cadence®

Virtuoso Software Licensing and Configuration User Guide

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

Configuring the Virtuoso Design Environment 7
Virtuoso Design Environment 8
Overview of Virtuoso Product Families 8
Types of Licenses 8
License Consumption in Product Tiers 9
Launching the Virtuoso Product Family Applications 13
Choosing Default Applications 13
Setting the CDS_Netlisting_Mode Environment Variable 14
Licensing in Design Framework II 14
License Check Out/Check In Behavior 15
License Activity for Various Applications 17
Advanced Features and Token Licenses 31
Tracking Token Licenses 35
lmstat Command 35
Software Product License Management Form 37
Software Product License Management Form: Status Tab 39
Software Product License Management Form: Token Usage Tab 46
Software Product License Management Form: Ordering Tab 48
Software Product License Management Form: Diagnostics Tab 53
Software Product License Management Form: Performance Tests Tab 56
Product Tier Features 59
ADE Product Family Features 60
Hardware and Software Requirements 67
Virtuoso Design Environment Hierarchy 68
Virtuoso Design Environment Executables 70
32- and 64-Bit Platforms 70
dbAccess Command-Line Executable 70
Virtuoso Design Environment Licensing Setup 73
Preface 77
Scope 78
Licensing Requirements 78

Related Documentation . . . 79
What's New and KPNS . . . 79
Installation, Environment, and Infrastructure . . . 79
Technology Information . . . 79
Virtuoso Tools . . . 79
Additional Learning Resources . . . 81
Video Library . . . 81
Virtuoso Videos Book . . . 81
Rapid Adoption Kits . . . 81
Help and Support Facilities . . . 81
Customer Support . . . 82
Feedback about Documentation . . . 82
Typographic and Syntax Conventions . . . 83

Setting Up the Virtuoso Software . . . 85
Quick Start: Linux and Unix Environments . . . 86
Configuration and Startup Procedures . . . 88
Setting the Installation Path . . . 88
Setting Up a User Account . . . 90
Setting the Focus Correctly . . . 90
Specifying Cadence Environment Variables . . . 92
Modifying the .rhosts File . . . 97
Modifying the .Xdefaults or Equivalent File . . . 98
Modifying the .cdsinit File . . . 101
Verifying Your System Configuration . . . 102
Distributing the User Files . . . 103
Configuring Remote Displays . . . 103
Library and Tool Issues . . . 105
Virtuoso Design Environment Executables . . . 106
32- and 64-Bit Platforms . . . 106
dbAccess Command-Line Executable . . . 106

Additional Virtuoso Software Licensing and Configuration Information 109
X Window System 110
Running 64-Bit Versions of Applications 111
TrueColor Visuals 114
Finding Available Visuals 114
Pseudocolor and TrueColor Visuals 115
Locale Settings 116

A
Licensing Environment Variables 119
CheckoutOrder Variables 120
UseNextLicense Variables 126
Other Licensing Variables 131

Configuring the Virtuoso Design Environment

The following topics are discussed in this chapter:

- Virtuoso Design Environment on page 8
- Launching the Virtuoso Product Family Applications on page 13
- Advanced Features and Token Licenses on page 31
- Tracking Token Licenses on page 35
- Software Product License Management Form on page 37
- Product Tier Features on page 59
- Virtuoso Design Environment Hierarchy on page 68
- Virtuoso Design Environment Licensing Setup on page 73

Use the *Cadence Installation Guide* and *Cadence License Manager* to install the Virtuoso® Design Environment software and configure the licenses.

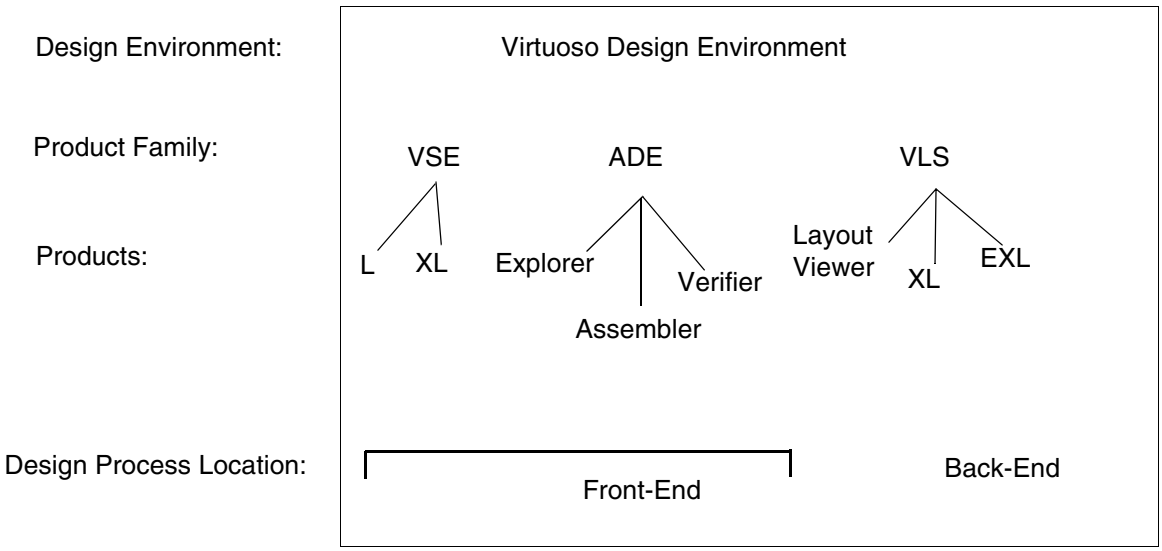
Virtuoso Design Environment

The Virtuoso platform provides a seamless transition from one Virtuoso design environment to another as a result of the updated common user interface (UI) and automated license checkout process.

This platform provides a unified front-to-back design environment. The front-end design environment consists of Virtuoso Schematic Editor® (VSE) and ADE product families. The back-end consists of the Virtuoso Layout Suite® (VLS) product family.

Each of the product families contains applications in a tiered packaging structure, and conform to common and consistent licensing behavior and operation.

Overview of Virtuoso Product Families



Types of Licenses

- User, host, and display (UHD) licenses: The L and XL product tiers have UHD licenses. This is the same type of license used prior to the IC 6.1 release.
- Job (J)-based and token-based licenses: The GXL tier uses job-based (J) licenses which are not shared. Tokens are used to start applications in the GXL tier.

  For information on the number of tokens required for the GXL product tier, see GXL Multi-Feature License Tokens on page 31.

License Consumption in Product Tiers

Product Tier: L

The L product tier offers basic design creation and implementation capabilities.

Base Product Name	License Feature Name	License Type
Virtuoso Schematic Editor L	`Virtuoso_Schematic_Editor_L` (95100)	UHD

Important

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 continued for flows that required additional performance. With improvements in Layout XL performance and addition of new flexible configuration of Layout XL in ICADVM20.1, which provides additional performance to support additional flows, the Layout L application has been removed from the *Launch* menu, and the associated *File Open* dialog boxes. If you are using Layout L flows in ICADV12.3, you should migrate to Layout XL flows in ICADVM20.1. For more information, see *Virtuoso Layout Suite XL: Basic Editing User Guide* and *Virtuoso Layout Viewer User Guide*

Product Tier: XL

The XL product tier contains IC design products that introduce new technologies and advancements in design automation. The XL tier offers tightened linkage between design and implementation phases in the areas of creation, analysis, implementation, and repair.

Base Product Name	License Feature Name	License Type
Virtuoso Layout Suite XL	`Virtuoso_Layout_Suite_XL` (95310)	UHD
Virtuoso Schematic Editor XL	`Virtuoso_Schematic_Editor_XL` (95115)	UHD

Product Tier: GXL

The GXL product tier incorporates advanced technology specifically needed to solve the difficult design challenges at 90 nm and below. The GXL tier provides advanced finishing tools suited for tackling the most difficult design problems and offers the greatest opportunity for productivity, yield, and time-to-market improvements.

Base Product Name	License Feature Name	License Type
Virtuoso Layout Suite GXL	`Virtuoso_Layout_Suite_GXL` (95323)	Multi-feature, Job Locked (J) 4 tokens.
	Note: License part number 95321 for Virtuoso Layout Suite GXL has been replaced by 95323.	**Note:** A token is one copy of a multi-feature license.

Important

In IC6.1.0 and ICADV12.1, and later releases for Advanced Nodes, the Layout GXL application provided an identical Layout environment to Layout XL, with the exception that it used `Virtuoso_Layout_Suite_GXL` job-based and token-based licenses versus 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.

However, `Virtuoso_Layout_Suite_GXL` 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 (Tokens per Feature (VLS GXL) on page 32. If you are using Layout GXL application in flows in IC6.1.7 or ICADV12.3, you should migrate to Layout XL flows in ICADVM20.1. For more information, see the license search order Software Product License Management Form: Ordering Tab on page 48

Product Tier: EXL

Note: The new EXL product tier provides access to all functionality and features of Virtuoso Layout Suite EAD (ICADV12.3) for advanced electrically aware designs. 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. If you are using Layout EAD application in flows in IC6.1.7 or ICADV12.3, you should migrate to the Layout EXL flow in ICADVM20.1.

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.

Base Product Name	License Feature Name	License Type
Virtuoso Layout Suite EXL	`Virtuoso_Layout_Suite_EXL` (95800)	UHD

Only one base license per product family tier is checked out at a time. As a result:

- The VLS EXL tier does not require an VLS XL tier license. A higher checked out license allows you to run lower-tier applications. For example, you can run VLS XL and VLS EXL in the same session with only a VLS EXL license checked out. However, the higher checked out license will remain checked out until all VLS windows are closed.

  The following license is required in addition to the base license:

  - Token-based features in VLS GXL can be launched from the VLS XL or VLS EXL tier. (Tokens per Feature (VLS GXL))

For information on the specific features in each of the product tiers, see Software Product License Management Form: Diagnostics Tab on page 53.

Other products that are not tier-based require separate licenses, as listed below.

Product Name	License Feature Name	License Type
Virtuoso Layout Viewer	`Virtuoso_Adv_Node_Framework` (95011)	UHD
Virtuoso ADE Explorer	`Virtuoso_ADE_Explorer` (95250)	UHD
Virtuoso ADE Assembler	`Virtuoso_ADE_Assembler` (95260)	UHD
Virtuoso ADE Verifier	`Virtuoso_ADE_Verifier` (95270)	UHD

Additional Feature Licenses

Some product features require you to check out additional licenses on top of the base product license for the product you are using.

For example, Layout Dependent Effects (LDE) resimulation flows require the `Virtuoso_Variation_Analysis_Op` (Product Number `95510`) license on top of the base product license.

Similarly, advanced node features require the `Virtuoso_Adv_Node_Opt_Layout` (Product Number `95511`) or the `Virtuoso_Adv_Node_Opt_Lay_Std` (Product Number `95512`) license on top of the base product license.

