports width-aware and constraint (CST)-aware snapping of pins. WSP snapping is possible only when the specified constraints, such as the `minwidth`, `allowedWidthRanges`, and `sig type`, are met.	
Support for Placement of Virtual Hierarchies	New
Virtuoso Placer has been enhanced to support the placement of virtual hierarchies in all three modes: Custom Digital, Analog Refine, and Like Schematic. Virtual figGroups are placed as they are inside the area boundary of the virtual hierarchy.	

Virtuoso Automated Device Placement and Routing Flow

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Automated Device Placement and Routing Flow Guide*

ICADVM18.1 ISR9

What's New and Enhanced

Features

Creating Modgens for Device Groups	New
The *Constraints* tab of the Auto Device P&R assistant now lets you create Modgens for the device groups identified by the circuit finders. To do this, set the `aprCreateModgens` environment variable to `t` before generating the constraints.	
Editing Modgens using the Auto Device Array Form	New
You can now use the Auto Device Array form to edit the Modgens that have been created using the Auto Device P&R assistant. The Auto Device Array form is accessible from both the Constraint Manager assistant and the Auto Device P&R assistant. The form provides options to edit Modgen patterns, create guard rings, and set routing preferences.	
Support for Modgens during Device Placement	New
The Virtuoso device-level automatic placer now has the capability to reshape Modgens while placing them. This helps achieve optimized placement of Modgens for the given floorplan. The rows and columns of the Modgen are modified while retaining the base pattern. Use the *Allow placer to reshape Modgen* option on the *Placement* tab of the Auto Device Array form to control this behavior.	
Running the Incremental Placement	New
The *Placer* tab of the Auto Device P&R assistant now includes the *Run ECO Place* option, which lets you run the placer only on the unplaced devices, without impacting the already placed devices in the design. The following SKILL function has been enhanced to support this feature: apPlaceAuto: A new argument `?incr` has been added.	
Importing Row Templates and Width Spacing Patterns (WSPs)	Enhanced
The *WSP/Row* tab of the Auto Device P&R assistant now includes the following new options: ■ *Use row template*: Lets you import row settings from an existing row template. ■ *Import Width Space Patterns*: Lets you import a WSP pattern from an existing WSP pattern file.	

ICADVM18.1 ISR7

What’s New and Enhanced

Features

Auto Device P&R Assistant - A Quick Help	New
A new task-based quick help functionality, Auto Device Placement and Routing Task Assistant, has been introduced for the Virtuoso automated device-level layout flow. You access this quick help by clicking the  button available in the Auto Device P&R Assistant. The Task Assistant provides task-oriented information to get you started with the Virtuoso automated device-level layout flow.	

ICADVM18.1 ISR4

What’s New and Enhanced

Features

Virtuoso Automated Device Placement and Routing Flow	New
ICADVM18.1 ISR4 introduces a new Virtuoso automated device placement and routing flow, which comprises a series of tasks to help you generate automatically placed and routed layouts for device-level designs in advanced node processes. This functionality addresses the strong demand for an automated flow at these nodes due to increasing layout complexity, which results in costly iterations. The new Auto Device P&R assistant offers a simple, yet powerful interface that guides you through the following sequential tasks that require minimal user intervention: **1.** Generating instances, nets, and pins, and preparing instances for placement and routing **2.** Generating device groups, constraints, rows, and routing grids **3.** Placing the design **4.** Adding base layer fill **5.** Routing the design	

Voltus-Fi Custom Power Integrity Solution L

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Voltus-Fi Custom Power Integrity Solution XL User Guide*
- *Voltus-Fi Custom Power Integrity Solution L User Guide*
- *Power IR/EM User Guide*
- *Power IR/EM Known Problems and Solutions*

ICADVM18.1 ISR4

What’s New and Enhanced

Features

QRC Run Information	Enhanced
The QRC run information in the IR/EM Results form can now be provided by browsing and selecting the QRC run directory and run name. In addition, when you switch between the IR and EM tabs, the selected *QRC Run* information remains intact.	
Support to Clear Plots for Selected Nets	Enhanced
The IR/EM Results form lets you clear plots for selected nets from the Virtuoso layout by selecting *Clear Plot*.	
Variables Form	Enhanced
The Variables form that opens on the *EM* tab of the IR/EM Results form, now has *Browse* buttons for variables that require a file or directory path to be specified.	

ICADVM18.1 ISR1

What’s New and Enhanced

Features

Selecting Nets for Plotting IR/EM Results Enhanced	Enhanced
The IR/EM Results form is enhanced to select all nets in the list box for plotting results. The following options are available: ■ *Power Nets* only show power nets in the list box. ■ *Signal Nets* only show signal nets in the list box. ■ *All Nets* show all power and signal nets in the list box. ■ *Select All Nets* selects all the nets listed in the table.	

ICADVM18.1 Base Release

Power IR/EM in Support Mode

From ICADVM18.1, Power IR/EM is now in support mode. The software is still available for use but will be removed in a future release.

It is recommended that Voltus-Fi-L now be used to perform IR drop and Electromigration (EM) analysis using extracted views.

Voltus-Fi-L provides a number of benefits over the earlier VPS-L product, including:

- Enhanced UI similar to VFI-XL, thus making it easier for customers to switch between VFI-L and VFI-XL
- Enhanced EM engine to now be common with VFI-XL use. Key benefits here include multi-threading and run-time improvements

Voltus-Fi Custom Power Integrity Solution XL

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Voltus-Fi Custom Power Integrity Solution XL User Guide*
- *Voltus-Fi Custom Power Integrity Solution L User Guide*
- *Power IR/EM User Guide*

- *Power IR/EM Known Problems and Solutions*

ICADVM20.1 Base Release

What’s New and Enhanced

Batch Commands

update pgv	New
Updates a power-grid view (PGV) incrementally.	
set mmsim pgv nets	Enhanced
The following new parameter is added: -voltage_source_nets: Specifies the names of nets that are connected only to voltage sources and for which a current file, with a .pti extension, is not available.	

ICADVM18.1 ISR13

What's New and Enhanced

Features

Support for CSV Format in EMIR Reports	New
You can now generate an EMIR report in CSV format in addition to text and HTML formats. You can do so by setting the `report` option of the `emirutil` command as `csv`.	
Support for reportCurrentAmp in EMIR Control Files	New
The EMIR control file now supports the `reportCurrentAmp` option. It specifies whether to report the current of the resistor in amperes (A) or milliampere (mA).	
Saving the Check State of Violation Markers	New
The check state of a violation marker is now saved in the *EM/IR* tab of Annotation Browser and is reloaded when you start a new Virtuoso session.	

ICADVM18.1 ISR12

What's New and Enhanced

Batch Commands

create_pgv	Enhanced
The following new parameters are added: ■ -disable_signal_pgdb: Checks the nets or pins with port shapes in the LEF file and confirms that the input results database contains data for these nets. ■ -stop_on_missing_lef_pins: Specifies whether to generate PGVs even if the emir_bin database does not contain the data of the LEF pins or nets.	
run_top_simulation	Enhanced
The following new parameter is added: -use_config_file_flow: Specifies whether to store tapi commands in the vfi_emir.conf file instead of the vfi_input.scs file.	

SKILL Functions

vsaSetParallelLoading	New
Enables the parallel loading of OpenAccess shapes in the Voltus-Fi application interface.	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

Rearranging Columns in the Query Box	Enhanced
You can now rearrange columns in the Query Box. The new order is also saved when you save the configuration.	

Batch Commands

`create pgv`	Enhanced
The following new parameter is added: `-use_pwl_files`: When set to `true`, searches for the net-specific PWL files in the simulation directory and passes them to LibGen.	

ICADVM18.1 ISR10

What’s New and Enhanced

Batch Commands

`create_pgv`	Enhanced
The following new parameter is added: `-powergate_state_flow`: When set to `true`, searches for the power gate state files (`*pwg_state`) and passes them to LibGen.	
`extract_xdspf`	Enhanced
The following new parameter is added: `-cclFileName`: When specified, the CCL file is named as `vfiQuantus-<`*`filename`*`>.ccl`.	

ICADVM18.1 ISR9

What’s New and Enhanced

Features

Extract xDSPF Form	Enhanced
The following new fields have been added to the Extract xDSPF form: ■ *Add Extract Command*: Specifies that Voltus-Fi adds the extract commands to the CCL file. ■ *Reset*: Reverts to the default values.	
Voltus-Fi Extract xDSPF GUI Last File	Enhanced
Now, the values specified in the `.cdsinit` file override the values of the `.vfiExtractDSPFGuiLast` file of the previous session.	

ICADVM18.1 ISR8

What's New and Enhanced

Features

Optimizing the xDSPF File	New
Voltus-Fi uses a full-chip flat xDSPF and optimizes it for faster Spectre simulation. After an xDSPF file is optimized, Voltus-Fi automatically runs the SPF Checker utility on the optimized file to ensure that the DSPF file is compatible with the simulation requirements. To generate an optimized xDSPF file, run the `esd_optimizer` command in batch mode or use the new *ESD Optimizer* tab in the ESD form.	
Performing Rule-Based ESD Checks	New
Voltus-Fi performs rule-based resistance and current density checks. You can specify the rules in a single rule file. Voltus-Fi gets information of all the nets and clamps based on the inputs provided in the *ESD Common Setup* section of the ESD form, and runs resistance and current density checks in the background for the rules specified in the rule file. Voltus-Fi generates a pass/fail summary report and a detailed report for each rule. To perform rule-based ESD checks, run the `analyze_esd_network` command in batch mode or use the new *ESD Rule Check* tab in the ESD form.	
Extract xDSPF Form	Enhanced
The following new fields have been added to the Extract xDSPF form: ■ *Autorun Spfchecker*: Runs the SPF Checker utility after every extraction run. ■ *Filter RES Center*: Includes sub-nodes after eliminating resistors with small resistance values. ■ *Filter RES By Layer*: Specifies the minimum resistance value for the extracted parasitic resistors on the `RH_TN_6` LVS layer. ■ *Post Processing Script*: Specifies the post-processing script file.	
EMIR Reports	Enhanced
The EMIR reports generated by Voltus-Fi have been improved. You can now view operating temperature of the devices in the reports.	
Voltus-Fi Quantus Log File	Enhanced

Voltus-Fi now saves the `vfiQuantus.log` file in the output directory that is specified in the Extract xDSPF form.

Batch Command

`analyze_esd_network`	New
Performs rule-based resistance and current density checks in batch mode.	

ICADVM18.1 ISR7

What's New and Enhanced

Features

ElectroStatic Discharge (ESD) Analysis Form	Enhanced
The following enhancements are made to the ESD form: ■ The *ESD Setup* tab of the ESD form is enhanced to support the grouping of cells into two categories, namely *ESD Cell Devices* and *CDL Cell Devices*. These categories are displayed in the *ESD Instances* group box. The categories can be expanded to view and select the cell instances that are to be processed for the effective resistance and current density checks. For details, see ESD Instance Information. ■ The *Advanced Settings* button on the *ESD Setup* tab now allows you to run ESD zap simulations in the distributed Load Sharing Facility (LSF) run mode, in addition to the multi-CPU mode. For details, see Advanced Settings.	

Environment Variable

`vfiNoOfThreads`	New
Specifies the maximum number of threads to be used for EM analysis. It can be specified either in the `.cdsinit` file or the `.cdsenv` file.	

Batch Command

`esd optimizer`	New
Generates an optimized xDSPF file for faster Spectre simulation.	

ICADVM18.1 ISR6

What's New and Enhanced

Features

Support to Select Columns in the Query Box	Enhanced
In this release, the query feature of the software is enhanced to customize the columns you want to view in the Query Box pop-up window.	
New Keywords for Defining EM Rules	Enhanced
The following new keywords are added to specify the EM rules: ■ `Sv` = Specifies the distance between the two top vias enclosing a metal/via. ■ `Vu_current_direction {up | down}` = Specifies the current direction of the top vias across a metal/via. ■ `supply_net` = Specifies that the EM rule is applicable only to the supply net. ■ `ground_net` = Specifies that the EM rule is applicable only to the ground net.	
Support for N3 Process Node	Enhanced
The embedded extraction flow in Voltus-Fi is enhanced to support the N3 process node, in the GUI and batch mode. In the GUI, a new option is added in the Extract xDSPF form. In the batch mode, a new argument is added to the `-process` parameter of the `extract_xdspf` command. `extract_xdspf -process {n3 | n5 | n7 | n10pg | n10 | n20 | n16 | n12 | others}`	
ESD Form	Enhanced
The ESD Current report form that opens on clicking the *ESD Currents* button in the *ESD Run* tab of the ESD form, now has the *Save* button to save the current density report.	

Environment Variables

`vfiAutoChkBox`	New
Specifies to set the via array spacing automatically. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	

vfiBlocking	New
Specifies the file containing parasitic blocking device cell names, if any. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiCouplingmode	New
Specifies the coupling mode for capacitance extraction. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiColoring	New
Specifies to enable color-aware EM analysis. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiCompression	New
Specifies to generate xDSPF in the compressed format. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiCustom	New
Specifies the file with user-customized extract commands. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiDatatype	New
Specifies whether the input data will come from Pegasus or Calibre. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiElementCount	New
Specifies the number of violations you want to view in the *EM/IR* tab of the Annotation Browser, and the violation markers in the plot on the Virtuoso layout. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiEMOnlyIctFile	New
Specifies the ICT file that includes information about process and EM models to be used for EM analysis. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiExtractmode	New
Specifies the type of extraction to be performed. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiGndNet	New
Specifies the reference node on which extraction is to be performed. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiInputdir	New

Specifies the path to the directory in which the input data is stored. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiIRThreshold`	New
Specifies the IR threshold for viewing IR drop violations above the specified threshold value. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiLayermapFile`	New
Specifies the file used to map the layer names in the simulation database to the layer names in the technology file. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiMaxviarrcount`	New
Specifies the number of vias per side allowed in an array. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiMulticpu`	New
Specifies the number of CPUs to be used during the Quantus extraction run. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiNetsfile`	New
Specifies the file that contains the names of the nets to be included in extraction. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiOutdir`	New
Specifies the output directory in which the xDSPF file will be saved. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiOutdspf`	New
Specifies the name of the xDSPF file that will be generated. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiPgnets`	New
Specifies the power and ground nets that should be included in the xDSPF file. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiPinorder`	New
Specifies the sub-circuit pin order file to provide the order of pins to be used during extraction. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiProcess`	New
Specifies the process node. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiQrcRunDir`	New

Specifies the path to the QRC run directory. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiQrcRunName`	New
Specifies the QRC run name. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiReduction`	New
Specifies to enable the `reduce_i_cards` option in the CCL file that is used for generating the xDSPF file. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiResultDirectoryIr`	New
Specifies the state directory or the result file that stores the simulation results. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiRunname`	New
Specifies the name of the Pegasus or Calibre run. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiSignalem`	New
Specifies that signal nets are to be included in the xDSPF file. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiSubckt`	New
Specifies to comment out the top sub circuit statements from the xDSPF file. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiSubCondModel`	New
Specifies that the parasitic resistor models for subconductor layers should be included during extraction. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiSubNode`	New
Specifies the subnode character that will be used for subnode identification. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTechCorner`	New
Specifies the name of the technology corner to be picked from the library. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTechFile`	New
Specifies the technology file. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTechLib`	New

Specifies the path of the technology file to be used for reading the technology data. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTechName`	New
Specifies the technology filename to be used for extraction. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTechType`	New
Specifies the name of the directory that contains the technology file. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTemperature`	New
Specifies the temperature at which extraction will be performed. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTypeCheckboxMax`	New
Specifies to calculate the Current Density violations based on the maximum DC current. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTypeCheckboxAvg`	New
Specifies to calculate the Current Density violations based on the average value of the DC current. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTypeCheckboxAvgAbs`	New
Specifies to calculate the Current Density violations based on the average value of absolute current. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTypeCheckboxAcpeak`	New
Specifies to calculate the Current Density violations based on peak AC current. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiTypeCheckboxRms`	New
Specifies to calculate the Current Density violations based on the root mean square (RMS) value of the AC current. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiViaArrayspacing`	New
Specifies to enable via array spacing for grouping vias within the same array. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
`vfiViaCnt`	New
Specifies the fracture via count to be used for dividing pseudo vias into segments during extraction. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	

vfiViaSpacingEdit	New
Specifies a value for via array spacing. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	
vfiViaSpacingEditChk	New
Specifies to use the user-specified value for via array spacing. It can be specified either in the `.cdsinit` or the `.cdsenv` file.	

Batch Command

extract xdspf	Enhanced
The following new parameters are added: `-keep_pin_shorting_res`: Preserves the device pin shorting resistors on the `*wires` layers. When set to `true`, the shorting resistors will not be removed. `-keep_pin_shorting_res_by_layer`: Specifies the shorting layer names to be preserved.	