Note: Advanced node license is not required if you are opening mature node data.

Launching the Virtuoso Product Family Applications

You can start applications in each Virtuoso product family using one of the following methods:

- *File – Open* and *File – New* from the CIW
- Library Manager
- *Launch* menu from the design windows
- *File – Open* and *File – New* from the design windows
- Hierarchy Editor
- Navigator Assistant
- History
- Bookmarks

Choosing Default Applications

Opening a design in a specific product family tier and keeping that setting as the default upon startup is performed by using any of the following methods to open a design:

- *File – Open* and *File – New* from the CIW
- Library Manager
- *File – Open* and *File – New* from the design windows
- *File – Set Default Application* from the design windows

To set the default application using any of the methods above,

1. Choose the desired application in the *Application* field.
2. Select the *Always use this application for this type of file* option.

   This field is mapped to the environment variables

   - `maskLayoutDefaultApp`
   - `schematicDefaultApp`
   - `schematicSymbolDefaultApp`

In the event that the default application is not a base application (L tier), only the license of the chosen application will be checked out, not the base application. Only one of the base licenses in a given family is ever checked out at any one time.

Setting the CDS_Netlisting_Mode Environment Variable

When working in ADE, you need to set the `CDS_Netlisting_Mode` environment variable to `Analog` using one of the following methods:

- Type the following at the command line prior to launching `virtuoso`

  ```
  setenv CDS_Netlisting_Mode=Analog
  ```

- Type the following SKILL in the command interpreter window (CIW)

  ```
  setShellEnvVar("CDS_Netlisting_Mode=Analog")
  cdsSetNetlistMode()
  t
  cdsGetNetlistMode()
  "Analog"
  ```

Licensing in Design Framework II

In the ICADVM20.1 release, product license 111 and 95011 (`Virtuoso_Adv_Node_Framework`) is checked out at workbench start and while performing the following activities:

- Starting Virtuoso applications in either graph or non-graph mode
- Launching routines that interact with design framework II® (DFII) data (for example, CDF, Pcells, and param)
- Accessing Cadence data using `itkDb`
- Launching open simulation system (OSS) netlisters

In the event of a DFII license failure at startup, the Virtuoso application exits and an error message is displayed in the CIW as well as a log file.

License Check Out/Check In Behavior

License Usage When Using Interactive Commands

- **Product Tier**: **L**

  In the L product tier, if a schematic design is opened in read-only mode or changed to read-only mode by choosing *File – Make Read Only*, no license checkout is required.

  However, when a schematic design is opened for editing or when it is made editable by choosing *File – Make Editable* in the application window, the `Virtuoso_Schematic_Editor_L` license is checked out.

- **Product Tier**: **XL**

  In the XL product tier, the `Virtuoso_Layout_Suite_XL` license is checked out when a layout is opened. An additional `Virtuoso_Adv_Node_Opt_Layout` or `Virtuoso_Adv_Node_Opt_Lay_Std` is checked out when the layout design contains Advanced Node features (License Requirements for Advanced Node Features). These licenses are checked in when all layout windows are closed, including all Layout Viewer, Layout XL, and Layout EXL windows.

- **Product Tier**: **GXL**

  When you invoke commands requiring GXL capabilities from Virtuoso Layout Suite XL or EXL, `Virtuoso_Layout_Suite_GXL` tokens are checked out. These tokens are checked in after the completion of commands. If you invoke advanced node features, an additional `Virtuoso_Adv_Node_Opt_Layout` or `Virtuoso_Adv_Node_Opt_Lay_Std` is checked out. These licenses are checked in when the last layout window is closed.

- **Product Tier**: **EXL**

  In the EXL product tier, opening a design in Layout EXL checks out a `Virtuoso_Layout_Suite_EXL` license. An additional `Virtuoso_Adv_Node_Opt_Layout` or `Virtuoso_Adv_Node_Opt_Lay_Std` is checked out when the layout design contains Advanced Node features (License Requirements for Advanced Node Features).

License Usage When Using Interactive Commands

When you launch a GXL-based feature from the XL or EXL tier, the following token license activities occur:

- A token license is checked out during the execution of a command from a selected feature menu or toolbar.
- A token license is checked in automatically after completion of a command for some features (the Analog Auto Placer is one example of automatic check in).
- A token license is not checked out when a feature is selected in the menu or toolbar.

Note: The *Context* and *Window – Toolbar* menus do not affect the state of a token license. These menus are navigation aids and are not linked to license activity.

For all of the Virtuoso GXL features, with the exception of the Custom Digital Placer and Floorplanning, licenses are checked in after completion of the command. For the Custom Digital Placer and Floorplanning, the license is checked in only when the task is disabled or the XL or EXL session is exited.

There is now an enforced restriction on license check-in that prevents license check in when an application or GXL capability is still using the license.

License Usage When Using Non-Interactive Commands

Non-interactive, lengthy, and batch commands check out the license at the beginning of the command and automatically check in the license upon completion of the command activity. This model applies to automatic placers and routers.

License Activity for Various Applications

License Requirements for Advanced Node and Advanced Methodologies Features

Depending on the advanced node feature being used, you will need one of the following licenses, in addition to the base product license:

- `Virtuoso_Adv_Node_Opt_Lay_Std` (Product Number 95512)
- `Virtuoso_Adv_Node_Opt_Layout` (Product Number 95511)

The `Virtuoso_Adv_Node_Opt_Layout` (95511) license supports all the advanced node features in this release. A subset of advanced node features is available using the `Virtuoso_Adv_Node_Opt_Lay_Std` (95512) license.

The following table lists the license requirements for different features:

Table 1-1 License Requirements for Advanced Node Features

Product Capability/Feature	**License Required**
Technology database contains a layer with more than two masks	`Virtuoso_Adv_Node_Opt_Layout`
Cellview or technology database contains WSP data. For more information, refer to the *Virtuoso Width Spacing Patterns User Guide*	`Virtuoso_Adv_Node_Opt_Layout` If `Virtuoso_Layout_Suite_EXL` (95800) is already checked out, either `Virtuoso_Adv_Node_Opt_Layout` or `Virtuoso_Adv_Node_Opt_Lay_Std` is required.
Cellview or technology database contains a constraint or constraint parameter that requires 95511 license. For more information, refer to Supported Constraints and Parameters: ICADVM20.1 Only-95511 in the *Virtuoso Technology Data Constraints Reference.*	`Virtuoso_Adv_Node_Opt_Layout`
Technology database contains snap pattern definitions.	`Virtuoso_Adv_Node_Opt_Lay_Std`
Technology database contains a layer with 2 masks and no layer with more than 2 masks.	`Virtuoso_Adv_Node_Opt_Lay_Std` or `Virtuoso_Adv_Node_Opt_Layout`
Cellview or technology database contains a constraint or constraint parameter that requires 95512 license. For more information, refer to Supported Constraints and Parameters: Advanced Nodes Only in the *Virtuoso Technology Data Constraints Reference.*	`Virtuoso_Adv_Node_Opt_Lay_Std` or `Virtuoso_Adv_Node_Opt_Layout`

Product Capability/Feature	License Required
Advanced node features that are not listed in this table.	`Virtuoso_Adv_Node_Opt_Lay_Std` or `Virtuoso_Adv_Node_Opt_Layout`
When no advanced node data is present.	Neither `Virtuoso_Adv_Node_Opt_Lay_Std` nor `Virtuoso_Adv_Node_Opt_Layout`

The checkout order of these licenses is controlled by the `VLSAdvOptLicenseCheckoutOrder` cadence environment variable.

```
"license" "VLSAdvOptLicenseCheckoutOrder" 'string "95511 95512"
```

Table 1-2 Advanced Node License Checkout Order

License Checkout Order	Description
`"95511 95512"` Default	It means, the application will first try to check out license 95511 for all features. If license 95511 is not available and a feature requires license 95512, then license 95512 is checked out instead.
`"95512 95511"`	If you specify this checkout order, the application will first try to check out license 95512 for features requiring license 95512. If license 95512 is not available, the application will check out license 95511 instead.
`"95512"`	If you specify this checkout order, the application will only check out license 95512 and will disable all features that require license 95511.
`"95511"`	If you specify this checkout order, the application will check out license 95511 for all features.

You can also reset the checkout order using the controls in the Software Product License Management Form: Ordering Tab.

Virtuoso Visualization and Analysis XL and ADE Licenses

Virtuoso Visualization and Analysis tool runs only in the XL mode, which supports all the features of Virtuoso Visualization and Analysis L and XL from previous releases. It follows the license check out procedure given below.

- If Virtuoso Visualization and Analysis XL is opened from within ADE Explorer or ADE Assembler, it shares the same license tokens as that of the product.
- If Virtuoso Visualization and Analysis XL is opened in stand-alone mode or from Virtuoso, it can either check out the Virtuoso Visualization and Analysis XL license or a tier of Maestro license, depending upon the preferences you set in the `VIVALicenseCheckoutOrder` `cdsenv` variable. By default, this variable is set to `VIVA, ADE`, which results in the following license check out tasks being performed:
  - Checks out the Virtuoso Visualization and Analysis XL license.
  - If the checkout operation in the previous step fails, checks out a Maestro (Virtuoso ADE Explorer or Virtuoso ADE Assembler) license according to the checkout order specified in the `checkoutOrder` .cdsenv variable.

  If this variable is set to `ADE, VIVA`, the license check out tasks are performed in the following order:
  - Checks out the license of ADE Explorer or ADE Assembler according to the checkout order specified in the `checkoutOrder` .cdsenv variable.
  - If the checkout operation in the previous step fails, checks out the Virtuoso Visualization and Analysis XL license.
- When Virtuoso Visualization and Analysis XL is opened in stand-alone mode or from Virtuoso, the tool releases license when all the Virtuoso Visualization and Analysis XL windows are closed.
- When Virtuoso Visualization and Analysis XL is opened from within ADE and you close the ADE window, the Virtuoso Visualization and Analysis XL process holds the ADE license and continues to operate until all the main windows are closed.

License Usage in ADE Explorer and ADE Assembler

Depending on the run mode and feature being used ADE Explorer and ADE Assembler require different licenses. The following table describes the license consumption:

Table 1-3 License Checkout Behavior for ADE Explorer and ADE Assembler

Action	License Checkout Preference Order (depending on the availability)
Open ADE Explorer	`Virtuoso_ADE_Explorer` `Virtuoso_ADE_Assembler`
Open ADE Assembler	`Virtuoso_ADE_Assembler`
Switch from ADE Explorer to ADE Assembler	Check in `Virtuoso_ADE_Explorer` license and check out `Virtuoso_ADE_Assembler` license
Switch from ADE Assembler to ADE Explorer	Use only `Virtuoso_ADE_Assembler` license

In addition to the Virtuoso ADE Assembler license, you would require the `Virtuoso_Variation_Option` (95265) license for the following advanced run modes in Virtuoso ADE Assembler:

- High yield estimation
- Improve yield run mode
- Advanced Monte Carlo options, such as sample reordering, creation of statistical corners, and mismatch contribution

License Checkout When Invoking LDE Resimulation Flows

You can run simulations with Layout-Dependent Effects (LDEs) in ADE Assembler and ADE Explorer. When you are using a third-party LDE engine, in addition to the base product license, you need the following licenses:

- License Number 95230 (`Virtuoso_LDE_Analyzer`) or 95311(`Virtuoso DFM Option`)
- License Number 95510 (`Virtuoso_Variation_Analysis_Op`) or 95600 (`Virtuoso_Layout_Suite_EAD`)

  Note: If license 95600 (`Virtuoso_Layout_Suite_EAD`) is already checked out, license 95510 (`Virtuoso_Variation_Analysis_Op`) is not required.

When you are using the PVS deck, you need the following licenses in addition to the base product license:

- License Number 95230 (`Virtuoso_LDE_Analyzer`) or a combination of 95311 (Virtuoso DFM Option) and 96220 (Cadence Physical Verification System Layout vs. Schematic)
- License Number 95510 (`Virtuoso_Variation_Analysis_Op`) Virtuoso Implementation Aware Design Option or 95600 (`Virtuoso_Layout_Suite_EAD`)

Using the LDE resimulation flow, you can include the LDE parameters — extracted from the Modgen constraints or from a layout view — into the simulation netlist and analyze their effect on the circuit performance. For more information, see *Advanced Node (20nm): Simulating Designs with Layout-Dependent Effects (LDEs)* in the *Virtuoso ADE Assembler User Guide*.

License Checkout When Running the Virtuoso Schematic and Verilog Driven Mixed-Signal Flow

To run the Virtuoso Schematic and Verilog Driven Mixed-Signal Flow, you must set the `CDS_ENABLE_VMS` environment variable before starting Virtuoso. When you do this a `Virtuoso_MixedSignalOpt_Layout` license (Product Number 95710) is checked out in addition to the base product license.

License Checkout While Running the Electrically Aware Design (EAD) Flow

While running simulations in ADE Assembler to generate electrical data and save datasets for the EAD flow, in addition to the base license for ADE Assembler, the EAD flow requires either of the following licenses:

- `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_Variation_Analysis_Op` (Product Number: 95510)

  Note: If 95600 or 95800 is already checked out, 95510 is not used.

License Checkout for Virtuoso Photonics Platform

To enable the photonics flows, you can set either the `Virtuoso_Photonics_Platform` shell or the `Virtuoso_Photonics_Option` environment variable.

After setting the `Virtuoso_Photonics_Platform` variable, when you start Virtuoso, an attempt is made to check out and initialize the `Virtuoso_Photonics_Platform` license

(Product Number 95551). If this license is available, you will not need separate licenses for Virtuoso CurvyCore technology, Virtuoso Schematic Editor, Virtuoso Layout Suite, and Analog Design Environment features.

Alternatively, you can also set the `Virtuoso_Photonics_Option` environment variable to enable the photonics features. After setting this environment variable, when you start any Virtuoso application, such as Virtuoso, stream, itkDB-based binaries, an attempt is made to check out and initialize the `Virtuoso_Photonics_Option` license (Product Number 95550). Subsequently, as you open editing cockpits, relevant product licenses will be checked out (for example, VLS-EXL for layout editing), as needed.

In both the cases, if the specified license is not found, the launch of the executable fails.

To enable Virtuoso CurvyCore technology, you would need the `Virtuoso_MultiTech_Framework` license in addition to the `Virtuoso_Photonics_Option` license. However, in case of `Virtuoso_Photonics_Platform` license, Virtuoso CurvyCore technology is enabled automatically.

Note: You can use the `dbIsPhotonicsEnabled()` API to check if the photonics features have been enabled.

License Checkout for Virtuoso MultiTech Framework

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)

License Check out for Virtuoso RF IC Design Solution

- `Virtuoso_Adv_Node_Framework`
- `Virtuoso_Schematic_Editor_XL`
- `Virtuoso_Layout_Suite_EXL`

One of the following Cadence EM Solvers:

- `EMX`
- `AXIEM_3D_Planar_EM`
- `Clarity_3DSolver`

Note: Virtuoso Layout Suite EXL includes `Virtuoso_Adv_Node_Framework` license feature.

Note: Virtuoso Layout Suite EXL includes the Electromagnetic Assistant and the interface to EM solvers.

License Check out for Virtuoso RF System Design Solution - Module Layout in Virtuoso

- `Virtuoso_MultiTech_Framework`
- `Virtuoso_Schematic_Editor_XL`
- `Virtuoso_Layout_Suite_EXL`
- `Virtuoso_RF_Option`
- `Clarity_3DSolver`

Note: Virtuoso Layout Suite EXL includes the Electromagnetic Assistant and the interface to EM solvers.

You must set one of the following shell environment variables before you launch Virtuoso:

- `Virtuoso_MultiTech`
- `Virtuoso_RF_Option`

Note: The `Virtuoso_RF_Option` shell environment variable will be deprecated and removed in a future release.

License Check out for Virtuoso RF System Design Solution - Module Layout in Allegro

- `Virtuoso_MultiTech_Framework`
- `Virtuoso_Schematic_Editor_XL`
- Allegro Package Designer Plus
- SiP Layout Option
- `Clarity_3DSolver`

You must set the following shell environment variable before you launch Virtuoso:

- `Virtuoso_MultiTech`

License Checkout for Virtuoso Power Manager

- `VIRTUOSO_POWER_MANAGER`

License Checkout When Running PVS-CV in Virtuoso

When running Physical Verification System (PVS) Constraint Validation (CV) in Virtuoso, the following license is checked out in addition to the base product licenses, which are checked out depending on the requirement of other features being used:

- `Phys_Ver_Sys_Const_Validator` (Product Number 96300)

Important

Use PVE12.1.1 version or later for running the PVS-CV feature.

License Checkout When Running Virtuoso IPVS

Virtuoso Integrated Physical Verification System (IPVS) requires the following licenses:

Basic Licenses:

- `Virtuoso_DRC_Opt`: Virtuoso Integrated Physical Verification System Option for Virtuoso Layout Suite license (Product Number: 96400)
- `Phys_Ver_Sys_Results_Mgr`: Cadence Physical Verification System Results Manager (Product Number: 96240)

Advanced Licenses:

- `Virtuoso_IPVS_Adv_Ana_Opt`: Virtuoso Integrated Physical Verification System Advanced Analysis Option (Product Number:96310)

Note: For more information, see *Licenses Requirements and Behaviors* in the *Virtuoso IPVS User Guide.*

License Checkout for Running Third-party Simulators

For running third-party (non-MMSIM) simulators in ADE product family (Explorer/Assembler) the Virtuoso® Analog Oasis Run-Time Option license (`OASIS_Simulation_Interface`, Product Number 32100) is required. Prior to usage, the simulator must be authorized for integration into the simulation environment. To integrate a commercially available simulator contact Cadence Connections (connections@cadence.com) and to integrate a proprietary simulator, contact Cadence Customer Support.

Once the integration is complete, the Oasis Run-Time Option license will be checked out and held during netlisting and simulation. The license is checked back in when the netlisting and simulation run is complete.

License Checkout When Running Express Pcell in Virtuoso

The Express Pcell capabilities are available through Cadence Design Framework II (Product Number 111) without a need to check out the VLS XL license.Abstract Generator License Usage

Abstract Generator requires the `Virtuoso_Layout_Suite_XL` or `Virtuoso_Layout_Suite_EXL` and `Virtuoso_Adv_Node_Opt_Layout` or `Virtuoso_Adv_Node_Opt_Lay_Std` licenses.

License Checkout When Running Design Planning and Analysis

`Virtuoso_Layout_Suite_EXL` and 12 GXL Tokens

Pcells and License Usage

Pcell Interoperability and Performance: The actions of saving, editing, or opening a database using Core Cached Pcells capability, from either the GUI or SKILL and from any application, causes a `Virtuoso_Layout_Suite_XL` license to be checked out.

Process Rule Editor: The use of the Process Rule Editor, or associated SKILL function, to save, edit, or open a techDB or design object requires a `Virtuoso_Layout_Suite_XL` or `Virtuoso_Schematic_Editor_XL` license to be checked out.

Summary of License Checkout Behavior

The following table summarizes the license check-out behavior for different product features:

Table 1-4 Summary of License Check-Out Behavior

Product Capability/Feature	Base Product License Required	Additional License Required	Close Application Window[1]
VSE L (Read only mode)	None	None	NA
VSE L (Edit mode)	`Virtuoso_Schematic_Editor_L`	None	License checked in
VSE XL	`Virtuoso_Schematic_Editor_XL`	None	License checked in
ADE Explorer	`Virtuoso_ADE_Explorer` or `Virtuoso_ADE_Assembler`	None	License checked in
ADE Assembler	`Virtuoso_ADE_Assembler`	None	License checked in
ADE Verifier	`Virtuoso_ADE_Verifier`	The implementations whose simulations are started from Verifier require their appropriate licenses.	License checked in
Command-Line IP Selector	AMS_environment (Product Number 70000)	None	License checked in
VLS XL	`Virtuoso_Layout_Suite_XL`	`Virtuoso_Adv_Node_Opt_Layout` or `Virtuoso_Adv_Node_Opt_Lay_Std`	License checked in
VLS EXL	`Virtuoso_Layout_Suite_EXL`	`Virtuoso_Adv_Node_Opt_Layout` or `Virtuoso_Adv_Node_Opt_Lay_Std`	License checked in

Product Capability/Feature	**Base Product License Required**	**Additional License Required**	**Close Application Window**[1]

Layout Dependent Effects (LDE) Flows	`Analog_Design_Environment_XL`
	Or
	`Analog_Design_Environment_GXL`
		`Virtuoso_Variation_Analysis_Op`	License checked in

[1] Close all windows running this application except the CIW.			
* The `Virtuoso_Adv_Node_Opt_Layout` or `Virtuoso_Adv_Node_Opt_Lay_Std` license is checked in when the last layout window is closed.			

Setting License Check Environment Variables

There are a range of license check environment variables that can be used for detailed license checkout reporting. For example, you can set an environment variable to report whether a license check is taking too long or if a particular license or token has been checked out.

These license environment variables, as shown in an example for each below, can be set in one of two ways:

1. Using `setenv` before running Virtuoso.
2. Using the SKILL function `setShellEnvVar()` during a Virtuoso session.

- `CDS_LIC_PRINT_FILTER`

  This environment variable can contain a string with the names of license features and token capabilities, for detailed license checkout reporting using the format:

  "`Feature1:Token1,Token2,...:TokenN,Feature2, ...,FeatureM`"

  If `CDS_LIC_PRINT_FILTER` is not defined, the license feature “111” will be the default.