Variable

enableReportScaleParam	New
When set to `true`, this option reports the layout scale factor in the EM text report. In addition, the scaled dimensions for the `path-length`, `width`, `needed Width`, and `viaArea` fields are displayed in the EM analysis report and GUI. The syntax is provided below: `emirutil enableReportScaleParam=true` It can also be specified in the Variables form that opens from the *EM* tab of the IR/EM Results form.	

ICADVM18.1 ISR5

What’s New and Enhanced

Features

ESD Form	Enhanced
The ESD form now has an *ESD Model* tab. This tab provides the option to select the type of Transmission Line Pulse (TLP) modeling information to be used for performing static and dynamic ESD analysis.	

ICADVM18.1 ISR4

What's New and Enhanced

Features

QRC Run Information	Enhanced
The QRC run information in the IR/EM Results form can now be provided by browsing and selecting the QRC run directory and run name. In addition, when you switch between the IR and EM tabs, the selected *QRC Run* information remains intact.	
Support to Clear Plots for Selected Nets	Enhanced
The IR/EM Results form lets you clear plots for selected nets from the Virtuoso layout by selecting *Clear Plot*.	
ESD Form	Enhanced
The ESD form now has *ESD Setup* and *ESD Run* tabs. The *ESD Setup* tab has common setup fields for resistance and current density checks. The *ESD Run* tab provides options for performing effective resistance and current density checks.	
Testbench Settings Form	Enhanced
The Testbench Settings form for ESD analysis is enhanced to: ■ Specify the default vales for resistance, capacitance and inductance ■ Include an image that depicts the testbench that Voltus-Fi realizes to create the input stimulus to the DUT for ESD analysis	
Advanced Settings Form	Enhanced
The *Simulation Settings* tab of the Advanced Settings form for ESD analysis is enhanced to let you click *Default* to specify the default values for simulation.	
Support to Save Run Settings in Extract xDSPF Form	Enhanced
Creating the CCL file and running Quantus by using the *Generate and Run* option in the Extract xDSPF form saves the form settings in the `vfiExtractDSPFGuiLast` file. This file is loaded automatically at the start of the next session.	
Variables Form	Enhanced
The Variables form that opens on the *EM* tab of the IR/EM Results form, now has *Browse* buttons for variables that require a file or directory path to be specified.	

Environment Variable

vfi_ignore_conf_file	New
When specified, the variable settings in the configuration file are ignored and the GUI settings specified for the variables are used.	

Batch Commands

set_mmsim_pgv_nets	Enhanced
The following new parameter is added: -feedthru_nets: Generates EM views for the specified feedthrough nets in the PGV flow.	
create_pgv	Enhanced
The following new parameter is added: -create_no_device_nets: When set to true, creates power-grid views even when the net has no current files.	

SKILL Functions

vsaShowAllResistors	New
When specified, all resistors on the Voltus-Fi GUI including those for which there are no layout shapes matching the specified resistor coordinates are displayed.	

ICADVM18.1 ISR3

What’s New and Enhanced

Features

Support for Specifying Simulation Settings for ESD Current Density Checks	Enhanced
The Advanced Settings form for ESD is enhanced to let you specify the following simulation settings while performing current density checks: ■ *Spectre Simulation Options* ■ *Spectre Command Line Options* ■ *Spectre Transient Options* In the batch mode, these options can be specified in the ESD configuration file.	

ICADVM18.1 ISR2

What's New and Enhanced

Features

Support for IR Drop Analysis Resistor Power Plot Type	Enhanced
The GUI is enhanced to let you plot resistor power, which is the value of current through a resistor multiplied by the voltage across the resistor. The option is available in the drop-down list of the *Net Plot* group box in the IR/EM Results form.	
Support for Comparing Reports in GUI and Batch Mode	Enhanced
The GUI is enhanced to support comparing reports generated in GUI and Batch Mode for different plot types. For this, a new option, *Compare Report*, is provided in the Display page of the IR/EM Results form. The difference between the reports can be viewed in the vimdiff window.	
Report Format Enhanced	Enhanced
Earlier, the GUI reports for the different analyses types generated using the *Save Report* option in the IR/EM Results form did not display any header, version number, and column heading information. In this release, the report format is enhanced to replicate the header and column heading information that is displayed in the batch reports.	
ESD Enhancements	Enhanced
The following enhancements are made for performing ESD Analysis in Voltus-Fi: ■ New tabs are provided for performing the following current density checks: ❑ DC Mode: Current Density Check for DC simulation model ❑ CDM Mode: Current Density Check for Charged Device Model (CDM) ❑ HBM Mode: Current Density Check for Human Body Model (HBM) ❑ PERC: Programmable Electrical Rule Check (PERC) checks ■ A new group box, *ESD Common Setup,* is provided for specifying inputs common to effective resistance checks and current density checks in the ESD form. ■ Support is provided for reading the DSPF file in the gzip format in GUI and batch mode.	

Environment Variable Made Obsolete	Enhanced
The `VFI_REMOVE_BULK_SHORTING` environment variable is made obsolete. When set to `true`, this variable removed bulk shorting in the design. This variable is no longer needed because this behavior is now made default.	

Variables

violation_columns	New
Specifies the columns to be displayed in the Violation Browser. You can specify multiple columns separated by a space. The syntax is provided below. `emirutil violation_columns=[LENGTH_COLUMN RESISTANCE_COLUMN]` The variable can also be specified in the Variables form that opens from the EM tab of the IR/EM Results form.	
genRthReport	New
When set to `true`, generates a thermal resistance report with cell name and its corresponding resistance (`Rth`) value. Default is `false`. This variable is only applicable to the thermal-aware DC flow. The syntax is provided below. `emirutil genRthReport=true`	

Batch Commands

analyze_esd_current_density	Enhanced
The following new parameter is added: `-simulation_model:` specifies the mode to be used for ESD event analysis for current density checks.	
analyze_esd_generic	Enhanced
The following new parameter is added: `-tech_file:` specifies the technology file that contains the EM limits for ESD analysis.	

ICADVM18.1 ISR1

What's New and Enhanced

Features

Selecting Nets for Plotting IR/EM Results Enhanced	Enhanced
The IR/EM Results form is enhanced to select all nets in the list box for plotting results. The following options are available: ■ *Power Nets* only show power nets in the list box. ■ *Signal Nets* only show signal nets in the list box. ■ *All Nets* show all power and signal nets in the list box. ■ *Select All Nets* selects all the nets listed in the table.	
Supporting Ternary Operator in the ICT File	Enhanced
The ICT file now supports ternary operator, which is an expression of the form: `expr1? expr2: expr3` Where, `expr1` is a conditional expression. If `expr1` evaluates to `true`, then `expr2` is evaluated, else `expr3` is evaluated.	
Adding Custom ESD Events or Zaps	Enhanced
Earlier, while performing ESD current density checks, zaps were generated by default by the software. In this release, the Voltus-Fi GUI is enhanced to let you add zaps for ESD current density checks. For this, a new tab, *Custom zaps*, is provided in the ESD form.	

Variables

`enableResShape`	New

When set to `true`, enables tagging of shapes on the layout based on full resistor shapes and terminal nodes. The syntax is provided below. `emirutil enableResShape=false` The variable can also be specified in the Variables form that opens from the EM tab of the IR/EM Results form.	
emVariableValues	New
Specifies the EM variable values for current density limit calculation in the EMIR configuration file. The syntax is provided below. `emirutil emVariableValues=[(variable=configA value=#) (variable=configB value=#)]` The variable can also be specified in the Variables form that opens from the EM tab of the IR/EM Results form.	
deltaTLayers	New
Specifies layer-specific deltaT values. The syntax is provided below. `emirutil deltaTLayers=[(layer=layer_name1 deltaT=temp_value1)(layer=layer_name2 deltaT=temp_value2)]` The variable can also be specified in the command file by using the set_variable command, and in the Variables form that opens from the EM tab of the IR/EM Results form.	
extendedRMSReport	New
When set to true, reports the `rmsT`, `Lv2v`, and relax factor values in the RMS EM report. The syntax is provided below. `emirutil extendedRMSReport=true`	
removeBulkShorting	New
When set to `true`, this variable removes bulk-shorting in the design. The syntax is provided below. `emirutil removeBulkShorting=true` The variable can also be specified in the Variables form that opens from the EM tab of the IR/EM Results form.	
layermap	New

Specifies the mapping between the layer names in the simulation database and those specified in the technology files for EM analysis. The syntax is provided below. `emirutil layermap=[(rcxLayer=rcx_layer_name1 ictLayer=ict_layer_name1)(rcxLayer=rcx_layer_name2 ictLayer=ict_layer_name2)]` The variable can also be specified in the Variables form that opens from the EM tab of the IR/EM Results form.

mdLayers	Enhanced
The default value of this variable is now `MD_STI MD_OD MD_STI_SRM MD_OD_SRM M0_OD M0_STI MD_STI_CPP85 MD_OD_CPP85.m`.	

Environment Variables

vfi_dfVFI_SPLIT_NON_RECTANGULAR_SHAPE	New
When specified, sets the option in the VFI GUI to split the non-rectangular layout shapes, based on the resistors extracted over them.	

SKILL Functions

vsaSetFinerDisplayOff	New
When specified, disables the finer gradient display for the IR/EM plots on the Virtuoso layout.	

ICADVM18.1 Base Release

Power IR/EM in Support Mode

From ICADVM18.1, Power IR/EM is now in support mode. The software is still available for use but will be removed in a future release.

It is recommended that Voltus-Fi-L now be used to perform IR drop and Electromigration (EM) analysis using extracted views.

Voltus-Fi-L provides a number of benefits over the earlier VPS-L product, including:

- Enhanced UI similar to VFI-XL, thus making it easier for customers to switch between VFI-L and VFI-XL
- Enhanced EM engine to now be common with VFI-XL use. Key benefits here include multi-threading and run-time improvements

What's New and Enhanced

Feature

ElectroStatic Discharge Analysis	Enhanced
The following enhancements are made in the ESD analysis flow in the GUI mode: ■ Specifying the In-House testbench: You can specify your own testbench setup and override the Voltus-Fi generated testbench. The testbench can be specified in either Spice or Spectre format. ■ Specifying the Tool-Generated Testbench Flow: Voltus-Fi can now generate a testbench for charged device model (CDM) and human body model (HBM) analysis. The testbench consists of resistance, inductance, and capacitance (RLC) circuits that create a stimulus in the zap pins for CDM/HBM circuit testing. ■ Specifying Package Parameters for Tool-Generated Testbench Flow: You can now provide package RLC values to be used in the testbench for HBM/CDM ESD analysis. ■ Transmission Line Pulse (TLP) Modeling: The TLP modeling of ESD devices is enhanced to let you specify the trigger voltage (V_t), on-resistance (R_{on}) value, and off-resistance (R_{off}) value to model ESD devices for static DC simulation. You can also specify the current voltage (IV) data for performing complete transient ESD analysis.	

Virtuoso Fluid Guard Ring

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Fluid Guard Ring User Guide*
- *Virtuoso Fluid Guard Ring Frequently Asked Questions*
- *Virtuoso Layout Suite SKILL Reference*

ICADVM18.1 ISR3

What’s New and Enhanced

Feature

Creating an FGR Automatically	Enhanced
The Create Guard Ring form has been enhanced to create an FGR around selected devices automatically. For this, a new *AUTO* option has been added to the *Device* field on the Wrap tab of the Create Guard Ring form.	

ICADVM18.1 ISR1

What’s New and Enhanced

Constraints

Enhanced Support for the Clean Overlapping Contacts Command	Enhanced
The *Clean Overlapping Contacts* command now lets you clean contacts from two overlapping cells that have their fluid guard rings (FGRs) in the hierarchy.	

Virtuoso Symbolic Placement of Devices

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Symbolic Placement of Devices User Guide*
- *Virtuoso Layout Suite SKILL Reference*

ICADVM18.1 ISR6

What’s New and Enhanced

Feature

Support for additional rows and mixed rows patterns	New
The *Row Pattern* option on the *Placement* pane in the SPD Options form, now supports the *(Arbitrary)* option that lets you specify more than four rows in the row pattern. You can also use the *Mix P/N in one row* option to specify whether both P and N type devices can be specified in a single row. Environment variable: `PMmixedRow`	

Virtuoso Design Planner

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Design Planning and Analysis User Guide*
- *Virtuoso Floorplanner User Guide*

ICADVM20.1 Base Release

What's New and Enhanced

Features

Unified Design Planning and Floorplanning Functionality	New
The Virtuoso Design Planning functionality has been enhanced to support an expanded range of floorplanning, design planning, and congestion analysis capabilities, including support for hard blocks, soft blocks, and virtual hierarchies. This enhanced support is accessible through a unified *Plan* menu, *Design Planning* workspace, and *Design Planning* toolbar, all of which are available from the Layout EXL window.	
Generate a Custom Virtual Hierarchy Block Size using SKILL	New
You can now register a user-defined SKILL function to generate a virtual hierarchy of custom block size during a *Generate All From Source* or an *Update Components And Nets* run.	
Adjust Area Boundary Hierarchically	Enhanced
The Adjust Boundary form can now be used to adjust the area boundary of the parent virtual hierarchy when a virtual hierarchy or a soft block inside it has its area boundary adjusted. Environment variables: adjustBoundaryCheckOutside, adjustBoundaryIncludeTop	
Multiple Cell Support for Make Virtual Hierarchy	Enhanced
The Make Virtual Hierarchy command can now be used to integrate multiple selected cells as virtual hierarchies.	
New Default Cell Type for Make Cell	Enhanced
The Make Cell command now creates softMacro cell types by default.	

SKILL Function

`lxIsVirtualFigGroup`	New
Checks if the specified figGroup is a virtual figGroup.	
`lxGetVirtualFigGroupMasterName`	New

Returns the schematic cellview name for the specified virtual figGroup in the format 'lib/cell/view'.	
lxHiDesignPlanningOptions	New
Opens the Design Planning and Analysis Options form.	
lxHiSnapPatternOptions	New
Opens the Design Planning and Analysis Options form to create snap pattern grids and specifies that horizontal and vertical tracks be created inside the PR boundary.	

ICADVM18.1 ISR13

What’s New and Enhanced

Features

Reclone Command in Virtual Hierarchy Context Menu	New
Use the new *Reclone* command in the virtual hierarchy context menu to: ■ Clone virtual hierarchies that belong to the same layout master. ■ Update a clone family to add new members. ■ Add a Modgen constraint to a virtual hierarchy.	
Enhanced Adjust Boundary Form	Enhanced
The Adjust Boundary form can now be used to adjust top-level rectangular PR boundaries in addition to adjusting area boundaries for virtual hierarchies and PR boundaries for soft blocks. The **Enclose by** option now specifies the distance between the rectangular or rectilinear boundary and the PR boundary of the instances inside a virtual hierarchy or soft block. When the instance PR boundary does not exist, the instance bounding box is used to determine the distance. ❑ The **Enclose by** option now supports two controls — *Instances only* and *All shapes* that can be used to specify which shapes within a virtual hierarchy or a soft block are enclosed. These options are supported for both rectangular and polygonal boundaries.	
Enhanced Make Cell Form	Enhanced
The Make Cell form can now be used to create real cellviews for multiple virtual hierarchies. In addition, the form can now been used to only push the internal routes to the made cell.	

Environment Variable

`generateSoftBlocksInTargetLibrary`	New
Controls soft block layout creation for read-only libraries.	

ICADVM18.1 ISR12

What’s New and Enhanced

Features

Generating Virtual Hierarchy for Selected Cells	New
The new *Generate – Virtual hierarchy* option on the Virtual Hierarchy Options form can now be used to selectively generate the virtual hierarchy for all hierarchy levels or for the cells selected using the CPH Cells table. Environment variable: `vhSelectiveMode`	
Controlling Congestion-Aware Pin Creation	New
Use the new *Pin layers* options on the Virtual Hierarchy Options form to control whether congestion-aware pins are created on all routing layers or are restricted to specific layers. Environment variables: `pinLayerLimit`, `pinLayerLimitNum`	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

Support for Hierarchical Make Cell Generation	New
The Make Cell form now supports an *All levels* option to enable layout view creation not only for the selected virtual hierarchy but also for the virtual hierarchies at all levels inside the selected virtual hierarchy.	
Support for PR Boundary Adjustment of Soft Blocks	New
The Adjust Boundary form can now also be used to adjust the PR boundary of a selected soft block. Related SKILL function: `lxHiAdjustBoundary`	

ICADVM18.1 ISR10

What's New and Enhanced

Features

Check Against Source Report Refreshed to Delete Previous Markers	Enhanced
To ensure that the *Check Against Source* command reports markers only for the selected virtual hierarchy, markers from a previously reported run are now deleted before a new report is issued.	
Make Cell – All Clones Option Supported Only for Virtual Hierarchy Clones	Enhanced
The *Make Cell – All Clones* option is now supported only for virtual hierarchy clones. For non-clone virtual hierarchies or when the *All Clones* option is deselected, the *Push – Into block* option is enabled.	
Additional Make Cell – Type Value Supported	Enhanced
The *Make Cell – Type* functionality can now support an additional `digital softMarco` value. Related environment variable: makeCellType	
Area Estimation for PR Boundary Deferred Until Virtual Hierarchy Area Boundary is Calculated	Enhanced
If the PR boundary area estimation function is registered as `deferred`, the PR boundary area estimation is now postponed till the virtual hierarchy area boundaries are created.	