  You can obtain a list of all registered token capabilities by examining the content of the Software Product License Management Form: Token Usage Tab.

  For example:

  ```
  setenv CDS_LIC_PRINT_FILTER "111, Virtuoso_Layout_Suite_XL,
  Virtuoso_Layout_Suite_GXL:VLS_GXL"
  setShellEnvVar("CDS_LIC_PRINT_FILTER=111, Virtuoso_Layout_Suite_XL,
  Virtuoso_Layout_Suite_GXL:VLS_GXL")
  ```

  You can also use a number of aliases with the `CDS_LIC_PRINT_FILTER` environment variable to cover multiple feature settings:

Environment Variable Value Alias	**License Tool Interprets `CDS_LIC_PRINT_FILTER` for**
`all`	All tiered licenses, including all token capabilities and license “111”.
`vlsall`	All licenses related to the Virtuoso Layout Suite family.
`adeall`	All licenses related to the Analog Design Environment family.
`vseall`	All licenses related to the Virtuoso Schematic Editor family.

When setting the CDS_LIC_PRINT_FILTER environment variable value, you can also combine application names and aliases, for example:

```
setenv CDS_LIC_PRINT_FILTER
"Virtuoso_Layout_Suite_XL,vseall,adeall":virtuoso
setenv CDS_LIC_PRINT_FILTER "vlsall,vseall,adeall,111":virtuoso
```

Note: If you set "Virtuoso_Layout_Suite_GXL:all", the license tool will interpret CDS_LIC_PRINT_FILTER as containing all token capabilities for the token license "Virtuoso_Layout_Suite_GXL".

- CDS_LIC_PRINT_TIME

  This boolean environment variable can be set to output the time (in milliseconds) to the CIW that is taken during a license check-out process. The values that can be set for CDS_LIC_PRINT_TIME are 0 (do not print time) and 1 (print time), the default being 1.

  For example:

  ```
  setenv CDS_LIC_PRINT_TIME 0
  or
  setShellEnvVar("CDS_LIC_PRINT_TIME=0")
  ```

- CDS_LIC_PRINT_ALWAYS

  Note: This boolean environment variable can be set to output messages for a particular license feature each time that license feature is checked out. The values that can be set for CDS_LIC_PRINT_ALWAYS are 0 (do not print each time) and 1 (print each time), the default being 0.

  Note: If CDS_LIC_PRINT_ALWAYS is not defined, or its value is 0, the message will be output only one time to the CIW.

  For example:

  ```
  setenv CDS_LIC_PRINT_ALWAYS 1
  or
  setShellEnvVar("CDS_LIC_PRINT_ALWAYS=1")
  ```

- CDS_LIC_TIMER_INTERVAL

  This environment variable is used to set the number of seconds required to pass, for a license search, before a timeout occurs. If the license check time exceeds the value of CDS_LIC_TIMER_INTERVAL, a message will be output to the CIW informing you that the check time has been exceeded. The default value is 15 seconds.

  For example:

  ```
  setenv CDS_LIC_TIMER_INTERVAL 10
  or
  setShellEnvVar("CDS_LIC_TIMER_INTERVAL=10")
  ```

- CDS_LIC_PRINT_TYPE

  This environment variable can be set to output all license call related messages (information and warning) to the CIW, or only the warning messages. If the value of CDS_LIC_PRINT_TYPE is set to "warn", then only warning messages will be output to the CIW. If set to "all", both information and warning messages will be output. The default is "warn".

  For example:

  ```
  setenv CDS_LIC_PRINT_TYPE "all"
  or
  setShellEnvVar("CDS_LIC_PRINT_TYPE=all")
  ```

- CDS_LIC_TIMEOUT_DIALOG

  This environment variable is used to display the timeout dialog when the license checkout time of a license feature, as defined by CDS_LIC_PRINT_FILTER, exceeds the timer interval specified by CDS_LIC_TIMER_INTERVAL. The value of CDS_LIC_TIMEOUT_DIALOG can be either "1" (which will display the dialog) or "0" (which will not display the dialog). The default value is "0".

  Note: For the initial 111 license, the timeout dialog will be displayed even when there is no license 111 in CDS_LIC_PRINT_FILTER and irrespective of CDS_LIC_TIMEOUT_DIALOG being set.

- CIC_ENABLE_LIC_PERF

  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.

  For example:

  ```
  setenv CIC_ENABLE_LIC_PERF "limit=0.08 pings=4 interval=0.5"
  ```

  Where,

  limit = [time] is the threshold value for checkout time (in seconds).

  pings = [count] is the number of pings sent to the license server. When this option is set, a specified number of pings will be sent to each license server (defined in CDS_LIC_FILE), when the license request time exceeds the threshold value. The default is 0, which means no pings are sent to the license server.

  interval = [time] defines the time interval (in seconds) between pings.

  The results from the ping commands are printed to the CIW.

Advanced Features and Token Licenses

The advanced technology features in the XL and EXL tiers are accessed by tokens or multi-feature licenses which are defined by a multi-feature license model.

Note: See also Tracking Token Licenses.

A token is one copy of a multi-feature license. Each feature requires a specific number of tokens. GXL multi-feature licenses are Job-based (J) licenses which are not shared between processes.

GXL Multi-Feature License Tokens

A specific number of tokens is required to access the features in either of these tiers:

- `Virtuoso_Layout_Suite_GXL`

Note: VSE does not have a GXL product tier.

Table 1-5 on page 32 lists the tokens needed for the VLS GXL multi-feature licenses.

If all of your license tokens are currently in use, or you have insufficient tokens to access a particular feature, additional tokens can be obtained by contacting your local Cadence representative.

Table 1-5 Tokens per Feature (VLS GXL)

Features	Tokens Required	Version
Analog_Auto_Placer	8	6.18
Analog_Auto_Placer_Adv_Node	24	6.18
Cell_Planner	4	6.18
Design_Planning_Analysis	12	6.18
Digital_Auto_Placer	2	6.18
Digital_Auto_Placer_Adv	24	6.18
Floorplanning	4	6.18
Integrated_Short_Locator	3	6.18
Layout_Migrate	20	6.18
Layout_Yield_Optimize	8	6.18
Module_Generator	2	6.18
Slotting	4	6.18
Space_based_Router	12	6.18
Symbolic_Placement_Devices	2	6.18
VCAR	8	6.18
VLS_GXL	4	6.18
VPLGen	1	6.18

Important Points to Note

- All of the features listed in Table 1-5 can be started in the XL or EXL window.
- VPLGen: Using any of the Virtuoso Parameterized Layout Generator commands to create, edit, or update a VPLGen causes one VLS GXL token to be checked out. This GXL token remains checked out for the duration of the VLS XL or EXL session.
- Slotting does not require any GXL tokens in the Layout EXL tier.

Virtuoso Space-based Router Tokens

Some Virtuoso Space-based Router features, such as viewing, navigation, and DRC checking, can be run from Virtuoso Layout Suite XL or EXL without additional licensing. Space_based Router tokens are required to perform routing and other tasks that modify the routing and routing environment, based on the number of nets in the design. When you invoke the first command that requires Space_based Router tokens, the tokens are checked out and are not released until you exit Virtuoso Space-based Router.

Some Virtuoso Space-based Router commands can use multiple processors in one workstation to accelerate processing. Additional Space_based Router tokens are required for the additional processors. The below table shows the number of additional tokens that you will need, based on the number of processors, or threads, that you want to use. License tokens can be reserved globally for use by all commands in a session or can be checked out as needed for a command. Tokens Per Feature: VSR for Automatic Routing (Space_based_Router)

Number of Nets	Number of Processors	Number of Tokens
0 to 1	N/A	VLS XL license required
2 to 9999	1	12
2 to 9999	2 to 4	24 (+12)
2 to 9999	5 to 8	36 (+12)
2 to 9999	9 to 12	48 (+12)
2 to 9999	each additional 4 processors (over 4 processors)	+ 12 tokens
10,000 to 249,999	1	24
10,000 to 249,999	2 to 4	48 (+24)
10,000 to 249,999	5 to 8	72 (+24)
10,000 to 249,999	9 to 12	96 (+24)
10,000 to 249,999	each additional 4 processors (over 4 processors)	+ 24 tokens
>= 250,000	1	48
>= 250,000	2 to 4	72 (+24)
>= 250,000	5 to 8	96 (+24)
>= 250,000	9 to 12	120 (+24)

>= 250,000	each additional 4 processors (over 4 processors) + 24 tokens

For example, if your design has 100,000 nets and you want to run Virtuoso Space-based Router using six (6) processors, you will need 48 Space_based Router tokens, 24 for the net count plus 24 for the thread count.

Tracking Token Licenses

The following methods can be used to track token license activity.

- lmstat Command on page 35
- Software Product License Management Form on page 37

lmstat Command

The `lmstat` command can be used to track token license activity from the command line. It reports activity for one license server only. Use the following arguments with the `lmstat` command.

Option	Description
`-a`	Display everything
`-c license_file`	Use "license_file" as license file
`-f [feature_name]`	List usage info about specified (or all) features
`-i [feature_name]`	List info about specified (or all) feature(s) from the increment line in the license file
`-S [DAEMON]`	Display all users of DAEMON licenses
`-s [server_name]`	Display status of all license files on server node(s)
`-t timeout_value`	Set connection timeout to "timeout_value"
`-v`	Display FLEXlm version, revision, and patch
`-old`	Allow communications with an old server that uses communications version 1.2 or earlier
`-help`	Print this message

Example

1. Type the following in a shell window:

   ```
   lmstat -c <license.file> -f "<feature.name>"
   ```

   For example,

   ```
   lmstat -c sj234flex4 -f "Virtuoso_Layout_Suite_GXL"
   ```

2. Review the output.

   In this example, `designer1` on host `lne2` has the `VLS_GXL` license checked out (4 tokens).

   ```
   Users of Virtuoso_Layout_Suite_GXL:
    (Total of 400 licenses issued;  Total of 4 licenses in use)


     "Virtuoso_Layout_Suite_GXL" v6.100, vendor: cdslmd floating license


   designer1 lne2 VLS_GXL (v6.100) (server 54), start Tue 4/17 2:40, 4 licenses
   ```

   For more information about the `lmstat` command, see the *Cadence License Manager*.