Environment Variables

`updateSoftBlocksFromSymbol`	New
Updates the softblock master during an *Update Components And Nets* run if the bound schematic symbol is modified.	
`makeCellPushInBlock`	Enhanced
Default value changed to `nil`.	

ICADVM18.1 ISR9

What’s New and Enhanced

Features

Virtual Hierarchy Area Estimation using SKILL	New
Use the *Register* button on the Adjust Area Boundary form to register a user-defined SKILL function that can control the area estimation for selected virtual hierarchies.	

Environment Variables

`pinOptOneConnectionPerSide`	New
Controls the number of connections for congestion-aware pin optimization.	
`autoPlaceOnPRBoundaryEdit`	New
Controls whether the top-level design is automatically placed when the PR boundary is modified.	

ICADVM18.1 ISR8

What’s New and Enhanced

Features

Removing Highlights from All Virtual Hierarchies	New
Use the new *Unhighlight All Virtual Hierarchy* shortcut command to remove highlights from all the virtual hierarchies in the design.	
Generating Soft Blocks for Schematics with No Instances	Enhanced
The Design Planner can now generate soft blocks for schematics that have no instances or physical binding but only pins.	

ICADVM18.1 ISR6

What’s New and Enhanced

Features

Dynamic Display Measurements of a Virtual Hierarchy	New
You can now see dynamic measurements when performing a *Stretch* or a *Constant area stretch* for a virtual hierarchy, which can help you be more accurate when deciding the area of the virtual hierarchy.	

ICADVM18.1 ISR5

What’s New and Enhanced

Features

Viewing the Virtual Hierarchy Types	New
You can now easily identify the different types of virtual hierarchies in a design if the bounding box for each type uses a different color.	
Viewing the Virtual Hierarchy Placement Status	New
To know the placement status of a virtual hierarchy, you can now just look at its bounding box to see if it is a solid line.	
Adjusting the Virtual Hierarchy Area Boundary	Enhanced
In this release, several enhancements were made to the virtual hierarchy area boundary adjustment functionality. ■ Use *Width*, *Height*, and *Constant area* to control the area boundary of virtual hierarchies. ■ Adjust area boundary automatically across hierarchies to accommodate instances and figGroups, other than row region, that have moved outside of the virtual hierarchy area boundary. ■ Use *PR Boundary based* and instance *BBox based* area boundary estimators for rectilinear boundaries. Also register a user-defined SKILL function for area estimation of virtual hierarchy area boundaries of any shape.	

ICADVM18.1 ISR3

What’s New and Enhanced

Features

Enhanced User Interface and Functionality	Enhanced
In this release, the Design Planner was enhanced to support: ■ **Virtual Hierarchy Creation for Selected Schematic Instances**. The Design Planner menu and toolbar have been enhanced to support an additional command—*Generate Selected From Source*—to allow selected schematic instances to be used for virtual hierarchy generation. ■ **Congestion-aware, optimized net length, or on boundary pin creation for made cell**. The Design Planner Make Cell form now supports three new options to allow more control on pin creation. ■ Interactive folding **support from top level for devices inside a virtual hierarchy**. To fold the devices inside a virtual hierarchy, you can now choose them from the top level.	

ICADVM18.1 ISR2

What’s New and Enhanced

Features

Enhanced User Interface and Functionality	Enhanced
In this release, the Design Planner was enhanced to provide: ■ **Menu-based command access**. The Design Planner commands are now also available through the Plan menu in Layout EXL. ■ **Support to choose the new cell type to be created**. The Design Planner Make Cell command now enables you to choose the cell type—*custom*, *soft block*, or *hard block* for the new cell. ■ **Renamed placement status value**. The virtual hierarchy *Freeze* placement status is replaced with *Fix placement status*, supporting values `Fixed` or `None`.	

ICADVM18.1 ISR1

What’s New and Enhanced

Features

New and Enhanced Features and Enhanced User Interface	New
In this release, the Design Planner has not only enhanced what it offered, but has also introduced some new and exciting capabilities. Use the new Design Planner for its: ■ **Enhanced user interface**. *Detach* and *Embed* commands stand replaced with more intuitive *Make Virtual* and *Make Virtual Hierarchy* commands. ■ **Soft Block generation capability**. For blocks that have missing schematics, the Design Planner now allows automatic generation of soft blocks. ■ **Enhanced Navigator assistant support**. View information about interface nets, soft blocks, and see more informative tooltips. ■ **Enhanced selection support**. See the details of the selected virtual hierarchy instance or pin in the Show Selection Info toolbar.	

ICADVM18.1 Base Release

What’s New and Enhanced

Features

Virtuoso Design Planner — New Layout Design Planning Tool	New
In ICADVM18.1, Virtuoso introduces a new layout-place-route tool called Design Planner in Virtuoso Layout Suite EXL layout editor (Layout EXL). Design Planner allows for improved design planning at the top, block, and cell level. The in-built real-time congestion analysis support makes the congestion analysis data available early on in the design life cycle, making Design Planner an iterative layout-place-route tool.	

Virtuoso Concurrent Layout

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso Concurrent Layout User Guide*
- *Virtuoso Layout Suite SKILL Reference*

ICADVM20.1 Base Release

What's New and Enhanced

Features

Support Added for Layer-Based Partitions	New
You can now create layer-based design partitions in Concurrent Layout using the Define Design Partition form. Layer-based design partitions are useful for tasks such as routing and interactive wire editing. **Related:** Defining Layer-Based Design Partitions, Editing in a Layer-Based Design Partition **Environment variables:** `includeNonMaskableLayers`	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

Support to Check Against Source	New
You can now use the *Check Against Source* option in the *Manager Mode: Design Partition Options* menu to run the Layout XL command Check Against Source for all instances in the current design partition.	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

Enhanced the Delete Design Partition Option	Enhanced
When you delete a design partition by using the *Delete Design Partition option in the* Concurrent Layout assistant or the *Delete* button on the Define Design Partition form, the selected design partition and all associated design partition views in the hierarchy are deleted.	

ICADVM18.1 ISR9

What’s New and Enhanced

Features

Enhanced Support for Hierarchical Edit-in-Place from the Top Design Partition	Enhanced
Hierarchical editing in the CLE flow lets multiple designers to concurrently Edit In Place (EIP) into hierarchical subcells. Edit scope is retained because area partitions defined at the top are pushed down the hierarchy.	
Split Crossing Objects Options	Enhanced
The **Split Crossing Objects Options** form lets you specify how to split the objects that are part of single or multiple design partitions. Environment variables: autoSplitObjTypes autoSplitObjCrossingPartition splitObjFilterByPalette	

ICADVM18.1 ISR7

What's New and Enhanced

Features

Additional commands supported when edit scope is Only Edit Inside Partition	New
The Only Edit Inside Partition edit scope, now supports the following edit commands: *Edit - Copy*, *Edit - Stretch*, *Edit - Delete*, and *Edit - Quick Align*. The following commands are already supported by this option: *Create – Shape*, *Create – Instance*, *Create – Pin*, *Create – Label*, *Create – Fluid Guard Ring, Create – Wire, Create – Via*, and *Edit – Move*.	
More intuitive sign off options	Enhanced
The following right-click options have been added to the ***Details*** pane to enhance and speed up the sign off process: ■ *Set/Unset Highlight* ■ *Set/Unset Filter* ■ *Set/Unset Signed-off*	
Enhanced Support for Editing Hierarchical Designs	Enhanced
Use model of hierarchical editing has been enhanced to make it easier to first edit and then merge the hierarchical design partitions after editing. Related enhancements: The ***Hierarchical Edit Mode*** option has been added to the *Assistant* section of the Concurrent Layout Options form. Environment variable: cleHierEditMode	
Support to view instance occurrences in the hierarchical design	New
Concurrent Layout now lets you view instance occurrences when you are editing a hierarchical design. To use this option, in the Layout Configuration form, click the ***ViewOcc*** option. This will display the View Instance Occurrence form that lets you view the instance occurrences up to a specific display level.	

Support to check for design partition overlaps	New
Concurrent Layout lets you check for area overlaps in design partition. You can enable this feature using the following: ■ In Concurrent Layout assistant, use the ***Check Partition Area Overlaps*** command in the *Manager Mode Design Partition Options* menu. ■ On the Concurrent Layout Options form, use the ***Create partition views to check area overlaps*** option. ■ On the View Design Partition form, use the ***Check*** button to check the selected area for overlaps. Environment variable: createPartcheckAreaOverlap	
Support to translate design partition views to an OASIS file	New
You can now use the ***XOasis Out from VM for DRC*** option in the ***Designer Mode: Design Partition Options*** menu to translate the current design partition view to an OASIS file from virtual memory for running DRC checks. Option ***XStream Out from VM for DRC*** to translate the current design partition view to a stream file from virtual memory for running DRC checks was added in ICADVM18.1 ISR6.	

ICADVM18.1 ISR1

What’s New and Enhanced

Feature

Concurrent Menu	New
A new menu with flow-related commands for concurrent layout editing.	

Documentation Updates

Working with Concurrent Layout Editing	New
A new chapter has been added to *Virtuoso Concurrent Layout User Guide* to describe how to use the various options on the Concurrent menu and Concurrent Layout assistant to perform concurrent layout editing.	

ICADVM18.1 Base Release

What’s New and Enhanced

Virtuoso® Concurrent Layout (CLE *with E for editing*) has been introduced in the ICADVM18.1base release. Concurrent Layout is a layout editing environment that enables several designers to work concurrently on the same cellview within Virtuoso. This helps them in parallelizing their efforts and, in turn, increases the productivity of the layout design team.

Virtuoso Photonics Solution

This chapter provides a high-level overview of the new and enhanced features in the ICADVM18.1 and ICADVM20.1 releases.

Related Documentation

- *Virtuoso Photonics Solution Guide*

ICADVM20.1 Base Release

What’s New and Enhanced

SKILL Functions

phoAddWaveguidePorts	New
Adds circular optical ports at both ends of a ccPath.	
phoTechIsModePropLocalOnly	New
Retrieves the `localOnly` flag of the modeProp table for the specified waveguideDef. The table is searched only in the local technology database.	
phoTechLoadModePropDataTable	Enhanced
New key arguments, `localOnly` and `overwrite`, have been added.	

ICADVM18.1 ISR11

What’s New and Enhanced

SKILL Function

ccCreateConnectorWidthExpressionList	New
Creates a list of curved segments with the corresponding width expressions defining a tapered connector path of the specified tapering style with the specified start and end width around the centerline specified by the input polyCurve.	

ICADVM18.1 ISR10

What’s New and Enhanced

Environment Variable

generateSchematicMfactor	New
Creates mfactors for layout instances and dummies during a *Generate Schematic From Layout* run.	

SKILL Functions

cGenOffsetFig	New
Returns a copy of the specified polyCurve that is expanded by the specified offset value.	
phoDrawCurveDeltaConnectorSymbol	New
Generates the symbol artwork for a photonics Curve Delta connector waveguide.	

ICADVM18.1 ISR9

What’s New and Enhanced

SKILL Functions

ccCreatePolygon	New
Creates a polygon with the specified vertices as the ccSurface in the specified cellview and on the specified layer-purpose pair.	

ICADVM18.1 ISR8

What’s New and Enhanced

Features

Virtuoso Photonics Solution: Electronic-Photonics Design Environment	New
The Cadence® Virtuoso® design framework now supports an electronic-photonics co-design solution called the Virtuoso® Photonics® Solution, which can be used for designing hybrid systems. Leveraging the core Virtuoso design framework capabilities, the Virtuoso Photonics Solution enables electronic designers to design and simulate for optical signals within the familiar Virtuoso environment. The Virtuoso Photonics Solution is supported in the advanced nodes Virtuoso® Layout® Suite EXL (Layout EXL).	

SKILL Functions

ccGetPolyCurveMinRadius	New
Returns the minimum curvature radius of a polyCurve, ignoring any tangent discontinuity points.	
ccWideConnector	New
Creates specification of a path around the specified connector centerline with the width specified by a cubic spline function.	
ccCreateLineSegment	New
Creates a ccCurve that represents a straight line segment connecting two specified points.	
ccCopyFig	New
Copies the specified ccShape to the specified cellview and layer-purpose pair using the specified transform.	
ccMoveFig	New
Moves the specified ccShape to the specified cellview and layer-purpose pair using the specified transform.	

ccOffsetPolyCurve	New
Returns a copy of the specified polyCurve that is expanded by the specified offset value.	
ccRenameFacet	New
Returns a copy of the specified facet with a different name.	

Environment Variable

`generateSchematicMfactor`	New
Creates mfactors for layout instances and dummies during a *Generate Schematic From Layout* run.	

ICADVM18.1 ISR5

What’s New and Enhanced

SKILL Functions

ccCreateSurfaceFromPath	New
Creates a ccSurface object from the boundary of the given ccPath.	
ccCubicSplineConnector	New
Returns a polyCurve representing a cubic spline function in the form where y is a function of x through the given list of points (x,y) with strictly increasing coordinates x.	
ccGetCurvePoint	New
Returns a point on a curve or a polyCurve segment at the specified value of the curve function parameter.	
ccGetCurveDerivative	New
Returns a derivative of a curve or a polyCurve segment at the specified value of the curve function parameter.	

ICADVM18.1 ISR3

What’s New and Enhanced

Features

Curvy Core Functions	New
Curvy shapes are building blocks of photonic devices such as ring resonators, waveguide bends, and Bragg gratings. You can use these SKILL functions to work with the curvy core functionality of photonics.	

SKILL Functions

phoCompose	New
Composes the given list of waveguides into a composite of the specified master.	
phoComputeModeProps	New
Computes the path length and average values for the selected mode properties along the center line of the given path.	
phoTechDeleteModePropDataTable	New
Deletes the modeProp table for the specified waveguideDef from the given technology database.	
phoTechGetModePropDataTableParamValues	New
Returns the list of values specified in the modeProp table for the given parameter.	
phoTechHasModePropDataTable	New
Checks whether the given technology database contains a modeProp table for the given waveguideDef.	
phoTechLoadModePropDataTable	New
Creates a region boundary.	

Part 6: Routers

Virtuoso Space-based Router

This chapter provides a high-level overview of the new and enhanced features in the ICADVM18.1 and ICADVM20.1 releases.

Related Documentation

- *Virtuoso Space-based Router User Guide*
- *Virtuoso Layout Suite SKILL Reference*
- *Virtuoso Technology Data ASCII Files Reference*
- *Virtuoso Technology Data Constraints Reference*

ICADVM18.1 ISR12

What's New and Enhanced

Environment Variables

db.enable_effective_width_on_blockages	New
Specifies a list of cell types for a cell or a design. When set, it applies effectiveWidth on oaLayerBlockages on cells or designs of the specified cell types provided the blockages do not have the effectiveWidth or spacing attribute already set.	
db.enable_min_spacing_on_blockages	New
Specifies a list of cell types for a design. When set, it applies minimum spacing on oaLayerBlockages on cells or designs of the specified cell types provided the blockages do not have the effectiveWidth or minimum spacing already set.	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

Keeping Manual Routes	New
A new option, *Keep Manual Routes*, has been added to the Virtuoso Space-based Router Options form and the *Delete Routing* options in the Wire Assistant. Using this option, you can retain the routes that are created by interactive and assisted routing and delete only the routes that are created by automatic routing.	

Environment Variables

highlightTrunkHaloType	New
Lets you specify the halo effect around a trunk. The default value is `plain`.	
highlightTrunkHaloWidth	New
Lets you specify the thickness of the halo that is added around a trunk. The default value is `thin`.	

ICADVM18.1 ISR9

What’s New and Enhanced

Features

Keeping Fig Group Instances	New
A new option, *Keep Fig Groups Shapes*, has been added to the Virtuoso Space-based Router Options form and the *Delete Routing* options in the Wire Assistant. Using this option, you can retain the fig group shapes when deleting the routing topology. You can also use the `deleteRoutingKeepFigGroupShapes` environment variable to keep or delete the fig group shapes.	
Switching Between Pin Connections	New
A new option, *Allow One Connection*, has been added to the Virtuoso Space-based Router Options form. Using this option, you can specify the number of connections for congestion-aware pin optimization. You can also use the `pinOptOneConnectionPerSide` environment variable to keep or delete the fig group shapes.	

Documentation Updates

Congestion Analysis Environment Variables	New

The following congestion analysis environment variables have now been documented.