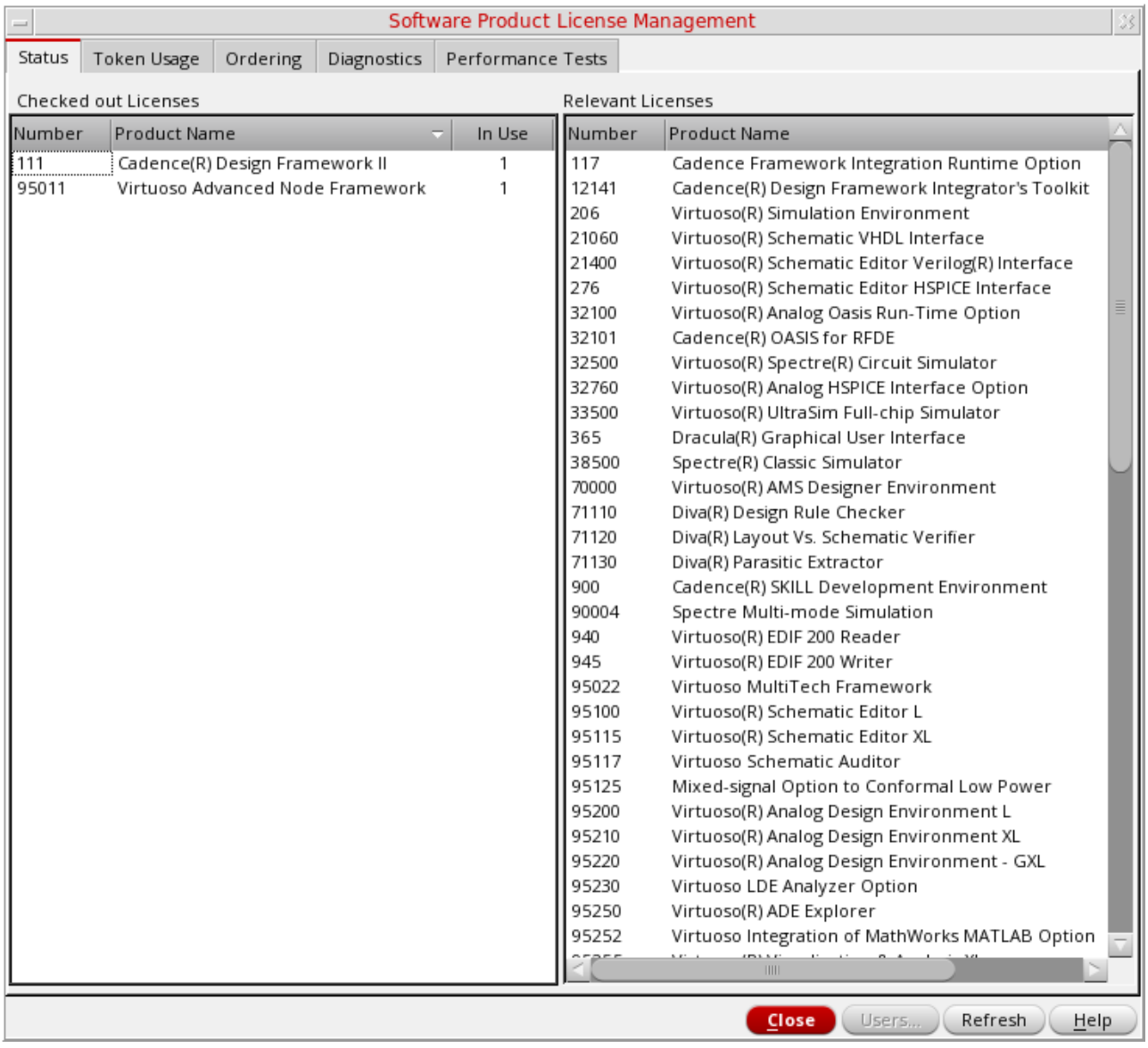
   Note: The `tokenPrintCapabilities` SKILL function that printed the details of registered token licenses, including the number of tokens required for each capability and whether or not it is in use, no longer prints the token details. It will be deprecated in a future release. For more information on `tokenPrintCapabilities`, refer to the *Virtuoso Design Environment SKILL Reference*.

Software Product License Management Form

You can use the Software Product License Management form to manage the licensing activity.

To open the form:

➡ Select *Options – License* in the CIW. The Software Product License Management form is displayed.



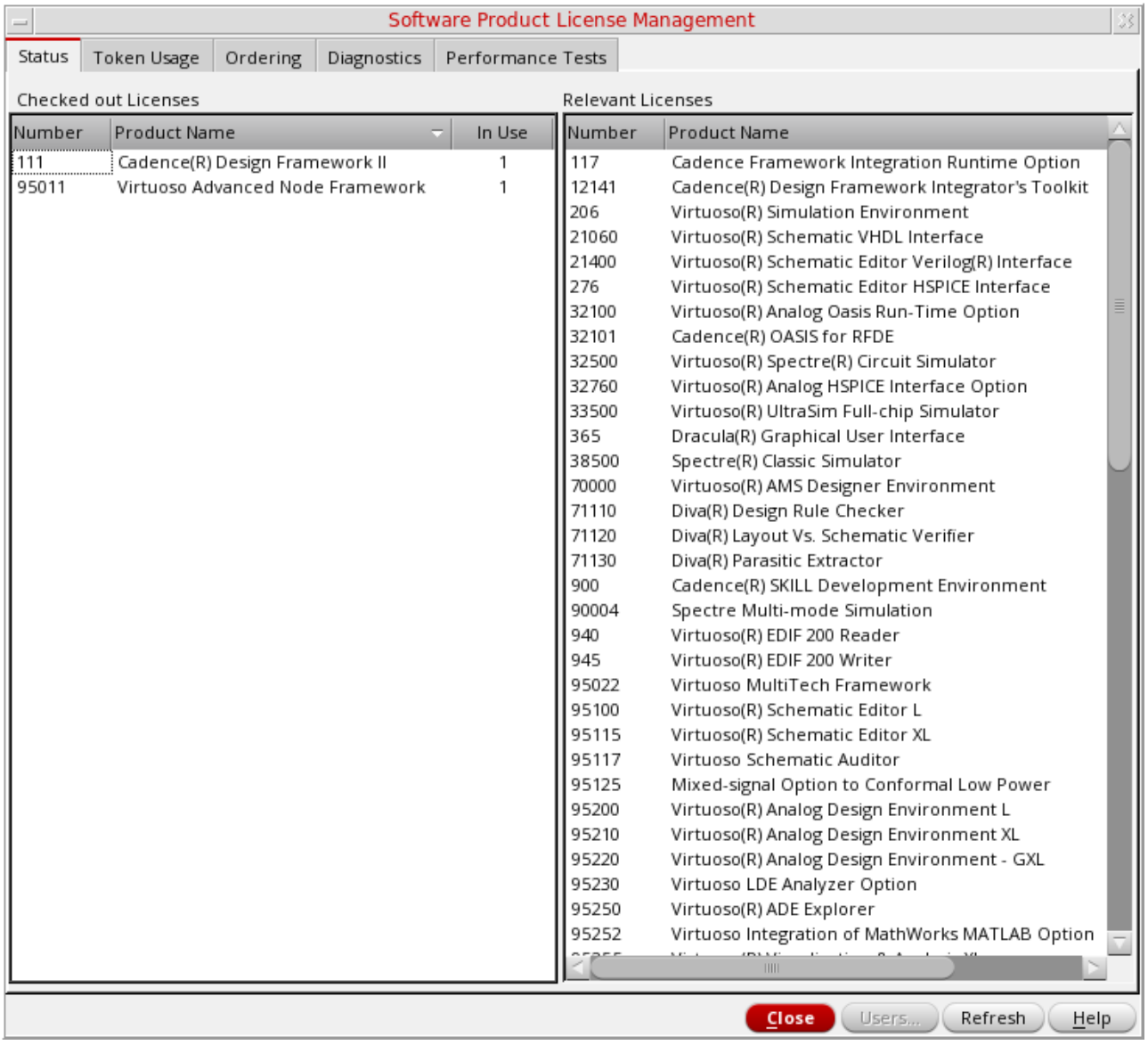
The form contains the following tabs to handle the licensing process:

- **Status**: Displays the license status in two panes – *Checked Out Licenses* and *Relevant Licenses*. For more information, see the Software Product License Management Form: Status Tab.
- **Ordering**: Contains the options related to license ordering. For more information, see the Software Product License Management Form: Ordering Tab.
- **Diagnostics**: Contains the options for configuring the license diagnostic settings. For more information, see the Software Product License Management Form: Diagnostics Tab.
- **Performance Tests**: Contains the options to run a diagnostic test that determines the license-checkout performance on a feature-by-feature basis and across each license server. The tests can be looped and repeated over a period of time to calculate the average, median, max, and min times for each checkout. For more information, see the Software Product License Management Form: Performance Tests Tab.

Software Product License Management Form: Status Tab

To open the Software Product License Management form - Status tab

➡ Select *Options – License* in the CIW. The Software Product License Management form is displayed with the *Status* tab selected by default.

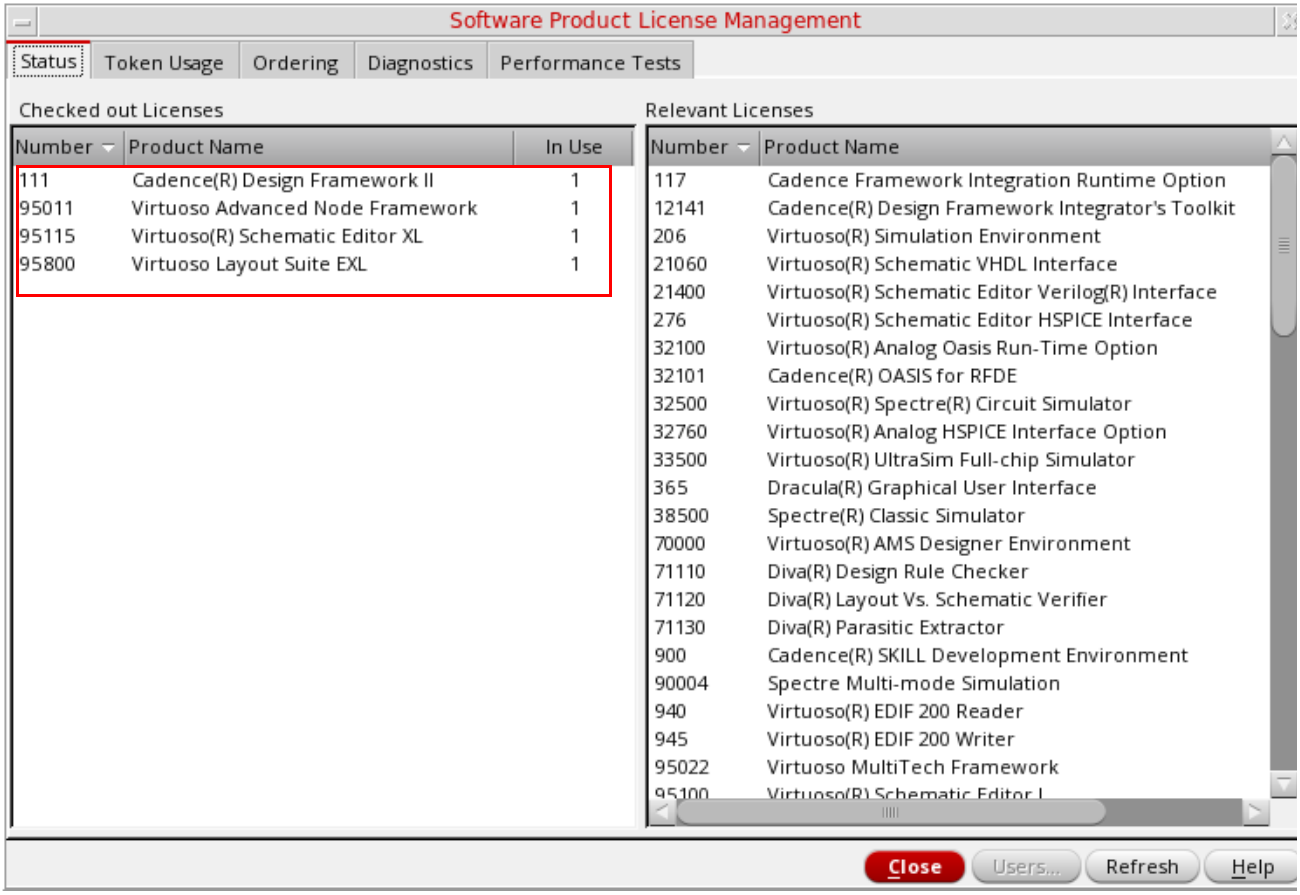


Checking the License Status

You can check the license status using either of the following ways:

- Use the *Checked Out Licenses* pane of the *Status* tab.

  This pane lists the licenses that are currently checked out, and their count.



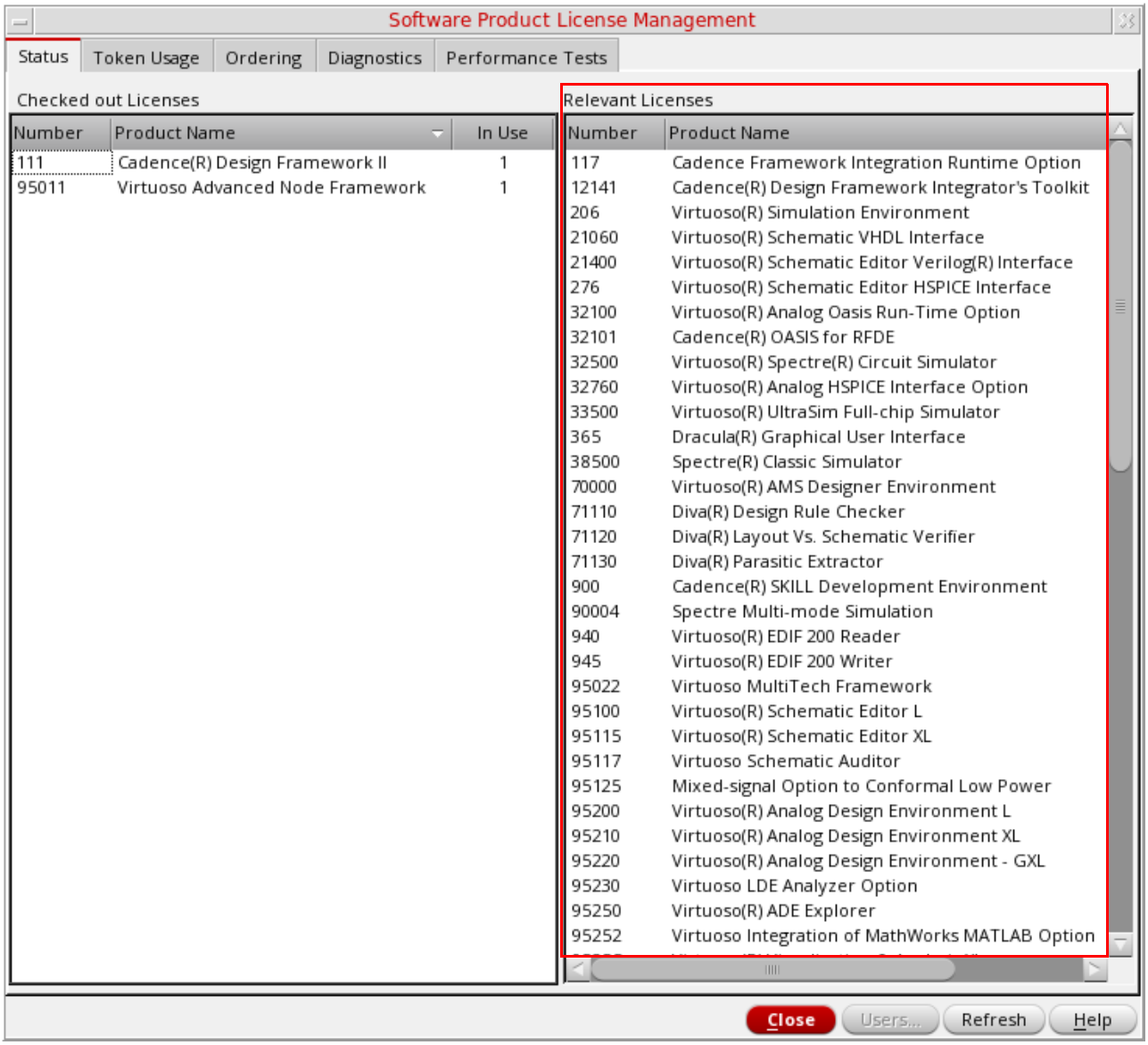
- Run the `lbuiGetCheckedOutLicenses` SKILL function to return a list of checked out licenses, which appear in the *Checked Out Licenses* pane.

  The list contains each license feature name and the number of licenses (tokens) checked out. In addition, the license activity information gets recorded in the `CDS.log` file. This log file can be used later for creating reports to analyze the license usage.

  For more information on `lbuiGetCheckedOutLicenses`, refer to the *Virtuoso Design Environment SKILL Reference*.

Viewing Relevant Licenses

The *Relevant Licenses* pane on the *Status* tab lists the DFII licenses relevant to all of the tools incorporated into the workbench.

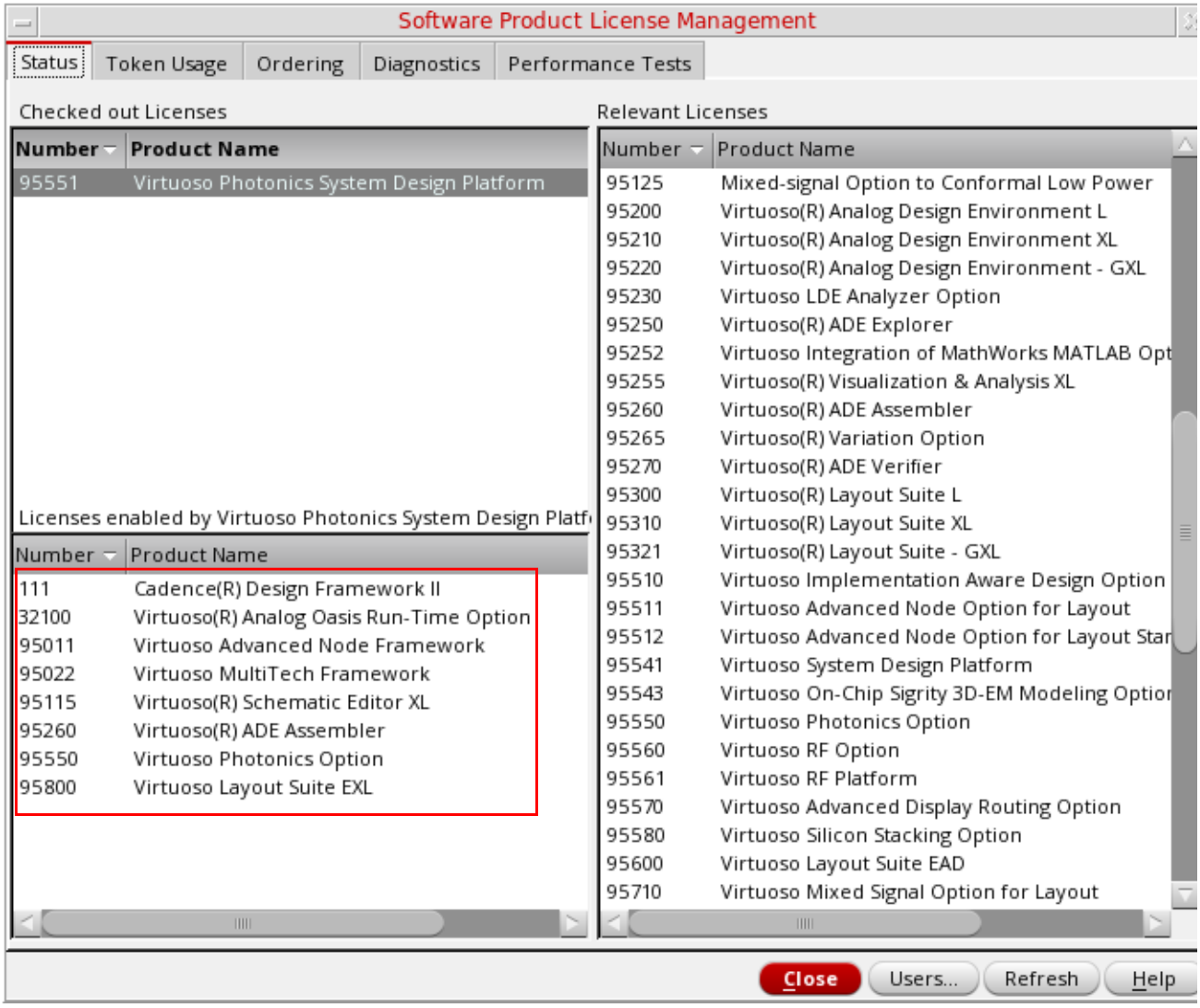


Note: You can sort the licenses displayed in the *Checked out Licenses* and *Relevant Licenses* panes by clicking the arrow ( ) next to the *Number*, *Product Name*, or *In Use* column headers.