- cmapAnalyzeIncludeBlockage
- cmapAnalyzeMode
- cmapHideAnalyzedDataInFilter
- cmapHiLiteAnalyzedBrightness
- cmapHiLiteDimAnalyzedBrightness
- cmapHiLiteDimFilteredBrightness
- cmapHiLiteFilteredBrightness
- cmapHiLiteGlobalPathWidthPercent
- cmapHiLiteSelectedBrightness
- cmapHiLiteSpecGlobalPathWidth
- cmapHistogramRanges
- cmapMultiThreads
- cmapNumTracksPerCell
- cmapUserDefinedNumTracks

ICADVM18.1 ISR8

What’s New and Enhanced

Features

Congestion Analysis Task Assistant - A Quick Help	New
A new task-based quick help functionality, Congestion Analysis Task Assistant, has been introduced for the Congestion Analysis assistant. You access this quick help by clicking the () button available in the Congestion Analysis assistant toolbar. The Task Assistant provides task oriented information to get you started with Congestion Analysis.	

ICADVM18.1 ISR7

What’s New and Enhanced

Features

Tree Route Task Assistant - A Quick Help	New
A new task-based quick help functionality, Tree Route Task Assistant, has been introduced for the Tree Route flow. You access this quick help by clicking the () button available in the Wire Assistant and in the Tree Route subform in the Virtuoso Space-based Router Options form. The Task Assistant provides task oriented information to get you started with the Tree Route flow.	

Environment Variables

`vsrFlow`	New
Lets you specify the flow command in the `vsrFinal.il` file. By default, it is null.	

ICADVM18.1 ISR6

What’s New and Enhanced

Tcl Commands

fix_diff_track_errors	New
Gathers all shapes that violate the EOL rules and fixes these violations by deleting and re-routing these shapes.	

Environment Variables

db.ignore_routing_grid_check_for_via_metal_layers	New
Ignores the routing grid checks for metal layers with via shapes if the corresponding cut shape is enclosed by a non-via metal.	
db.proute_tap_to_depth	New
Specifies a value to insert vias on the power ground grid mesh shapes that are not at the top-level.	

ICADVM18.1 ISR5

What's New and Enhanced

Features

Scenic Ratio Capabilities	New
For easier visualization of critical scenic nets to help drive global biasing, the *Scenic Net* column is now available in the Navigator assistant. The scenic net ratio helps in floorplanning and die size reduction so that it does not affect the performance of critical nets.	
Congestion Analysis Assistant	Enhanced
The following GUI enhancements have been made in the *Congestion Analysis* assistant toolbar options. ■ The *Show 0% Global Cells Only* option is now renamed as *Show Empty Global Cells Only* in the *Filter* drop-down list. ■ The *Include Blockage on Analysis* option is now renamed as *Show Unusable Global Cells* in the *Filter* drop-down list. ■ A new option *Show Unusable Global Cells* has been added to the *Congestion Analysis* subform in the Virtuoso Space-based Router Options form. **Related**: Toolbar, Displaying Empty Global Cells, Display Settings	

Environment Variables

cmapEnablePinOptimizeStep	New
Lets you run additional steps to optimize pins on soft-blocks and on opaque virtual hierarchies.	
cmapShowUnusedGCellMode	New
Enables the display of the global cells that are blocked and are essentially not available for any global paths or interactive routing.	

Constraints

oaMinViaClearance	Enhanced
The following parameter has been added to this constraint: ■ `overlapNotAllowed`: Specifies whether to show violations for overlapped or stacked vias when the minimum spacing between via cuts on the specified layers is `0`.	

ICADVM18.1 ISR4

What’s New and Enhanced

Features

Trunk Generation	New
The new *Generate Trunks* routing step in the Pin to Trunk flow lets you customize the trunks and automatically create them. An extensive list of options is available in the *Trunk* subform that helps you to customize the trunk generation. **Related**: Using Automatic Trunk Generation in Pin to Trunk Route Flow	
Trunk Mesh Routing	New
The new *Trunk Mesh Routing* mode in the Pin to Trunk flow lets you customize the way meshes are routed. To customize the Trunk Mesh routing options, a new Trunk Mesh Routing Configuration form is available that lets you specify the way each layer is to be routed in the mesh. **Related**: Using Trunk Mesh Routing in the Pin to Trunk Route Flow, Trunk Mesh Routing	
Tree Route Flow	New
New Tree Route Flow lets you automatically route advanced node device-level designs with a minimum set of options. In addition, Tree Router lets you automatically identify and route mesh, ring (tap cell rings), and tree structures in the design. **Related**: Using Tree Routing, Tree Route	
Working with Structured Routing	New
The automated structured routing for device-level using Virtuoso Space-Based Router is based on more automation and less GUI options to be specified for routing the design. The automated structured routing uses one of the following features. ■ Using Automatic Trunk Generation in Pin to Trunk Route Flow ■ Using Trunk Mesh Routing in the Pin to Trunk Route Flow ■ Using Tree Routing	

ICADVM18.1 ISR3

What’s New and Enhanced

Tcl Commands

`delete routing`	Enhanced
The following argument has been added to this command: ■ `-partition_delete_type`: The option is ignored unless there is a concurrent design partition active.	

ICADVM18.1 ISR2

What's New and Enhanced

Environment Variables

db.load prboundary shape as boundary	New
Identifies the layer-purpose pair to be used for creating prBoundary in lower-level (non-top) designs, provided they do not have the `oaPRBoundary` object.	
db.prboundary_layer_name	New
Specifies the layer name for the prBoundary shape.	
db.prboundary_purpose_name	New
Specifies the purpose name for the prBoundary shape.	

Documentation Updates

Removed the documentation of the `minDensity` and `maxDensity` constraints from the Technology file chapter of the Virtuoso Space-based Router user guide. This is because the two constraints are not supported by automatic routing.

ICADVM18.1 ISR1

What's New and Enhanced

Features

Enhancements in Congestion Analysis Assistant	New
The following GUI enhancements have been made in the Congestion Analysis assistant. ■ Columns in the *Global Cell Track Utilization* table in the Congestion Analysis assistant now have a sorting capability to help you analyze the displayed data. ■ The image for the *Filter* icon in the Congestion Analysis toolbar has been enhanced. ■ A new *Congestion Map Visible* option is now available in the *Filter* drop-down list to toggle the display of the congestion map. ■ A new option *Show 0% Global Cells Only* is now available in the *Filter* drop-down list, which displays all global cells that have 0% congestion in the heatmap and histogram. ■ Global Bias Setup form now has a Green (+) symbol for positive bias setting. ■ Global Bias Setup form now has Red (-) for negative bias setting. **Related**: Congestion Analysis Forms, Using Global Bias Constraint, Toolbar	
Improved User Experience	Enhanced

The GUI options in the *Automatic* section of the Wire Assistant have been enhanced for an improved experience while using the Wire Assistant.

- The options change for different route flows.
  - When the routing flow is set to *Pin to Trunk*, the relevant and useful Pin to Trunk options are now displayed.
  - The *Wrong Way Tax* and the *Remove Pre-Route Dangles* options have been hidden when the routing flow is selected as *Custom / Digital (MST)* or *Pin to Aligned Pin*.
  - The *Prefer Violations to Opens* option is now deselected by default.
- The routing steps in the *Pin to Trunk* route flow have been renamed in the VSR Options form and Wire Assistant.
- The Topology Style field has been renamed to *Route Flow*.
- The icons for the *Design Style* options have been improved.
- The *Minimum Spanning Tree* routing flow has been renamed to *Custom / Digital (MST)*.
- The Wire Assistant Visibility form has been updated.
  - The *Route Flow* option is now always available.
  - The *Custom / Digital (MST)* options are now grouped and available in a subheading.
  - New subheadings have been added to group the options available for *Pin to Aligned Pin* and *Pin to Trunk* route flow.

In addition, tooltips for all the renamed and enhanced options have been appropriately updated.

Note: The GUI enhancements corresponding to the Wire Assistant have also been made in the Virtuoso Space-based Router Options form.

Related: Specifying Routing Flow Options

Environment Variables

`gapFillPurpose`	New
Specifies the purpose of all modified or newly created `SADPFillPatch` shapes when synchronized back to OpenAccess.	

trimBridgePurpose	New
Specifies the purpose of all modified or newly created `bridgeMetal` shapes when synchronized back to OpenAccess.	
removePrerouteDangles	Enhanced
To improve the usability, the default value of the variable has changed from `t` to `nil`. Also, the related GUI check box *Remove Pre-Route Dangles* is now deselected by default.	

ICADVM18.1 Base Release

What's New and Enhanced

Features

Congestion Analysis Assistant - New Congestion Analysis Tool	New
In ICADVM18.1, Virtuoso introduces a new assistant for congestion analysis through the new Virtuoso Layout Suite EXL layout editor (Layout EXL). which lets you run comprehensive checks and ensure that accurate results are extracted and displayed. In addition, it also lets you view and analyze the routing density of a design from initial floorplanning to detailed placement and routing. **Related**: Congestion Analysis Forms	
Virtuoso Space-based Router Options Form	Enhanced
In ICADVM18.1, a new subform for Congestion Analysis options has been added to the Virtuoso Space-based Router Options form.	
Interrupting a Routing Run	Enhanced
The capabilities to interrupt a routing run that is taking too long has been enhanced. New statistics that have been completed between the two actions are now displayed in the User Interrupt Detected form.	

SKILL Functions

`rdeRemoveChamferFill`	New
Removes all the right-angled triangle shapes of purposes specified by `outputPurpose` and `materialRemovalPurpose`.	
`vsrLoadPreset`	Enhanced
Supports a new `?path` argument that is set for the location of the preset file.	
`vsrSavePreset`	Enhanced
Supports a new `?path` argument that is set for the location of the preset file.	
`vsrDeletePreset`	Enhanced
Supports a new `?path` argument that is set for the location of the preset file.	

Tcl Commands

proute_stripes	Enhanced
The following argument has been added to this command: ■ `-ignore_same_net_shape`: Ignores pre-existing same net shapes.	

Documentation Updates

Reorganization of Content	New
■ The *Virtuoso Space-based Router User Guide* has been split into two user guides. The content for the interactive and assisted routing commands has been moved out of the *Virtuoso Space-based Router User Guide*. ■ The *Virtuoso Space-based Router Quick Start* guide is no longer available. Information from this guide has been added to the appropriate sections of the *Virtuoso Space-based Router User Guide* and *Virtuoso Interactive and Assisted Routing User Guide*. ■ The batch checking content has been moved from the *Virtuoso Space-based Router User Guide* and has been included in the Virtuoso Layout Suite documentation.	

Virtuoso Interactive and Assisted Routing

This chapter provides a high-level overview of the new and enhanced features in the ICADVM18.1 and ICADVM20.1 releases.

Related Documentation

- *Virtuoso Interactive and Assisted Routing User Guide*
- *Virtuoso Layout Suite SKILL Reference*
- *Virtuoso Technology Data ASCII Files Reference*

- *Virtuoso Technology Data Constraints Reference*

ICADVM20.1 Base Release

What’s New and Enhanced

Features

Automatic Snapping in Stretch for Package Routing	New
In Virtuoso RF, the Stretch command now supports the automatic snapping of packaging wires in diagonal, orthogonal, and anyAngle modes. Automatic snapping while stretching a packaging wire helps create a correct DRC connection.	

Constraints

`minCutRoutingSpacing`	Enhanced
Support for the following parameters has been added for the *Create Bus* and *Create Stranded Wire* commands. ■ `cutClass` ■ `exceptSameNet` ■ `enclosingLayer` ■ `anyOppositeExtension` ■ `insideLayers`, `outsideLayers` ■ `insidePurposes`, `outsidePurposes`	
`minCornerToCornerDistance`	Enhanced
Support for the `extensionLength` and `oppositeDirection` parameters has been added for the *Create Wire* command.	

ICADVM18.1 ISR13

What’s New and Enhanced

Features

Renaming of Create Geometric Wire Form	Changed
In Virtuoso RF, the Create Geometric Wire form has been renamed to Create Packaging Wire.	

Constraints

`allowedSpacingRanges`	Enhanced
The support of the `exceptWidthEdge` parameter has been added to this constraint for the *Create Bus* and *Create Stranded Wire* commands.	

ICADVM18.1 ISR12

What’s New and Enhanced

Constraints

minViaSpacing	Enhanced
The support of the following parameters has been added to this constraint for the interactive routing commands. ■ parallelRunLengthRange: Measures the parallel run length orthogonal to the direction parameter, when specified. ■ viaConnectivityType: Applies to via connectivity type. ■ exceptViaConnectivityType: Applies to via cuts with certain connectivity types.	

ICADVM18.1 ISR11

What's New and Enhanced

Features

Crossing Region Boundaries	New
A new option, *Allow Crossed-region Off Track Wires*, has been added to the *Use Width Spacing Pattern* menu of the *Create Wire* context-sensitive menu. You can use the new option to ignore the region boundary and continue routing across regions.	
Display of Alternate Colors for snapPatternDef Tracks	New
The Create Wire command now displays alternate colors for odd and even tracks.	
Support of cdsGenVia for Different Purposes	New
The interactive routing commands support `cdsGenVia` for different purposes other than `drawing` for top or bottom layers.	

Environment Variables

`blockageAvoidanceSearchThreshold`	New
Controls the threshold value for the search limit for blockage avoidance.	
`touchBlockageMode`	New
Places slots within the regions that are partially covered by the blockage. The regions that are fully covered by the blockage are not slotted.	

Removed Features

During interactive routing, you can no longer switch between various width spacing patterns for a particular metal layer in the global grid or the selected region. You can switch between width spacing patterns using the Track Pattern Assistant. For more information, see Editing Width Spacing Pattern Regions.

ICADVM18.1 ISR10

What's New and Enhanced

Features

Support for the Push-and-Shove Feature in Interactive Routing	New
A new command, *Push And Shove*, has been added to the *Create Wire* and Stretch context-sensitive menus. You can use the push-and-shove options to reroute a wire and fix violations. **Related**: Introduction to the Push-and-Shove Feature **Environment Variables**: `pushAndShoveMode`, `pushAndShoveJumpWire`, `pushAndShoveAllowJog`, `pushAndShovePushVias`, `pushAndShoveCheckerStopLevel`	
Slotting Licensing	Enhanced
The slotting license depends on the layout tier and requires specific number of tokens to access the feature. For more information, see Table 1-6 Tokens per Feature (VLS GXL) in the *Virtuoso Software and Licensing User Guide*.	
Behavior of Ctrl and Shift Keys	Enhanced
The behavior of `Ctrl` and `Shift` keys for smart snapping, layer tapping, and via insertion features has been enhanced. You can use different combinations of the two keys.	

ICADVM18.1 ISR9

What's New and Enhanced

Features

Support for Must-Connect Pins	New
The *Create Wire* command now supports the must-connect pin connectivity model. **Related**: Working with Pin Connectivity Models	
Improved Metal Density Functionality	New
The following enhancements have been made to improve slotting capabilities. ■ A new command, *Check Slot Consistency*, has been added to the *Density Analysis* button of the *Metal Density* toolbar. The command options can be used to check and locate the floating shapes. The *Density Analysis* menu displays the following new options. ❑ *Check Slot Consistency All* ❑ *Check Slot Consistency By Area* ❑ *Check Slot Consistency Current Window* ■ A new tab, *Slot Consistency*, has been added to the Metal Density Options form. This tab consists of the options that can be specified to check floating slot shapes. **Related**: List of Slotting Environment Variables, Metal Density Options Form, Performing Metal Density Checks	

ICADVM18.1 ISR8

What's New and Enhanced

Features

Supporting pathSegs with Different WSPs Across Regions	New
Interactive routing provides an environment variable `allowOffTrackForCrossRegion`, which when set to `t` provides the flexibility to cross the region boundaries even when the WSPs (and allowed WSPs) do not match. This is one step further than the automatic switching of active pattern across regions.	

Environment Variables

`viaMaxNumCuts`	New
Specifies the maximum number of cuts that can be added to a via.	
`allowOffTrackForCrossRegion`	New
Lets a wire cross regions regardless of it being on track or off track. **Related**: Supporting pathSegs with Different WSPs Across Regions	

ICADVM18.1 ISR7

What’s New and Enhanced

Environment Variables

areaDelimitedMode	New
Defines different ways of finding shapes. It considers the shapes that are partially enclosed within the area and lets you create, delete, and check slotting shapes based on the area.	

Constraints

minHoleArea	New
This constraint is now supported by the *Create Stranded Wire* command.	

ICADVM18.1 ISR6

What's New and Enhanced

Features

Slotting now available only in Layout EXL	New
The slotting feature and the related *Metal Density* toolbar are now available only in ICADVM18.1 Layout EXL.	
Modified Use Color Mode Option	Enhanced
The *Use Color Mode – cycle* option in the *Create Bus* and *Create Stranded Wire* context-sensitive menu has been replaced with *Use Color Mode – optimized*. The *optimized* option is also the default coloring mode. The `weStrandedColorMode`, `weBusColorMode` environment variables have also been updated for this change. **Related**: Specifying the Bus Color Mode, Specifying the Color Mode in Stranded Wire	

SKILL Functions

`weGetAdjustedWidthForTracks`	New
Returns the Width Spacing Pattern (WSP) track information.	