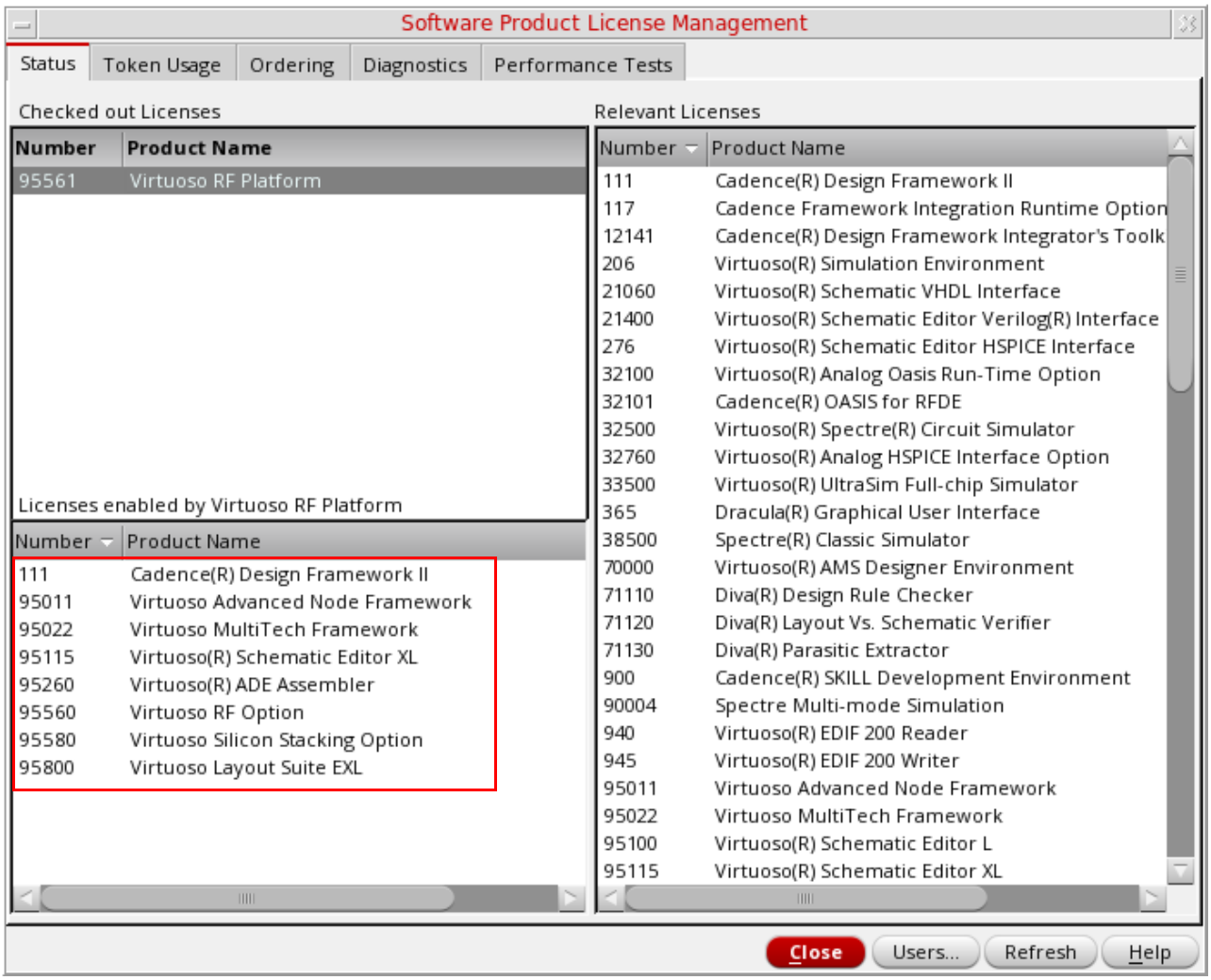
Viewing Solution-based Licenses

Licenses checked-out for a solution can be viewed when you set the specific environment.

The following licenses are checked out when the Virtuoso Photonics solution is enabled:



The following licenses are checked out when the Virtuoso RF solution is enabled:



Adding Custom Products to the Status Tab

To add custom products and their license information to the *Status* tab, set the `CDS_PRODUCTS_DFII_LIC_UI` shell variable. The `CDS_PRODUCTS_DFII_LIC_UI` variable points to the file where the information about the custom products and license is stored in the following format:

```
<Product Number><separator><Product Name><separator><License Feature Name><endl>
where,
<separator> is ':'
and
is '\n'
```

Here, the definition for the custom product and license information use this format in one line and any blank lines are ignored.

Refreshing the License Status

To update the *Checked Out Licenses* and *Relevant Licenses* panes, click *Refresh*.

This action will update the *Checked Out Licenses* and *Relevant Licenses* lists to show the current list of products being run, and the respective license count, along with any changes to the available license list since the initial access of the Software Product License Management form.

Displaying the License Usage Information

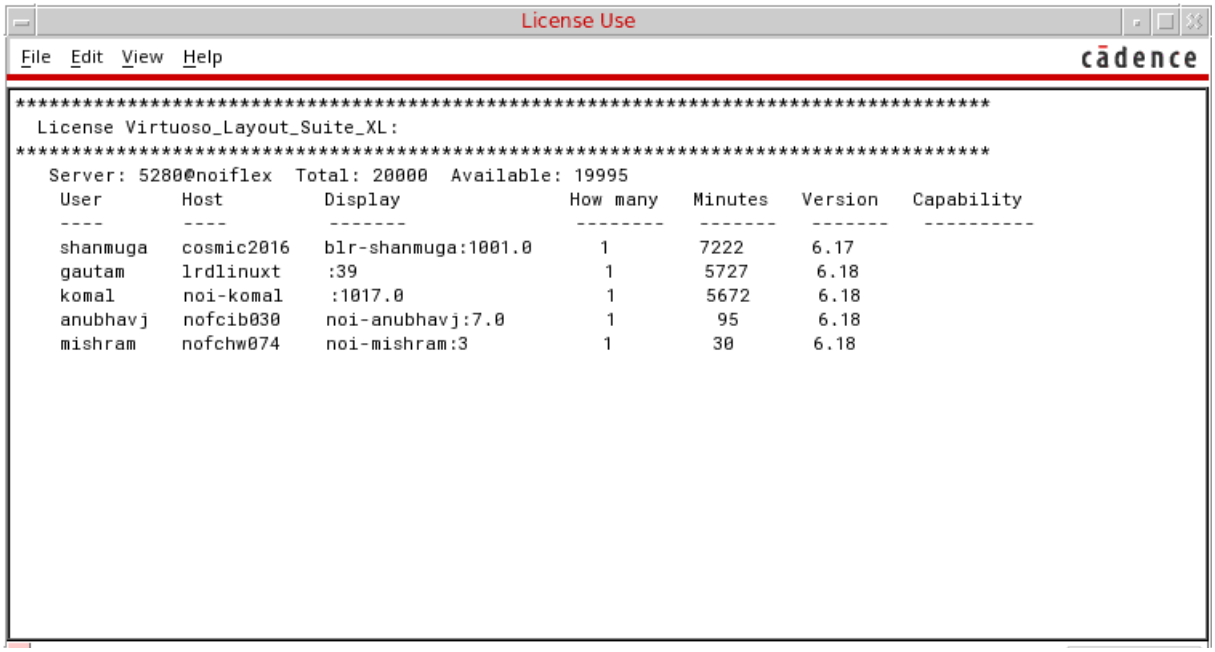
You can display the license status using either of the following ways:

- Select a license from one of the lists on the Software Product License Management Form and click *Users* to display the License Use form.

  Note: The *Users* button is disabled by default. It gets enabled when you select a license in the Software Product License Management form and click *Refresh*.

  The License Use form displays the license usage information (for example: user logins, host names, display time, number of licenses, and license version) for the selected licenses, independent of the license-checkout status. Information in this form is read from the license servers that are defined in the `CDS_LIC_FILE` and is extracted from more

than one license server. If you do not have any licenses for a selected feature in any license file, the License Use form will display total and available licenses as “0”.



Note: You can display the license usage information of more than one license in the License Use form. To do so, hold the CTRL key to select multiple licenses or hold the SHIFT key to select a contiguous range of licenses before clicking *Users*.

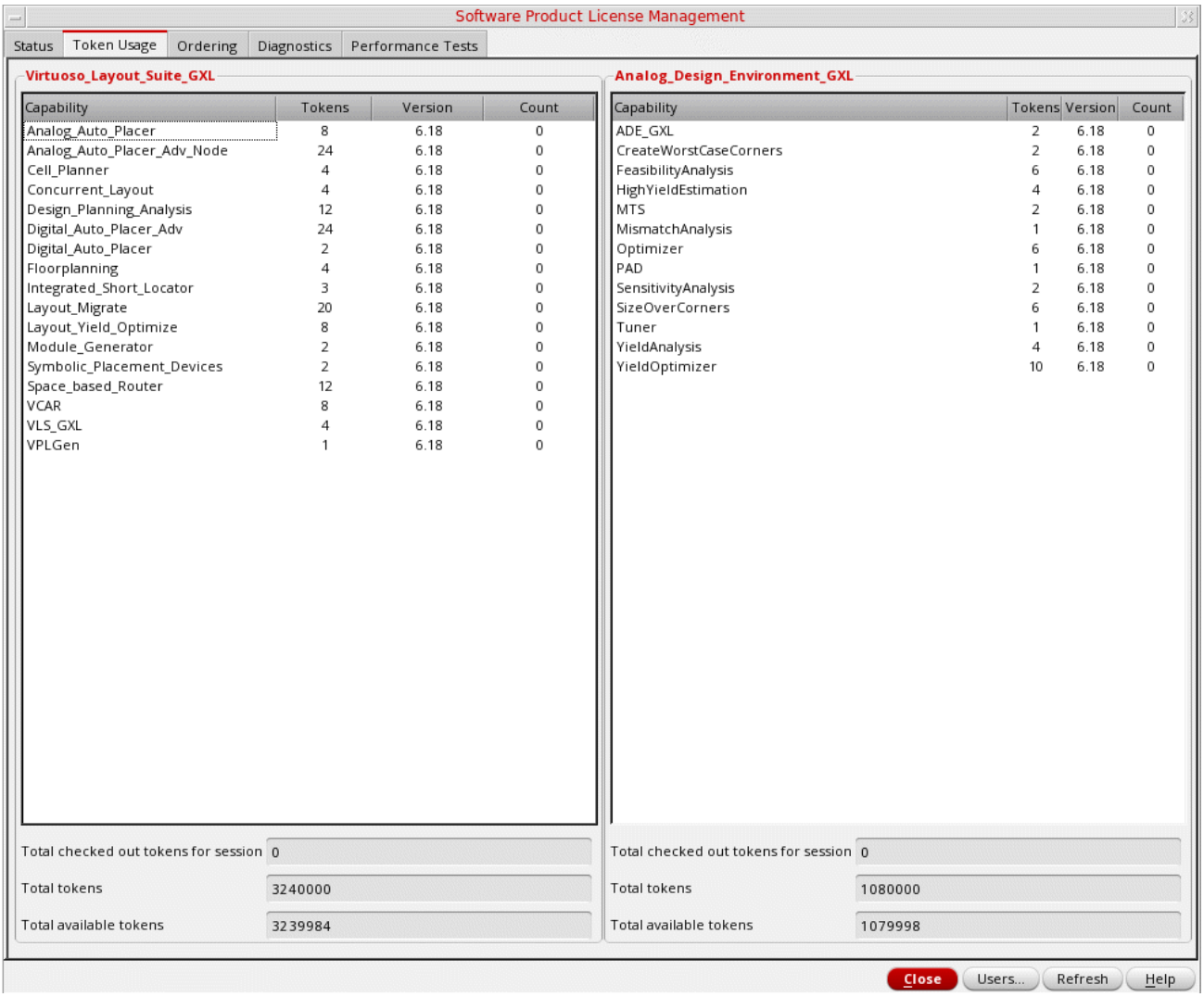
Hold the CTRL or SHIFT key and click to deselect any previously highlighted licenses before choosing a new license for viewing in the License Use form.