Constraints

`minCornerSpacing`	Enhanced
This constraint is now supported by the *Create Bus* command.	

ICADVM18.1 ISR5

What's New and Enhanced

Features

Displaying Hint Box for Interactive Routing Commands	New
The *Interactive Routing Bindkey* hint box is now automatically displayed for the *Create Wire* and *Create Bus* commands (only in Layout EXL). To control the display of the hint box for the interactive routing commands, following enhancements have been made: ■ A new option *Show Hints* is now available in the *Wire Editing* section of the Layout Editor Options form. ■ A new environment variable `weShowHints` has been introduced. By default, the environment variable is set to `t`.	
Locking Unselected Vias	New
While stretching or splitting a wire, you can now control the change in the position of a via. The following enhancements have been made for this: ■ A new environment variable `lockUnselectedVias` has been introduced to control the change in position of the unselected vias that are locked. ■ A new check box *Lock Unselected Vias* has been added to the Split and Stretch Options form. This new option has also been added to the *Keep Wires Connected To* context-sensitive menu of the Split and Stretch commands.	
Multiple Slot Specification Cells	New
To consider multiple contexts, capability to view multiple specification cell entries in the Metal Density Options form has been added. The following enhancements have been made for this: ■ You can define a list of contexts for slotting using the `sltSetContextSpecificationCells` SKILL function. ■ The two environment variables that let you use the defined context are, `useOneSlotSpecCell` and `slottingContext`. ■ The *Context* field has been added to the Metal Density Options form when the `useOneSlotSpecCell` and `slottingContext` environment variables are specified.	

Environment Variables

patternGravityLayerTransitionType	New
Controls the dynamic display of the preferred layers on WSP tracks. **Related**: weCyclePatternGravityLayerTransitionType	
useOverlapRegionForVia	Enhanced
To improve via consistency with automatic routing, the default value of the variable has changed from `nil` to `t`.	

SKILL Functions

weHiInteractiveRouting	New
Maps the function to leHiCreateGeometricWire (paths mode) or the leHiCreateWire SKILL function depending on the technology file fabric type and the active application.	
weHiCycleViaDefUp	New
Cycles up the via definition for the pending vias, while using the *Create Wire*, *Create Stranded Wire*, or the *Create Bus* command.	
weHiCycleViaDefDown	New
Cycles down the via definition for the pending vias, while using the *Create Wire*, *Create Stranded Wire*, or the *Create Bus* command.	
weCyclePatternGravityLayerTransitionType	New
Used to cycle the patternGravityLayerTransitionType environment variable.	

Removed Features

- The following options have been removed from the *Wire Editing* section of the Layout Editor Options form.
  - *Instance Pins Patterns*
  - *Blockage: Use Minimum Width*

ICADVM18.1 ISR4

What’s New and Enhanced

Constraints

minVoltageSpacing (One layer)	New
This constraint is now supported by the *Create Bus* command.	
minVoltageExtension	New
This constraint is now supported by the interactive routing commands.	

ICADVM18.1 ISR3

What’s New and Enhanced

Environment Variables

hideConvertToPolygon	New
Hides the options to convert the original shapes covered by the slot shapes into slotted polygons.	
hideReconstruct	New
Hides the options to reconstruct a slotted polygon to a regular polygon.	
minPolygonEdgeLength	New
Truncates the slot with edges inferior to the value passed into the environment variable.	

Constraints

minSpacing (Two Layers)	New
This constraint is now supported by the *Create Bus* command.	

ICADVM18.1 ISR2

What’s New and Enhanced

Environment Variables

`weAreaBasedRulesEnabled`	Enhanced
The area-based rule support is now enabled by default for interactive routing commands. Therefore, the default value of the variable has changed from `nil` to `t`.	

Documentation Updates

Removed the documentation of the `minDensity` and `maxDensity` constraints from the Technology file chapter of the Virtuoso Interactive and Assisted Routing user guide. This is because the two constraints are not supported by the interactive routing commands, such as Create Wire and Create Bus.

ICADVM18.1 ISR1

What’s New and Enhanced

Features

Supporting Package Constraints	New
For VRF, interactive routing now supports three package constraints: pkgMaxLineWidth, pkgMinLineWidth, and pkgMinSpacing.	
Improved Metal Density Functionality	New

The following enhancements have been made to improve the slotting capabilities.

- A new *Tools* icon has been added to the *Metal Density* toolbar. When you click this icon, the menu displays the following new commands.
  - Delete Slots
  - Convert Slots
  - Reconstruct Slots
- A new *Slotting* button has been added to the Metal Density Options form to inform you about the validity of the parameters setting. The color of the *Slotting* button (Green, Yellow, or Red) represents whether or not slotting can be done. Clicking the *Slotting* button displays the status of slotting parameters.
- Environment Variables can now be used to predefine all fields in the Metal Density Options form.
  - **LAYERS**

    `usePaletteLayers`, `material`, `metalDensityLayers`
  - **SLOT GEOMETRY**

    `slotLength`, `slotWidth`, `lSpacing`, `wSpacing`, `slotToEdge`, `enableSlotSpecCell`, `slotSpecCell`, `slotOrientationMode`
  - **SLOTTING STYLE**

    `widthThreshold`, `lengthWidthRatio`, `slotStaggered`, `slotVia`, `excludePins`, `densityAwareMode`, `directionalSlotPattern`, `turnSlotPattern`

Related: List of Slotting Environment Variables, Metal Density Options Form, Performing Metal Density Checks

ICADVM18.1 Base Release

What's New and Enhanced

Features

Metal Density Toolbar	New
A new toolbar has been introduced to address and manage metal density issues, such as density checks, thin or local analysis, and fixing density errors through slotting. Using this toolbar, you can perform DRC correct slotting, `maxDensity` checks, and analyze any kind of shapes. **Related**: List of Slotting Environment Variables, Metal Density Options Form	
Support of VRF for Interactive Routing	New
The interactive routing commands have been enhanced to support curved paths that are available in cellviews along with package constraints.	
Using Halo Settings	New
New GUI options are available in the Layout Editor Options form to generate layer halos for collinear wire elements instead of one halo. **Related**: weHalo, weHaloDashStyle, weHaloTrueColor Layout Editor OptionsForm	
Supporting End-to-End Spacing for Colored Shapes	New
Color-specific end-of-line spacing notification is now supported in interactive routing. The *Create Wire* command now provides four types of color-aware spacing notifications instead of two (`sameMask`/`diffMask`).	
Changing Via Cut Color	New
While using the interactive routing commands, such as *Create Wire*, *Create Bus*, and *Create Stranded Wire*, you can now change the via cut color. To control the via cut color feature, two new environment variables `weViaCutColorPatterns` and `enableCutColorSupport` have been added. **Related:** `weCycleCutColorVia`	
Supporting End-to-End Spacing for Colored Shapes	New
Color-specific end-of-line spacing notification is now supported in interactive routing. The *Create Wire* command now provides four types of color-aware spacing notifications instead of two (`sameMask`/`diffMask`).	

Removed Features

- The support for the following interactive routing dormant commands and features has been discontinued. The documentation for these commands and features has also been removed.
  - Guided Routing command.
  - Differential pair interactive routing feature
  - Symmetry interactive routing feature
  - Push command
- The `autoReshape` environment variable is now private and the documentation for it has been removed.
- The `weGuidedRouteTopologyMode` environment variable is now private and the documentation for it has been removed.

Documentation Updates

Virtuoso Interactive and Assisted Routing User Guide	New
The Virtuoso Space-based Router user guide has been split into two user guides. The content for the interactive and assisted routing commands has been moved out of the Virtuoso Space-based Router User Guide. A new user guide has been included for the Interactive and assisted commands.	
Reorganized Virtuoso Space-based Router Quick Start Content	New
The *Virtuoso Space-based Router Quick Start* guide is no longer available. Information from this guide has been added to the appropriate sections of the *Virtuoso Space-based Router User Guide* and *Virtuoso Interactive and Assisted Routing User Guide*.	

Virtuoso Simulation Driven Interactive Routing

This chapter provides a high-level overview of the new and enhanced features in the ICADVM18.1 and ICADVM20.1 releases.

Related Documentation

- *Virtuoso Simulation Driven Interactive Routing User Guide*
- *Virtuoso Electrically Aware Design Flow Guide*
- *Virtuoso Interactive and Assisted Routing user Guide*

ICADVM20.1 Base Release

What’s New and Enhanced

Feature

Support for Over-Device Routing	New
Simulation driven interactive routing now supports over-device routing. In this, when you route over the device, the trunk is automatically connected to the device pins by adding vias.	

Removed Features

- The `weSdrShowToolbarAtStartup` environment variable is no longer supported, and the related documentation has been removed.

ICADVM18.1 ISR8

What’s New and Enhanced

Feature

Generating Dataset from DI Information	New
A new button, *Import DI as Dataset,* has been added to the SDR toolbar to create or update the EAD dataset with the design intent (`DI`) information.	

ICADVM18.1 ISR7

What’s New and Enhanced

Environment Variable

weSdrWidthMultiplier	New
The weSdrEMRatio environment variable is no longer available, and it has been replaced with weSdrWidthMultiplier. Use the new environment variable to automatically adjust wires according to the current, based on the scaling wire width.	

ICADVM18.1 ISR6

What’s New and Enhanced

Environment Variable

weSdrShowToolbarAtStartup	New
Enables you to show or hide the SDR toolbar when Layout EXL starts. **Related**: SDR Toolbar	

ICADVM18.1 ISR5

What’s New and Enhanced

Feature

Bindkey for Toggling Taper Mode	Enhanced
The bindkey for the *Toggle Taper Mode* command has been changed from `t` to `}`. The new bindkey is visible in the Interactive Routing Bindkey info balloon when the Create Wire command is started. **Related**: Bindkeys in Virtuoso Layout Suite	

ICADVM18.1 ISR4

What's New and Enhanced

Display of Sources and Sinks Map	Enhanced
The *Sources and Sinks Map* button on the SDR toolbar has been modified to let you change the source and sink map display to show either clusters or pins. A drop-down list has been provided on the *Sources and Sinks Map* icon to choose between clusters or instance pins. **Related**: Visualizing the Current Distribution Per Net	
Using J/JMax Width Multiplier	Enhanced
The *Scaling Factor* button on the SDR toolbar has been renamed to *J/JMax Width Multiplier*. Additionally, the use model for width scale factor has been simplified. The wire width is now multiplied by the value specified in that field to obtain the estimated width.	

ICADVM18.1 Base Release

What’s New and Enhanced

In ICADVM18.1, Virtuoso introduces a new interactive routing feature, Simulation Driven Routing, through the new Virtuoso Layout Suite EXL layout editor (Layout EXL). Virtuoso Simulation Driven Routing complements Virtuoso Electrically Aware Design flow by providing an environment for a layout designer to check if your layout is compliant with the datasets (most relevant information) extracted from the simulation. In addition, SDR addresses many of the electromigration (EM) and parasitic challenges of critical circuits and advanced-node designs earlier in the flow.

- **SDR Toolbar**

  To help the designers play around with the various Interactive Simulation Driven Routing features, a new toolbar has been introduced in Layout EXL. buttons have been included we are coming up with some additional SDR icons available on the EAD toolbar. The designers can use these icons to control all the interactive SDR settings.

- **Visual Assistance**

  To define and understand how the current has been estimated using SDR, the visual representations have been included to make SDR more intuitive and easy to understand. It is the key for the designer to understand and optimally use the application. The designers can see the current value used to estimate the wires and vias sizes as well as the targets that are used for estimation. Some of the visual representations that will be available in SDR are:

  - Estimated current is displayed next to the mouse pointer.
  - Blue circles are used to indicate the connected pins.
  - Yellow circles are used to indicate the pins to connect to.

Simulation Driven Interactive Routing offers the following benefits:

- Provides an easy way to visualize the net topology and current distribution per net before routing.
- Gives layout designers an interactive way to calculate the current into a wire, according to the topology.
- Supports auto sizing of wires and vias according to the estimated current.
- Provides an easy and flexible way to connect devices, according to the estimated current, as you route, especially for multi-finger devices.

Part 7: Translators

- Design Data Translators

Design Data Translators

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 base release and subsequent ISR releases.

Related Documentation

- *Virtuoso Design Data Translators Reference*
- *Design Data Translators SKILL Reference*

ICADVM20.1 Base Release

What's New and Enhanced

Features

Specifying Pcell Exclusion File During XStream Out Translation	New
The *Pcells Exclusion File* option has been added to the *General: Transformation Options* pane of the XStream In More Options form to specify the name of the exclusion file that contains a list of Pcells to be excluded during Pcell flattening.	
Specifying Pcell Exclusion File During XOasis Out Translation	New
The *Pcells Exclusion File* option has been added to the *General: Transformation Options* pane of the XOasis Out More Options form to specify the name of the exclusion file that contains a list of Pcells to be excluded during Pcell flattening.	
Support to Connect Floating Ports and Nets to A Cellview in Spice In	New
A new group box, *Cellview to Connect Floating Ports/Nets* is now available in the *Schematic Generation Options* tab of the Virtuoso Spice In form. Using the fields in this group box, you can connect the floating nets and ports in the schematic with any cellview. For information on these options, see the *Schematic Generation Options* section in the *Connectivity-to-Schematic User Guide.*	

ICADVM18.1 ISR13

What's New and Enhanced

Removed Feature

Removal of CDB to OpenAccess Translator	New
Support for the CDB to OpenAccess translator has been removed from Virtuoso in this release. As a result, you will not be able to access the software and documentation related to this feature in Virtuoso. Contact Cadence Customer Support to discuss how to complete any outstanding migration tasks.	

Features

DEF Out Supports Filtering of User-Defined Gray Shapes	New
The DEF Out translator now lets you filter user-defined gray shapes. The following options have been added to support this feature: ■ New command-line options added to the defout command: ❑ `noDefOnError` ❑ `errorOnGrayShapes` ❑ `errorOnUnlockedShapes` ■ New GUI options added to the DEF Out form: ❑ *Do Not Output DEF on Errors* ❑ *Output Errors for Gray Shapes* ❑ *Output Errors for Unlocked Color Shapes* ■ New arguments added to the `ldtrDefWriteOA` SKILL function: ❑ *`t_noDefOnError`* ❑ *`t_errorOnGrayShapes`* ❑ *`t_errorOnUnlockedShapes`*	

XStream In Translator Can Now Detect Overflow Cells	New
The following options have been added to the *General: Storage Options* pane of the XStream In More Options form to support detection of overflow cells during translation. ■ *Detect Overflow* Detects overflow cellviews during translation. ■ *Overflow Cell Name Suffix* Appends the specified string to the name of the overflow cells. ■ *Overflow View Name* Specifies a view name for the overflow cells.	
XOasis In Translator Can Now Detect Overflow Cells	New
The following options have been added to the *General: Storage Options* pane of the XOasis In More Options form to support detection of overflow cells during translation. ■ *Detect Overflow* Detects overflow cellviews during translation. ■ *Overflow Cell Name Suffix* Appends the specified string to the name of the overflow cells. ■ *Overflow View Name* Specifies a view name for the overflow cells.	
XOasis Translator Summary File Now Provides Information on Individual Advanced Objects Statistics	New
Individual Advanced Objects Statistics added to XOasis In and XOasis Out summary files list the Place and Route objects translated during XOasis translation.	

ICADVM18.1 ISR12

What's New and Enhanced

Features

CDL Out Task Assistant – A Quick Help	New
A new task-based quick help functionality, CDL Out Task Assistant, has been introduced for the CDL Out form. This assistant is accessible when you click the form help button. The Task Assistant provides information related to both analog and digital CDL netlisting using auCDL and CDL Out.	
XOasis Out Layer Map File Supports the Ignore LPP Qualifier	New
You can now you specify the `ignoreLPP` qualifier for XOasis layer mapping. When you specify this qualifier, the specified layer-purpose pairs (LPP) are ignored during mapping and the missing LPP warning is not displayed.	
XOasis Out Object Map File Supports the Qualifier Option	New
You can now specify a qualifier for object mapping in XOasis. The qualifier supports the value `OASISLayerName` for an OpenAccess object. You can specify this qualifier from the XOasis Out Object Map File Editor form or in the XOasis Object Map file.	

ICADVM18.1 ISR10

What's New and Enhanced

Features

Translating parametrized Tline cells by using the Allegro translator	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) The Allegro translator now preserves the instances of the parametrized `Tline` cells from the `rfTlineLib` library when they are exported to SiP from Virtuoso and then imported back.	
Prevent output of trim layers shapes into the SPECIALNETS section	Enhanced
DEF Out translator now lets you prevent trimmed floating shapes from being imported to the SPECIALNETS section. The following options have been added to the DEF Out translator: ■ New command-line option `skipTrimmedShapes` added to the defout command. ■ New GUI option *Do not Output Trimmed Floating Shapes into SPECIALNETS Section* added to the DEF Out form. ■ New argument *`g_skipTrimmedShapes`* added to the `ldtrDefWriteOA` SKILL function.	

ICADVM18.1 ISR9

What's New and Enhanced

Features

Specify OpenAccess map file in DEF In and DEF Out translators	Enhanced
DEF In and DEF Out translators have been enhanced to let you specify OpenAccess mapping file during translation. The following options have been added to the DEF In translator: ■ New command-line option `oaMapFile` added to the defin command. ■ New GUI option *OpenAccess Map File* added to the DEF In form. ■ New argument *`t_oaMapFile`* added to the `ldtrDefReadOA` SKILL function. The following options have been added to the DEF Out translator: ■ New command-line option `oaMapFile` added to the defout command. ■ New GUI option *OpenAccess Map File* added to the DEF Out form. ■ New argument *`t_oaMapFile`* added to the `ldtrDefWriteOA` SKILL function.	

ICADVM18.1 ISR8

What's New and Enhanced

Features

Support to detect vias with negative metal extension in XStream In	New
The following option has been added to the General: Others pane of the XStream In More Options form: ■ *Enable Negative Extension* When specified, XStream In detects vias with negative metal extension during translation.	
Support to detect vias with negative metal extension in XOasis In	New
The following option has been added to the General: Others pane of the XOasis In More Options form: ■ *Enable Negative Extension* When specified, XOasis In detects vias with negative metal extension during translation.	

ICADVM18.1 ISR7

What's New and Enhanced

Features

Support to specify user-defined SKILL file in DEF In and DEF Out translators	Enhanced
DEF In and DEF Out translators have been enhanced to let you specify a file that consists of user-defined SKILL routines. The following options have been added to the DEF In translator: ■ New command-line option *userSkillFile* added to the defin command. ■ New GUI option *User Skill File* added to the Virtuoso DEF In form. ■ New argument *t_userSkill* added to the ldtrDefReadOA SKILL function. The following options have been added to the DEF Out translator: ■ New command-line option userSkillFile added to the defout command. ■ New GUI option *User Skill File* added to the Virtuoso DEF Out form. ■ New argument *t_userSkill* added to the ldtrDefWriteOA SKILL function.	
Support to specify list of layers in DEF Out translator	Enhanced
DEF Out translator now lets you specify a list of layers used in the COMPONENTMASKSHIFT statement. The following options have been added to the DEF Out translator: ■ New command-line option *maskShiftLayers* added to the defout command. ■ New GUI option *Mask Shift Layer(s)* added to the Virtuoso DEF Out form. ■ New argument *t_maskShiftLayer* added to the ldtrDefWriteOA SKILL function.	

Support to generate GDS-compatible DEF file in DEF Out translator	Enhanced
DEF Out translator now lets you generate a DEF file with limits that make it compatible with GDS. The following options have been added to the DEF Out translator: ■ New command-line option `gdsCompatible` added to the defout command. ■ New GUI option *Generate GDS Compatible File* added to the Virtuoso DEF Out form. ■ New argument *`t_gdsCompatible`* added to the `ldtrDefWriteOA` SKILL function. **Note:** This feature is available in DEF Out translator from ICADVM18.1 ISR5.	

SKILL Functions

`acdlArtPrintIncludedNetlist`	New
Controls how an included netlist is included in the main netlist file. This function can be overridden by user settings. I	

ICADVM18.1 ISR6

What’s New and Enhanced

Features

Support for User-defined Functions in XStream Translator GUI	New
In the XStream In and XStream Out GUI, you can register or unregister user-defined callback functions for a specific predefined GUI event. These callback functions run at the occurrence of specific predefined XStream In or XStream Out GUI event, such as after translation is completed or *Cancel* or *Apply* button is clicked on the XStream In or XStream Out form.	
Support for User-defined Functions in XOasis Translator GUI	New
In the XOasis In and XOasis Out GUI, you can register or unregister user-defined callback functions for a specific predefined GUI event. These callback functions run at the occurrence of specific predefined XOasis In or XOasis Out GUI event, such as after translation is completed or *Cancel* or *Apply* button is clicked on the XOasis In or XOasis Out form.	

ICADVM18.1 ISR4

What's New and Enhanced

Features

XOasis Out Layer Map File Supports Additional Qualifiers	New
XOasis Out layer map file, now supports the following qualifiers: ■ pin Helps in defining a specific layer mapping for pin shape. When specified, the pin shapes are exported as per this qualifier. ■ OASISLayerName Writes a user-specified name to be associated with the given *layernum*:*datanum* in the OASIS file	
Support for Ignoring Missing Vias in XOasis Out	New
The following option has been added to the Report pane of the XOasis More Options form: ■ *Ignore Missing Vias* This option allows you to ignore missing via definitions during translation. By default, XOasis Out errors out.	
Support for Ignoring Missing Vias in XStream Out	New
The following option has been added to the Report pane of the XStream More Options form: ■ Ignore Missing Vias This option allows you to ignore missing via definitions during translation. By default, XStream Out errors out.	

Environment Variable

xoasAlwaysLookupMapFilesInTech	New
Specifies whether the XOasis In and XOasis Out GUI automatically opens the technology library including ITDB hierarchy to search for map files in the technology libraries. If the technology library is already open, setting this variable has no effect.	

ICADVM18.1 Base Release

What's New and Enhanced

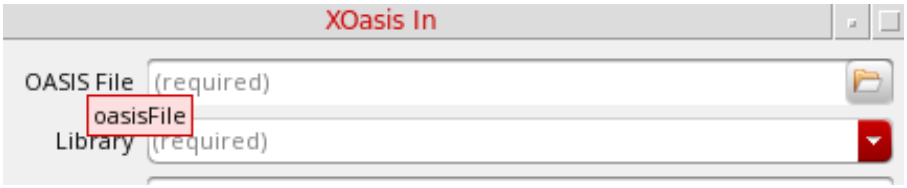
Feature

Support for Package Design Translation Using Allegro Translator	New
(ICADVM18.1 Virtuoso MultiTech Framework Only) The Allegro translator lets you import and export package designs from Allegro. While importing, the Allegro translator maps Allegro concepts into instances of Pcells in Virtuoso. These Pcell instances are mapped back to Allegro objects during export. You can use the Allegro translator to translate the following files: ■ Cadence SiP Layout (.sip) ■ Allegro Package Designer (.mcm) ■ Technology file with layer stack up and constraint information ■ Catalog of DRA ■ A board or package file created in Allegro PCB Layout (.brd)	

Graphical User Interface

New GUI for XOasis In and XOasis Out Translators	Enhanced
The GUI of XOasis In and XOasis Out translators has been enhanced to improve usability. Some of the new features include: ■ Only the most frequently used options are now available on the main form. Progressive disclosure has been used to hide the options that are used less often. ■ Additional options are available on the More Options form, which features a search-enabled tree view to help you locate the options you need. ❑ XOasis In More Options form ❑ XOasis Out More Options form	

- The corresponding command-line option is shown as a tooltip for each option in the form.



Part 8: Mixed-Signal Flows

Virtuoso Electrically Aware Design Flow

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 base release and subsequent ISR releases.

Related Documentation

- *Virtuoso Electrically Aware Design Flow Guide*
- *Virtuoso Parasitic Aware Design User Guide*

ICADVM18.1 ISR13

What’s New and Enhanced

Environment Variable

layoutEAD doAutoLayerMapping	New
Controls whether to automatically map a design layer to a layer in the EAD technology file if both the layers have the same name.	

ICADVM18.1 ISR7

What's New and Enhanced

Feature

Running EM Analysis on Power Grids	New
You can run EM analysis on power grids even when there are no devices connected to it. EAD places dummy devices in the layout and runs EM analysis, considering the values of current passing through these devices. You can choose whether you want to run the EM analysis on the original layout or on the scratch layout. If you do not want to modify the original layout, it is recommend that you run the analysis on the scratch layout. The following commands have been introduced to support this enhancement: ■ *EAD – Run grid analysis on scratch cellview*: Runs EM analysis in the scratch layout. ■ *EAD – Run Supply Grid Analysis (modifies layout)*: Runs EM analysis on supply grids in the original layout. ■ *Discard edits from grid analysis*: Discards the EM results and the changes made in the original or scratch layouts. ■ *Update grid in original layout*: Updates the original layout based on the changes made in the scratch layout.	

SKILL Functions

`eadDeleteScratchLib`	New
Deletes all data from the scratch library. The scratch library can be defined using the `scratchLib` environment variable.	
`eadGridAnalysisEnable`	New
Enables EM analysis on power grids in layout EAD.	
`eadPreEMSetUserStopCVList`	New
Sets all or the specified cells of user-specified libraries as stop views. The instances of the specified cells are not further descended during the pre-EM checks.	

Environment Variables

preEMDescendIntoPCell	New
Specifies whether to descend into a Pcell during the pre-EM checks.	
adjustedCurrentMsgRelTol	New
Specifies the relative tolerance limit to determine whether to print the adjusted current message if adjustment is not within the specified relative tolerance.	
scratchLib	New
Specifies the library in which the scratch layout is created when running EM analysis on power grids.	
ictemLayerMapFile	New
Specifies the path to EM Layer Map file that contains the layer mapping information.	

ICADVM18.1 ISR6

What's New and Enhanced

Features

iQuantus FS C Extraction	New
iQuantus Field Solver (iQuantus FS) engine is now integrated with EAD. You can use iQuantus FS to extract parasitic capacitances. It provides the signoff-level accuracy in the extraction results. A new drop-down list, *Engine*, is introduced to the *High Precision C Solver* section on the *Extraction* tab of the EAD Options form. To extract parasitic capacitances using iQuantus FS, select `quantusFS` from the *Engine* drop-down list.	
Support for Distributed Processing Jobs in EAD	New
You can now set up job policies and define the methods of how distributed processing jobs are submitted to the local or remote hosts. You can use these jobs to extract parasitics and to run EM analysis. The following enhancements have been made to support this feature: ■ A new section, *Job Setup*, has been added to the *Environment* tab in the *EAD Options* form. See the Environment section for more information. ■ A new form, *EAD Job Policy Editor*, has been introduced. You can use this form to set up, edit, and delete a job policy. See the Working with Job Policies section for more information. ■ The following commands have been added to the context-sensitive menu that opens when you right-click a net in the *Summary* pane of the EAD Browser: ❑ *Extract Parasitics With Policy*: See the Extracting Parasitics With a Policy section for more information. ❑ *Stop Job*: See the Stopping and Resubmitting a Job section for more information. ❑ *View Job Info*: See the Viewing Job Information section for more information. ❑ *Net Options*: See the Using the Distributed Processing Jobs section for more information.	

Environment Variables

layoutEAD defaultJobPolicy	New
Specifies the name of the default job policy to be used.	
layoutEAD qrcTechFileName	New
Specifies the name of the QRC technology file to be used by the Quantus high precision C engine. This file must be aligned with the EAD technology file.	
layoutEAD qrcLayerSetupFileName	New
Specifies the name of the QRC layer setup file to be used by the Quantus high precision C engine.	
layoutEAD qrcCapgenFileName	New
Specifies the name of the QRC capgen file to be used by the Quantus high precision C engine.	
layoutEAD hpcEngine	New
Specifies the name of the high precision C engine to be used by the Quantus high precision C engine.	

ICADVM18.1 ISR5

What’s New and Enhanced

SKILL Functions

eadCreateDataSetFromLayCSV	New
Creates a user-defined dataset, with the specified name, in the given layout cellview using the electrical current values from the specified layout CSV file.	
eadCreateDataSetFromSchCSV	New
Creates a user-defined dataset, with the specified name, in the given layout cellview using the electrical current values from the specified schematic CSV file.	

Environment Variables

setTermNodeAsReference	New
Specifies whether to set the top-level term as the voltage reference node when running EM analysis.	

ICADVM18.1 ISR4

Features

Support for iRCX-EM file as EM Data Source	New
You can now read EM data directly from the iRCX-EM file without the need to update the existing EAD technology file or to first convert the iRCX-EM file to ICT-EM file. When using an iRCX-EM file, ensure that the layer names in the EAD technology file and the given iRCX-EM file match. If the layer names in these files do not match, you must provide an iRCX-RC file along with the iRCX-EM file. The iRCX-RC file provides the missing layer mapping information. To specify the iRCX-EM and iRCX-RC files, select `iRCX-EM` from the *EM Data Source* drop-down list in the EAD Process Settings form, and then specify the path to the files in the *RCX-EM Data File* and *iRCX-RC Layer Map File* columns, respectively.	
Support for Multiple Current Injection Points in Wide Pins	New
By default, EAD considers center of the pin as the default current injection point. If the dataset mentions 0V as the voltage source for the pin, EAD recalculates the current flowing through the pins, and while doing that considers the resistance of the resistors to be zero. However, for wide pins, you can choose to consider multiple current injection points at specific points on the pins.	

SKILL Function

`eadCreateDataSetLayCSVTemplate`	New
Creates a CSV template file for a given layout cellview in a format that can later be used to fill in current values and to create a user-defined EAD dataset from it.	

ICADVM18.1 ISR2

What's New and Enhanced

Features

Exporting Parasitics in DSPF (Detailed Standard Parasitics File)	New
You can use the *Generate DSPF File* command provided in the EAD Browser toolbar to export the parasitic information displayed in the EAD Browser to a DSPF file and use that in Spectre simulation runs. You can specify the DSPF file name and location on the *DSPF* tab of the Parasitics & LDE Setup form. You can also set the default values for this form by using the environment variables given in the Environment Variables section below.	
Using Custom Variables in ICT Files	New
EAD now supports EM calculations that use custom variables in the ICT-EM file. If the EM calculations in the ICT-EM file are using certain variables that you would need to modify before running EM checks, you can register those variables with EAD by using the `eadRegisterCustomEMVariable` SKILL function. At any time before running EM checks, you can modify the values of the registered variables by using the `eadSetEMVariable` SKILL function and then run EM extraction to use the revised variable value for EM calculation.	
Creating a Copy of the Layout View in a Different Library	New
While working in the EAD workspace for Virtuoso Schematic Editor XL, accessing a layout in the copy mode creates its copy in the same library as that of the original layout. If you do not have write access to that location, you can create a copy in a different library by using the new *Copy Library* drop-down list.	

Environment Variables

`layoutEAD.gui netGroupsPath`	New
Specifies the path to the net group state .ini files for saving and reading the net group data. The net groups are saved in the <library-name>/<cell-name> directory in the specified path. By default, the net groups are saved in the .cadence directory.	
`layoutEAD.pex dspfFileName`	New

Specifies the default name to be used for the DSPF file in which parasitics are exported.	
`layoutEAD.pex dspfWriteInstances`	New
Specified whether to write instances in the DSPF file.	
`layoutEAD.pex dspfCapUnits`	New
Specifies the unit for capacitance values to be written in the DSPF file.	
`layoutEAD.pex dspfResUnits`	New
Specifies the unit for resistance values to be written in the DSPF file.	
`layoutEAD.pex dspfWriteSubNodes`	New
Specifies whether to write subnodes in the DSPF file.	
`layoutEAD.pex dspfSubNodeDelimiter`	New
Specifies the delimiter to be used while writing sub nodes in the DSPF file.	
`layoutEAD.pex dspfGroundCapPrefix`	New
Specifies the default prefix to be used for the ground capacitance values in the DSPF file.	
`layoutEAD.pex dspfCouplingCapPrefix`	New
Specifies the default prefix to be used for the coupling capacitance values in the DSPF file.	
`layoutEAD.pex dspfResistorPrefix`	New
Specifies the default prefix to be used for the resistance values in the DSPF file.	

SKILL Functions

eadRegisterCustomEMVariable	New
Registers a variable as a custom variable for the purpose of EM calculation.	
eadSetEMVariable	New
Sets the value of a custom variable defined by `eadRegisterCustomEMVariable`.	

ICADVM18.1 ISR1

Features

Pre-layout Electromigration (EM) Check	New
You can now run the pre-layout EM check (pre-EM check) on the specified layers of a Pcell. This helps in identifying and resolving EM violations even before creating the actual circuit layout. After analyzing the violations reported in the pre-EM check, you can make the suggested changes in the design, and verify the results by running the pre-EM checks again in ADE XL, ADE Explorer or ADE Assembler. Alternatively, you can run a what-if analysis in Virtuoso Schematic Editor by making changes in the setup, for example, changing the EM temperature or electrical data in the dataset. You can then run the EM checks again to analyze how these changes affect EM violations. To support this feature, following enhancements are made: ■ A new check box, *Pre EM Check*, has been added to the EAD Setup form. Use this check box to enable the feature. ■ A new command, *View PreEM Violations*, has been added in the *EAD* results view of ADE Assembler and ADE Explorer. Use this command to view the report of EM violations. ■ A new check box, *No Layout*, has been added to the EAD Browser assistant of the Virtuoso Schematic Editor. Use this check box when running pre-EM check in schematic without a layout. ■ A new command, *Run Pre-EM Checker*, has been added to the EAD Browser toolbar of Virtuoso Schematic Editor. Use this command to run pre-EM check in Virtuoso Schematic Editor.	

Environment Variables

`preEMCheckLayers`	New
Specifies the layer on which pre-EM check is performed.	
`preEMDetailReport`	New
Specifies whether to generate the detailed report of pre-EM analysis.	

preEMLogFile	New
Specifies the path where the log file containing the report of pre-EM analysis is saved.	
preEMViolThresholdPercent	New
Specifies the threshold value of J/Jmax ratio in percentage reaching which the pre-EM check starts reporting EM violations.	
clickClickUseNetCache	New
Specifies whether to process all the shapes found on a net while selecting points for the creation of point-to-point info balloons.	

ICADVM18.1 Base Release

What’s New and Enhanced

Feature

Availability of EAD in Virtuoso EXL	New
The EAD Browser and the commands used in the EAD flow have been moved to a new product, Virtuoso Layout EXL, and Virtuoso Layout EAD has been discontinued. Related enhancements: ■ You can use any one of the following licenses to use the EAD Browser: ❑ Virtuoso_Layout_Suite_EXL (95800) ❑ Virtuoso_Layout_Suite_EAD (95600) For more details, refer to Licensing Requirements. ■ You now use the EAD menu in Virtuoso Layout EXL to open EAD Browser. For more details, refer to Starting EAD Browser for a Layout in ICADVM18.1.	

Virtuoso Voltage Dependent Rules Flow

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 base release and subsequent ISRs.

Related Documentation

- *Virtuoso Voltage Dependent Rules Flow Guide*

ICADVM20.1 Base Release

What's New and Enhanced

Features

Managing Vsync Constraints in the Layout View	New
In Layout EXL, you can use the new VSync Constraints Visualizer form to create *Voltage Synced Nets* (vsync) constraints from the contents of a CSV file, list the vsync constraints currently present in the layout view, and delete those that are no longer required. See Defining and Checking Voltage Synced Nets for more information. **Form:** VSync Constraints Visualizer **SKILL APIs:** `vdrCreateVSyncConstraintsFromFile`, `vdrTransferVSyncConstraints`, `vdrVsyncVisualizerGUI`	

SKILL Functions

`vdrSetNetVoltageRange`	New
Sets net voltages in a specified list of cellviews. The voltages can either be specified directly as a list when the command is called or read from voltage information file.	

ICADVM18.1 ISR12

What's New and Enhanced

Features

Worst-Case Labels now Generated from Voltage Information File	Enhanced
If the same net is listed multiple times in a voltage information file, the generated labels now reflect the highest maximum and lowest minimum values for that net. In previous releases, only the last entry for any given net in the voltage information file was considered. See Generating Voltage Labels from a Voltage Information File for information.	
Verbose Mode when Generating Labels from a Voltage Information File	Enhanced
You can now generate labels from a voltage information file in verbose mode, which prints one message for each line in the file confirming the action taken for that entry. Verbose messages are printed in the CIW by default, but can also be saved in a separate log file. The `vdrCreateVoltageLabel` and `vdrCreateVoltageLabelEx` SKILL APIs have been enhanced with the optional *`g_verbose`* and *`t_logFile`* arguments to support this feature.	

ICADVM18.1 ISR10

What’s New and Enhanced

SKILL Function

Label Creation SKILL API with Keyed Optional Arguments	New
A new SKILL API called `vdrCreateVoltageLabelEx` has been added. This API provides the same functionality as the `vdrCreateVoltageLabel` API, but its optional arguments are called using keywords for convenience.	

ICADVM18.1 ISR9

What’s New and Enhanced

Features

Specifying the Marker Size in Marker Generation Flows	New
You can now specify the size of voltage markers created using any of the marker generation flows. See Generating Voltage Markers for Manually Entered Voltages for more information. Previously, this was possible only for label generation flows. You can specify the marker size interactively on the Voltage Dependent Rules form and procedurally when using the `vdrCreateVoltageMarkers` and `vdrCreateVoltageMarkersOnNets` SKILL APIs. You set the default size using the `vdrLabelHeight` environment variable, which now applies to all label and marker generation flows.	
Voltage Rounding in Marker Generation Flows	Enhanced
The rounding options for marker generation flows now round voltage values to two decimal places instead of one. This is consistent with the rounding applied in label generation flows. See Generating Voltage Markers for Manually Entered Voltages for more information about these rounding options.	

ICADVM18.1 ISR5

What’s New and Enhanced

Features

User Control of Vsync Shape Creation and Checking	New
Two new environment variables have been added to let you modify the vsync shape creation and checking layers that are defined in the process technology file. You can now temporarily override technology file settings to ensure that vsync shapes are created only for the desired layers and that these shapes can subsequently be checked by the VDR Sanity Checker. For more information, see `vdrVSyncCreateCheckLayer` and `vdrVSyncSanityCheckLayer` in the *Virtuoso Voltage Dependent Rules Flow Guide*.	

ICADVM18.1 ISR3

What’s New and Enhanced

Features

VDR Sanity Checker Checks only Constrained VDR Labels	Enhanced
The VDR Sanity Checker now recognizes and checks only constrained VDR labels drawn on layer-purpose pairs defined in a `voltageLabelMapping` constraint in the technology file. It no longer checks generic VDR labels created by Virtuoso when the technology file contains no LPP definitions for labels. See ■ Types of VDR Labels for general information about constrained VDR labels. ■ Sanity Checking Voltage Values in Constrained Labels for information about the constrained label sanity checker flow. **Note:** You can still use the VDR Sanity Checker to check the correctness and consistency of vsync shapes in the layout view.	

ICADVM18.1 ISR2

What's New and Enhanced

Features

VDR Sanity Checker	New
You can use the VDR sanity checker feature to check the voltage values tagged in the labels in the layout against the values stored in the schematic or layout net properties and report any discrepancies between the values. Differences greater than a specified tolerance are either captured in a user-specified report file or printed in the CIW. See Sanity Checking Voltage Values in Labels for more information. The form, SKILL APIs, and environment variables listed below have all been added or enhanced to support the feature. **Form:** VDR Sanity Checker **SKILL APIs:** `vdrRunSanityChecker`, `vdrSanityCheckerGUI` **Environment Variables:** `vdrSanityCheckerCheckAgainst`, `vdrSanityCheckerGenLogFile`, `vdrSanityCheckerLogFile`, `vdrSanityCheckerObjectType`, `vdrSanityCheckerTolerance`, `vdrSanityCheckerToleranceType`	
Defining and Checking Nets with Synchronized Voltage Values	New
Some advanced node processes feature the concept of *synchronized* (or *synced*) *nets*, for which voltage values must always transition in phase. The voltage difference that triggers a DRC violation for these synced nets is lower than in mature node processes. Virtuoso supports this enhanced check by allowing you to tag pairs of nets as *synced* using a *Voltage Synced Nets* constraint in the schematic, and to specify in the process technology file the layer-purpose pairs on which to draw vsync shapes for these nets in the layout view. See Defining and Checking Voltage Synced Nets for more information.	
Manual Editing of Labels in the Layout View	New
Alongside generic VDR labels, which can be generated only by Virtuoso, the system now supports a *constrained* label generation flow, which lets you create, copy, and edit labels directly in the layout canvas. You enable this flow by using the `vdrConstraintGroupName` environment variable to specify a constraint group containing a `voltageLabelMapping` constraint for at least one layer. See Types of VDR Labels for more information.	

Part 9: System Design Environment

Virtuoso MultiTech Framework

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso MultiTech Framework User Guide*

ICADVM20.1 Base Release

SKILL Functions

vmtLibImport	New
Creates a library containing technology and components from a specified `.sip` file. The library contains TILPs, footprints, and schematic symbols for dies, packages, SMDs, embedded components, and padstacks in a `.sip` file.	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

Creating Extracted Views in the Virtuoso MultiTech Framework	New
The Virtuoso MultiTech Framework now supports creating extracted views and backannotating parasitic models to the golden schematic. Models also get highlighted in the schematic after backannotation. Creating extracted views with stitched s-parameter models makes designs realistic and leads to better simulation results.	

SKILL Functions

vmtcsvCreateComponentCellViewsFromCsv	New
Imports a CSV file, creates CDF parameters for the column values in the CSV file, and creates symbol, TILP, and other simulation views.	
vmtcsvInstallCsvFile	New
Installs the CSV file into the destination cellview and creates the CDF parameters corresponding to the columns in the CSV file.	

ICADVM18.1 ISR10

What’s New and Enhanced

Features

Virtuoso MultiTech Framework	New
The Virtuoso MultiTech Framework enables various Virtuoso Schematic–driven layout generation flows. The key features include: ■ The layout generation process, *Generate All From Source*, is used to create the initial layout. ■ Edits in the schematic can be propagated to the layout through an update flow called *Update Components and Nets*. ■ Changes made in the SiP layout can be checked against the schematic to provide the in-design layout-versus-schematic (LVS) check that ensures that the layout never departs too far from the schematic. This process is called *Check Against Source*. ■ At any time in the physical layout process, you can cross-probe to identify corresponding parts, packages, and signals in the layout design and the schematic. ■ The unification of libraries, which provides a single point of entry for all Virtuoso-driven multiple technology flows.	

Virtuoso RF Solution

This chapter provides a high-level overview of the new and enhanced features in the ICADVM20.1 and ICADVM18.1 releases.

Related Documentation

- *Virtuoso RF Solution Guide*

ICADVM20.1 Base Release

What's New and Enhanced

Features

Virtuoso RF Solution Launch	Enhanced
To access the Virtuoso RF Solution functionality, now you can set any of the following SHELL environment variables: `Virtuoso_MultiTech` or `Virtuoso_RF_Option`	
Renamed RF-Module Menu	Enhanced
The name of the *RF-Module* menu in the Virtuoso RF Solution environment has been updated as *Module*.	
Renamed Co-Design Sub-Menu	Enhanced
The name of the *Co-Design* sub-menu in the *Module* menu has been updated as *Edit-In-Concert*.	
Simulations for Cross-Fabric Designs Support Bond Wires	Enhanced
Earlier, electromagnetic simulations for cross-fabric designs supported only the flip-chip IC fabrics on top that are connected using bumps and balls. You can now also use an IC connected through bond wires. Any bond wire selected for inclusion in the model or connected to the nets or shapes selected for a model is saved in the `.clf` file and sent to Clarity for simulation.	
Metal Layer Embedded in Substrate	New
The electromagnetic simulations now consider metal layers that are embedded within substrate layers. The details for such layers are not given in the ICT or QRC technology files. Therefore, you need to use the new `metalLayers` tag within the `externalStackup` tag to provide details of such hidden layers.	
Support for Material File	New
You can now use the new `materialFile` tag in the process stackup to specify the material file that contains information about the various properties of the different materials used in your design. The EM Solver assistant sends the path of this material file to Clarity 3D Layout where its details are loaded into Material Manager. You can then use these materials to define layers in Layer Manager in Clarity 3D Layout.	

Support for Automatic Snapping	New
The *Stretch* command now supports automatic snapping of packaging wires. The automatic snapping helps to create a correct DRC connection in Virtuoso RF Solution.	

SKILL Functions

`vrfExportPackage`	New
Exports the specified package layout if its `sipObjectType` value is set to `package`.	

Environment Variables

`exportNoBumpTerm`	New
Includes the top-level IC layout terminals, which do not have the corresponding IO cells in the layout, while exporting the die.	

ICADVM18.1 ISR13

What's New and Enhanced

Features

License for Electromagnetic Solver Assistant	Enhanced
The Electromagnetic Solver assistant no longer requires the Virtuoso RF Option (95560) license. It can use only the Virtuoso Layout Suite EXL license. The assistant also checks for the supported solvers in the PATH environment variable. If the solvers are found, details are shown in the Electromagnetic Solver assistant. Otherwise, it displays an error.	
Default License for Clarity	New
You can now use the `CDS_CLARITY_LICENSE` shell environment variable to specify a default license suite to be used for Clarity. This avoids the Choose License Suites form being displayed every time Clarity is launched to show a model.	
Previewing EM Layout Model	Enhanced
You can now preview the EM layout without a recipe file, which means it is not required to run shape simplification before using the *Preview EM Layout* command on the *Pre-Process* tab. The command is always enabled when the AXIEM or Clarity simulators are selected for a model. The preview has also been enhanced to show all the geometries that are saved in the `.clf` file including the clustered vias and simplified shapes. In addition, it shows the impact of cutting boundaries and scaling factors on the geometries.	
Scaling the Z-Axis in 3D Viewer	New
While reviewing the mesh in 3D Viewer, you can now choose a scaling factor for the z-axis to clearly see the thin layers or to view the mesh between the layers.	
Port Validation	Enhanced
The *Validate Ports* command on the *Ports* tab now runs checks that are run by the Clarity or 3D-EM simulator to ensure that the ports are valid before the model is sent for simulation.	
Process Corner Setup for EM Models	Enhanced

The electromagnetic simulations run from Virtuoso now consider geometries connected below the substrate in addition to those connected above it. To enable these simulations, the tool now supports the following enhancements in the custom stackup:

- `metalLayers`: This tag can now contain a negative elevation value to specify the height of the metal below the substrate. The elevation is measured from the top of the substrate layer to the top of the metal layer below it.
- `dielectricLayersBackside`: This new tag specifies the thickness, permittivity, and loss tangent details for a dielectric layer placed below the substrate.
- `viaLayers`: This tag can now be used to specify the through silicon vias in addition to other vias.
- `outerCoating`: This new tag specifies the details of an outer coating that insulates a through silicon via from the substrate.

Use these tags for ICs that contain through silicon vias or for interposer ICs where a through silicon via connects to the bottom Ball Grid Arrays (BGA).

An external stackup already contains the `metalLayers`, `dielectricLayersBackside`, and `viaLayers` tags. If your IC contains a through silicon via, you can add the `outerCoating` tag in the `.emproc` file that uses an external stackup.

Display of Nets and Ports	Enhanced

The performance of the display of a large number of nets on the *Selection* tab or a large number of ports on the *Ports* tab of the Electromagnetic Solver assistant has been enhanced.

Environment Variables

`portLayerLocation`	New

Specifies the location of the same-layer ports that can be placed either on the top edge or the bottom edge of the metal layer.

ICADVM18.1 ISR12

What's New and Enhanced

Features

Viewing wirebonds in the Objects panel of the Palette assistant	New
You can view wirebonds in the *Objects* panel of the *Palette* assistant. You can also change the Visibility (*V*) settings of wirebonds through the *Palette*.	
Specifying Process Corner Setup for EM Models	Enhanced
The `.emproc` files that you use to specify process setup for EM models now support a new tag, `excludeLayers`. Use this tag to specify a list of layers to be excluded from the CLF file that is sent as an input to the solver. The `substrate` tag has also been enhanced to support multiple substrate layers. Earlier, you could specify a maximum of two substrate layers.	
Simplifying Shapes For Electromagnetic Simulations	Enhanced
Earlier, any shape not connected to a top-level pin was considered as floating and ignored during shape simplification. Starting from this release, connectivity extraction has been improved and a shape that is not connected to a top-level pin is not considered as floating if it is connected to an instance terminal with a net name. If a shape meets the specified shape simplification rules, it is processed.	
Annotations Assistant in 3D Viewer	New
You can use the new Annotations assistant in Virtuoso 3D Viewer to the view the annotations for mesh elements with high aspect ratio errors.	

Environment Variables

`setNotEmbedded`	New
Specifies a list of one or more cellviews whose instances are not to be considered as embedded components.	

ICADVM18.1 ISR11

What’s New and Enhanced

Features

New 3D Viewer to Review Mesh for the Models Simulated by AXIEM	New
You can now use the new Virtuoso 3D Viewer to review the mesh for the electromagnetic models simulated by AXIEM.	

Environment Variables

`runDirectory`	New
Specifies the run directory for all the models created in the Electromagnetic Solver assistant. The tool also saves the `.clf` file for models at this location.	

ICADVM18.1 ISR10

What's New and Enhanced

Features

Support for the Push-and-Shove Feature	New
A new push-and-shove feature has been added in Virtuoso RF. This feature applies to routing objects, such as wires and vias. Enabling push-and-shove enhances the correct-by-construction design and makes the design DRC- and shorts-aware. **Related**: Pushing and Shoving in Interactive Routing	
Specifying Process Corner Setup for EM Models	Enhanced
The `.emproc` files that you use to specify process setup for EM models now support the following new tags and properties: ■ `dielectricSimplification`: Use this new tag in both external or custom stackup to specify how to set the permittivity and loss tangent values for the simplified dielectric layers that are created by combining all dielectric layers between two metal layers into one. ■ `resistivity` property for `substrate` tags: Use this property in both external or custom stackup to specify the resistivity of the substrate in Ω*cm. This can be an alternative to the `conductivity` property of a `substrate` tag. ■ `sheetResistance` property for `metalLayer` tags in custom stackup: Use this property to specify the resistance of the metal layer in Ω/square. This value is used to calculate conductivity as `sheetResistance * thickness`.	
Electromagnetic Simulations for Clarity Models	Enhanced
The following settings are now supported for Clarity models: ■ The `PEC Ground Ring` current return path on the *Selection* pane. ■ The `Diagonal` port type in the *Type* drop-down list on the *Ports* pane. It creates diagonal ports across layers. ■ The shape simplification setup on the *Pre-Process* pane. Use this to simplify the layout shapes before running EM simulations for Clarity models.	

Reporting minSpacing and Short Violations	Enhanced
DRD now reports `minSpacing` and short violations between shapes on top-level nets and shapes on local nets.	

Environment Variables

`drdEditApkDrcBlockageOverlapCheck`	New
Specifies whether DRD checking should consider overlaps of blockages with metal shapes.	
`drdEditApkDrcShortCheck`	New
Specifies whether DRD checking should consider shorts of metals with other metal shapes.	
`enableAxiemDotPinPorts`	New
Specifies whether to create a short edge port at the location of a dot pin.	
`limitAxiemPortsToPins`	New
Specifies whether to use one of the edges of a rectangular pin to create a port.	
`enableAdvAxiemPortGen`	Removed
This variable has been replaced by `enableAxiemDotPinPorts` and `limitAxiemPortsToPins`.	

ICADVM18.1 ISR9

What's New and Enhanced

Features

Specifying Process Corner Setup for EM Models	Enhanced

The method to specify the process corner setup for EM model creation has been simplified. This involves the following changes:

- The EAD Process Settings form has been removed. You now need to provide `.emproc` files in a process corner directory, specified by the `processCornerDirectory` environment variable. The `.emproc` files provide process corner details, which include the layer stackup from an external ICT or QRC technology file or a custom stackup, information about substrate layers, layer height adjustments, and a list of layer-purpose pairs from which the shapes are to be exported to the `.clf` extracted model file.
- The Electromagnetic Solver assistant dynamically reads all `.emproc` files available in the process corner directory and displays the list of available corners in the *Corner* column of the rows in which models are defined.
- Each process corner is required to have a corresponding `.emproc` file. This allows you to use different process settings for each corner.
- The *Custom* column, which was visible in the model details for IC layouts, has been removed. You can now define custom stackup in the `.emproc` file that is not referring to a technology file.

Viewing Layer Stackup	Enhanced

In the previous releases, you could view the layer stackup by using the *Draw* command on the Process Settings form. Starting from this releases, the commands to view layer stackup are available on the toolbar of the Electromagnetic Solver assistant.

To view the stackup as defined in the ICT or QRC technology file, or in a custom stackup, select a model and choose the *View Stackup* command from the toolbar of the Electromagnetic Solver assistant.

To view the effective stackup for a model created using AXIEM, use the *View AXIEM Stackup* command. For other solvers, export the model to that solver and then view the stackup up in the solver environment.

Creating an Extracted View From Smart View	New
You can now use Quantus QRC-based smart views to create extracted views after EM simulation.	
Extracting Models as Complete S-Parameter Cellviews	New
A new flow has been added that allows you to extract models for instances as a full cellview instead of an extracted view to be placed in a schematic. This feature is helpful when you cannot access the connectivity details of instances. In such cases, the tool uses the top level pins of the instance and extracts a model for it as a complete cellview, which is an S-parameter results file. Optionally, you can also create a symbol view, which you can use to instantiate the model in a schematic. The new cellview and symbol are saved as a part of the library.	
Probing from Board to Package and Die	Enhanced
In previous releases, you could probe nets from a package to a die in Co-Design mode. The support is now extended to the board fabric. You can probe and highlight a net from the board to the package and die.	

SKILL Functions

`vrfExportLayoutSkill`	New
Exports the die for both IC and package layouts. The function supports both `IO cell` and `Shape with overlapping label` as *Die Interface Type*. It generates a package type library that contains abstract, Technology Independent Layout Pcells (TILPs), schematic, and symbol views.	

Environment Variables

`discretizationMaxError`	New
Specifies the maximum acceptable error limit to be considered during discretization of the curved shapes in a package layout to polygons, which is done before sending the layout details to the EM solver. This error limit value is used by the Electromagnetic Solver assistant to identify the number of points for each polygon. For a high acceptable error limit, which also means a high error tolerance, each polygon shape has fewer points.	
`processCornerDirectory`	New
Specifies the path to the directory where `.emproc` files for process corner settings are saved.	

processDefaultCorner	New
Specifies the name of the default corner to be used for a new model created in the Electromagnetic Solver assistant.	
The environment variables listed below have been removed. You can now configure these settings in the .emproc files that define the process settings to be used for model creation. ■ ExtraSubThickness ■ ExtraSubPermittivity ■ ExtraSubConductivity ■ includeLPPs ■ heightAdjustmentLayerName ■ heightAdjustmentValue	Removed

ICADVM18.1 ISR8

What's New and Enhanced

Features

Controlling the Spacing between Bond Fingers on Guides	New
The Create Bond Wire form now includes the *Uniformly Distribute Objects on Guide* check box. When this check box is selected, bond fingers are distributed evenly on the guide. When not selected, the minimum spacing between adjacent bond wires is set to the `pkgMinBondwireSpacing` value defined in the technology file.	
Support for Area-based Rules	New
For Virtuoso RF Solution, interactive routing now supports three area-based rules: `minWidth`, `maxWidth`, and `minSpacing`. These rules are automatically honored when the geometric wire spans across regions.	
Creating Bond Finger Definitions	Enhanced
Instead of creating (reusable) bond finger profiles, you can now create bond finger definitions. The following changes have been made to the UI: ■ The *Profile* option in the Create Bond Wire form has been renamed to *Definition*. ■ The Add Bond Finger Profile form has been renamed to Add Bond Finger Definition. ■ The *Profile Name* option in the Add Bond Finger Profile form has been renamed to *Definition Name*.	

Environment Variable

enableAdvAxiemPortGen	Enhanced
You can now use the `enableAdvAxiemPortGen` variable to create valid edge ports for dot pins. Earlier, this variable allowed creation of edge ports only for rectangular pins.	

ICADVM18.1 ISR7

What's New and Enhanced

Features

Support for Clarity to Run Electromagnetic Simulations	New
The Electromagnetic Solver assistant now supports the new Clarity™ 3D Solver. With the capability to run on distributed computing, Clarity provides high performance and efficiency in resolving the most complex electromagnetic challenges. To run electromagnetic simulations, you can choose `Clarity` from the drop-down list in the Simulator column of the Electromagnetic Solver assistant.	
Support for Wirebonds	New
The Virtuoso RF Solution has been enhanced to support wirebond attached dies in addition to flip chip dies. The solution provides options to create and edit the following key components of bond wires: ■ **Guides:** Paths drawn around dies to define snapping locations for bond wire endpoints and bond fingers. ■ **Bond wires:** Wires that connect the die pads either to bond fingers on the component substrate or to another die pad on another die. ■ **Bond fingers:** A metal pad on the outer layer of component substrate to which a wire bond is attached. The Electromagnetic solver assistant now supports packages and RF modules with bondwires for full 3D analysis using the Clarity and Sigrity 3D-EM solvers.	

Environment Variables

Electromagnetic Solver Assistant	New
■ `vem.ic defaultSimulator`: Specifies the default simulator for an IC design layout. ■ `vem.package defaultSimulator`: Specifies the default simulator for a package design layout.	

ICADVM18.1 ISR6

What’s New and Enhanced

Features

Extracting Models for Cross-Fabric Designs	New
You can now open cross-fabric designs in the Co-Design mode in Virtuoso Layout EXL and use the Electromagnetic Solver assistant to extract models. The resulting s-parameter files contain ports or geometries for multiple fabrics or cellviews.	
Creating Extracted Views for Cross Fabrics	New
Going forward, you can create extracted views for hierarchical and cross-fabric models.	
Supporting Package TILP for LVA and Co-Design	New
These are the enhancements through this feature: ■ The package TILP can be attached to the substrate similar to a die abstract. The die TILP parameters apply to the package TILP attachment as well. ■ The Co-Design mode can be launched from the board and package layout. ■ The LVA checker can be run from the board, package, or die layout.	
Viewing Process Stack	New
For the models created in the Electromagnetic Solver assistant, you can now use the new *Draw* command on the Process Settings form to view the layer stack drawn above the substrate layer. The layer stack is displayed in the Virtuoso Layout Viewer window and shows the layer name, height, thickness, and permittivity for each dielectric or metal layer in your model. You can view the layer stacks in two modes: ■ *As Defined*: In this mode, the layer stack details are directly taken from the ICT file. It is available for both 3D-EM and AXIEM models. ■ *Effective*: In this mode, the dielectric layers are averaged for the optimization of a simulation run. This mode is available for only the AXIEM models.	
Port Generation for AXIEM	Enhanced

- Earlier, for pins with rectangular edges, ports were created using an exterior edge. Now you can set the enableAdvAxiemPortGen environment to use one of the rectangle edges of a rectangle pin for port creation.
- Earlier, automatic port generation used to fail for pins with multiple pinfigs. It has now been enhanced to create a port using the largest pinfig out of the available pinfigs.

ICADVM18.1 ISR5

What’s New and Enhanced

Features

Simplifying Shapes For Electromagnetic Simulations	New
You can now use the setup on the *Pre-Process* tab to simplify shapes, which reduces fine details in those areas of the layout that may not impact the result of electromagnetic simulations run using AXIEM. For example, you can remove small dangling shapes or merge closely placed shapes. These changes reduce the number of shapes to be considered during the simulation run, thereby, making the process faster. In addition, Virtuoso Layout EXL also automatically converts circle and ellipse shapes in an IC layout to polygon shapes with the number of sides defined by the simplifyToSides environment variable. The default value of this variable is `6`. Fewer sides reduce the number of unknowns in the electromagnetic mesh to be used by AXIEM.	
Automatic Port Generation for SMD Pads	Enhanced
For package layouts, automatic port generation for SMD pads has been enhanced to create ports on the same layer or across layers depending on the availability of a ground reference. While generating a port, Layout EXL first searches above and below the SMD pad for a ground reference. If found, a cross-layer port is created. If not, Layout EXL next searches for a ground reference on the same layer. If a reference is found, it creates a same-layer port.	

ICADVM18.1 ISR4

What’s New and Enhanced

Features

DRD Region Overrides	Enhanced
DRD now supports region overrides for verifying a layout.	
Specifying the Via Clustering Option for Models	Enhanced
You can now use the *Cluster Vias* check box on the *Pre-Process* tab of the Electromagnetic Solver assistant to specify the via clustering option of each model. This setting is saved along with other model information.	

ICADVM18.1 ISR3

What's New and Enhanced

Features

Creating TILP	New
You can now create padstack vias, SMD pads, embedded components, or die TILPs from a package layout by using the GUI options in Virtuoso RF.	
Policies for Managing Schematic/Layout Views in the Virtuoso RF/Virtuoso MultiTech Environment	New
Now, there are various policies applied for handling any type of the schematic or layout views in the Virtuoso RF/Virtuoso MultiTech Environment.	
DRD Net Overrides	Enhanced
DRD now supports net overrides for verifying a layout.	
Dynamic Shape Priority	New
Dynamic shapes now support a priority attribute that determines the order in which the overlapping dynamic shapes are voided.	
Running Electromagnetic Solver Extraction	Enhanced
■ For the AXIEM simulator, the *Open Simulator* menu now shows the *Generate Mesh*, *View 3D Mesh*, *Mesh and Simulate*, and *Stop* commands for improved handling of mesh and simulation. For more details, refer to Running an AXIEM Simulation. ■ The *Ports* tab on the Electromagnetic Solver assistant has been enhanced for the AXIEM simulator. The commands to generate ports from layout pins, add or delete ports, highlight ports on layout, and to change the order of the ports listed in the *Ports* table are now placed together. You can now specify the port type as `No reference`, `Differential`, `Connect to Lower`, or `Connect to Upper`.	

- For AXIEM simulations, you can now override the mesh settings for individual shapes by applying the `AXIEMMeshDensity` and `AXIEMZeroThickness` properties to those shapes in the layout.
- You can use the following two environment variables to adjust the height of any layer to ensure that the desired capacitance value for the cell is maintained. The shift helps in emulating the effect of the high-K dielectric used between the MIMCAP plates.
  - `heightAdjustmentLayerName`
  - `heightAdjustmentValue`
- You can use the `includeLPPs` environment variable to include additional layer-purpose pairs in the model.
- You can use the following three environment variables to specify an additional dielectric above the bulk silicon:
  - `ExtraSubThickness`
  - `ExtraSubPermittivity`
  - `ExtraSubConductivity`

ICADVM18.1 ISR2

What’s New and Enhanced

Features

Fixing the Difference in the Number of IO Pads	New
The LVA fixer addresses the mismatch in the number of IO pads that the LVA checker reports for the die abstract, and propagates the changes to the corresponding symbol and schematic views of the die abstract.	
Fixing Connectivity Mismatches	New
The LVA fixer addresses the connectivity mismatches that the LVA checker reports for the die abstract, and propagates the changes to the corresponding symbol and schematic views of the die abstract.	
Probing in the Co Design Mode	New
You can now probe nets, bump instances, and Ball Grid Array (BGA) balls in the Co Design Mode. When probing an object, you can specify a highlight color. The selected object is highlighted hierarchically in the selected color across the top-level and die layout. For example, when probing a BGA ball, the net connecting the BGA ball is highlighted hierarchically starting from the selected BGA ball.	

ICADVM18.1 ISR1

What's New and Enhanced

Features

Applying Thermal Shrink Factor	New
The Virtuoso RF Solution lets you encapsulate dies of different semiconductor materials into a single package.To overcome any connectivity issues due to differences in Coefficient of Thermal Expansion (CTEs), Virtuoso lets you set the thermal shrink factor for such dies. Applying the thermal shrink adjusts the initial positions of IO pads such that the connectivity is not broken when the dies are heated.	
Implementing Die Export 2.0	New
Die Export 2.0 and the associated form streamlines the process of creating a die abstract. The Die Export functionality in the Virtuoso RF Solution now supports two types of die IO Pads, cells and shapes with overlapping labels. In addition, the new Export Die form is applicable for both the Virtuoso RF Solution and Virtuoso MultiTech Framework.	
Supporting Package Constraints in Interactive Routing	New
For the Virtuoso RF Solution, interactive routing now supports three packaging constraints: pkgMaxLineWidth, pkgMinLineWidth, and pkgMinSpacing. These constraints are automatically honored while selecting the width and spacing of a wire on a specific layer.	
Running 3D Electromagnetic Extractions	Enhanced
■ You can now use the new `Electromagnetic` workspace that is provided in Virtuoso Layout EXL to open the Electromagnetic Solver assistant and a few UI components that help you configure and run simulations, and create 3D models. For more details, refer to Launching the Electromagnetic Solver assistant. ■ You can use the new AXIEM simulator to run electromagnetic simulations for IC designs. AXIEM uses the Method of Moments (MOM) technique, whereas Sigrity 3D-EM uses the Finite Element Method (FEM) technique. To use AXIEM, you require the `AXIEM_3D_Planar_EM` license. ■ You can now distribute simulation runs to compute farms, such as LSF or Open Lava, by using a Distributed Resource Management System (DRMS). For this, you can use the `drmsCommand` environment variable to specify the `bsub` command to be submitted for job distribution.	

SKILL Functions

vrfLowerPriority	New
Lowers the priority of a specified dynamic shape so that it is one unit less than the lowest priority currently set for dynamic shapes on the layer.	
vrfRaisePriority	New
Raises the priority of a specified dynamic shape so that it is one unit higher than the maximum priority currently set for dynamic shapes on the layer.	

Environment Variables

Electromagnetic Solver Assistant	New

You can use the following environment variables to customize the default values for the Simulation Settings form:

- Variables for both IC and package layout: ExplicitDC, MaxNumCPU, dxPlusSize, dyPlusSize, dzPlusSize, dxMinusSize, dyMinusSize, dzMinusCondition, UseConformal
- Variables in the vem.ic product partition to specify values for IC designs: dzMinusSize, MetalType, DielectricCondition
- Variables in the vem.package product partition to specify values for package designs: dzMinusSize, MetalType, DielectricCondition

ICADVM18.1 Base Release

What’s New and Enhanced

Feature

Virtuoso RF Solution for High Frequency Product Designs	New
ICADVM18.1 introduces the Virtuoso RF Solution, which offers the path-breaking analysis and innovative simulation-driven layout capabilities for more robust and efficient design implementation. It dissolves the traditional tool boundaries between IC and module/package designs. The solution has enabled designers create packages in Virtuoso by adding package design capabilities to other capabilities of Virtuoso. This solution is, primarily, for the RF module designs that can be edited, simulated, extracted, or back-annotated in Virtuoso. It allows the package and die designs to evolve simultaneously and in concert with one another (Co Design). Layout versus Schematic (LVS) integrity can be maintained during logic and placement changes in the package and die. In addition, the Virtuoso RF Solution gives package layout designers productivity-enhancing features to run a variety of EM solvers within Virtuoso. Designers can fine-tune portions of the system by extracting and simulating circuit paths and context around the circuit paths.	