- Run the `lbuiPrintLicenseUseReport` SKILL function to display the license usage report (for example: user logins, host names, display time, number of licenses, and license version) in the CIW or the specified output log file.

  For more information on `lbuiPrintLicenseUseReport`, refer to the *Virtuoso Design Environment SKILL Reference*.

Software Product License Management Form: Token Usage Tab

The *Token Usage* tab displays the number of tokens required to run each capability of the Virtuoso Layout Suite GXL and Analog Design Environment GXL licenses, and the respective usage status.



Note:

- A license does not need to be selected to view the token-usage information because token-license usage is always provided in the GXL product tier.
- When an application that requires tokens is being launched, the *Total Tokens* and *Total available tokens* fields on the Token Usage form display the message “Collecting data from server”. This message is replaced with the token information once the launching process of the application completes.

- The L and XL product tiers have user, host, and display (UHD) based licenses. If you are using the same host and the same display, you can use the same L or XL license for multiple sessions of an application. However, the GXL tier uses job-based licenses, which are not shared. It means that each run of a GXL feature will check out the required number of tokens and multiple runs will require multiple sets of tokens. For more information on tokens, and licensing in general, see Advanced Features and Token Licenses

Software Product License Management Form: Ordering Tab

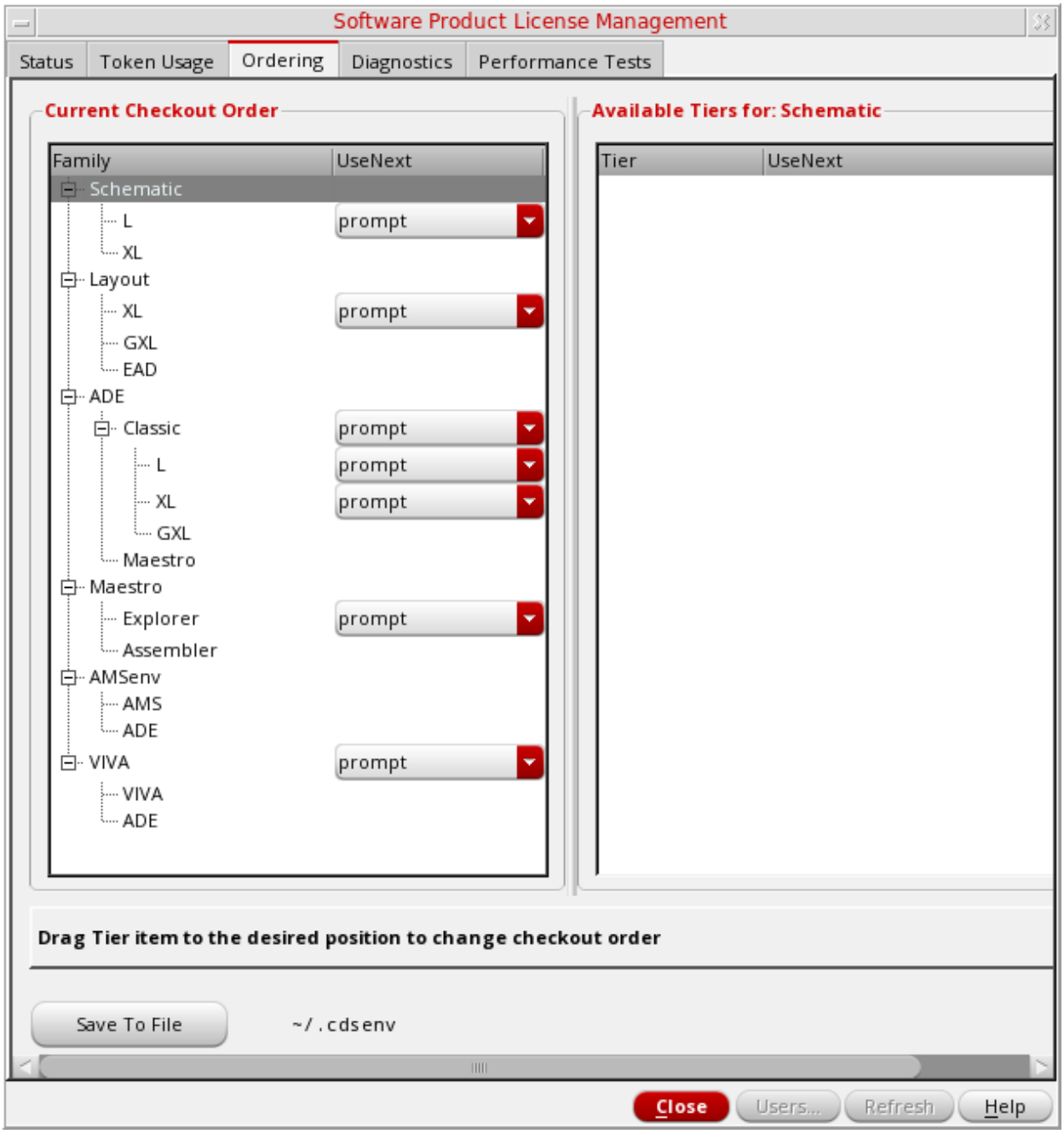
The *Ordering* tab provides an interface for defining the license checkout order of the Virtuoso application tiers.

Each of the product families (Schematic, Layout, Layout Option, ADE, and VIVA) contains applications in a tiered packaging structure. Each higher-tier application provides additional features with more automated design assistance.

Only one license per product family tier is checked out at a time. Lower-tier applications require either their own tier license or a higher-tier license to run. For example, you can run VLS XL and VLS EXL in the same session with only a VLS EXL license checked out. If an attempt to check out the license of a lower-tier application fails, the application looks for a higher-tier license. You can set the license checkout order using the *Ordering* tab, as shown below.

To view and use the displayed controls and buttons:

1. Click the *Ordering* tab in the Software Product License Management form. The *Ordering* tab has two panes: *Current Checkout Order* and *Available Tiers*.



- *Current Checkout Order:*

  Displays the list of product families along with their current checkout order and *UseNext* settings.

Family: Lists the names of the product license families. For example, Schematic, Layout, Layout Option, ADE, AMSenv, and VIVA.

To change the checkout order of a product family, click and drag the required tier up or down to move it to the desired location.

Note: The GXL and EXL tiers in the Layout family cannot be separated; they must appear together in the checkout order.

Note: EAD licenses will automatically forward-enable Layout EXL.

UseNext: Displays the *UseNext* setting, which controls whether to check out the next available license for Virtuoso tier-based applications when the license required for a requested application is not available.

If a tiered application has an associated *UseNext* setting, a drop-down list appears next to its name. If you reorder a tier within the product family, the *UseNext* drop-down moves along with it.

The *UseNext* drop-down list provides the following options:

- *prompt:* Confirm before checking out a higher-tier license.
- *always*: Always check out the higher-tier license (as per selected checkout order).
- *never*: Never check out the higher-tier license and display an error message instead.

When *prompt* is set as the *UseNext* drop-down value and the license for a requested application is not available, the Next License dialog box displays, which lets you check out a higher tier license, subject to availability. Four options are offered: *Session*, *Skip*, *Always*, or *Never*.

- *Session*: Checks out a higher-tier license, if available, for the current session. If a higher-tier license is unavailable, the Next License dialog box will only be displayed once. However, a new virtuoso session will cause the dialog box to display again.

  The *Session* option does not override the setting in the local .cdsenv file and updates only the internal or in-memory value of the corresponding `UseNextLicense` .cdsenv variable. Because of this, the *Session* response is applicable for the current session only. However, if you want to restore the Next License dialog box settings in the same session, reset the `UseNextLicense` .cdsenv variable to `prompt` using the `envSetVal` function in the CIW or select *prompt* from the *UseNext* drop-down list.

- *Skip*: Skips the next license. For example, if L license is not available, and you click *Skip* in the Next License dialog box, the XL license will be skipped and an attempt to check out the GXL license will be made.
- *Always:* Virtuoso will always try to check out the next license, if the requested license is not available. The Next License dialog box is not displayed again when virtuoso is restarted.
- *Never*: Virtuoso will never try to check out the next license, if the requested license is not available. The Next License dialog box is not displayed again when virtuoso is restarted.

The *Always* and *Never* options are stored in the following environment variables in ~/.cdsenv as soon as they are set.

For example,

```
license VLSXL_UseNextLicense string "prompt"/"always"/"never"
license VSEL_UseNextLicense string "prompt"/"always"/"never"
license ADEL_UseNextLicense string "prompt"/"always"/"never"
license ADEXL_UseNextLicense string "prompt"/"always"/"never"
```

If a license is not available warning messages will be displayed in the CIW.

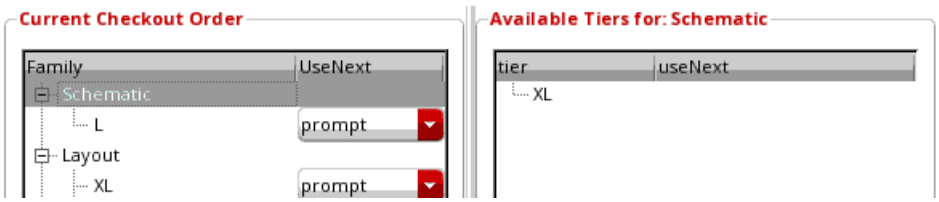
Note: The settings selected in the *Current Checkout Order* pane are reflected in the `.cdsenv` variables *`CheckoutOrder`* and *`UseNextLicense`*.

- *Available Tiers*:

By default, the *Current Checkout Order* pane displays all applicable tiers for each product family. However, you can choose to deselect a tier and remove it from the current checkout order list by dragging and moving it to *Available Tiers*.

Note: If there is only one tier in the family tree, you are not allowed to remove it from the *Current Checkout Order* list by dragging and moving it to *Available Tiers* list, because a family tree cannot be empty.

When you select a product family in the *Current Checkout Order* pane, you can see the current checkout order of its tiers in the left pane and the list of available (applicable) tiers in the right pane. For example, in the image below, the XL tier for the product family Schematic has been moved to the *Available Tiers for: Schematic* pane.

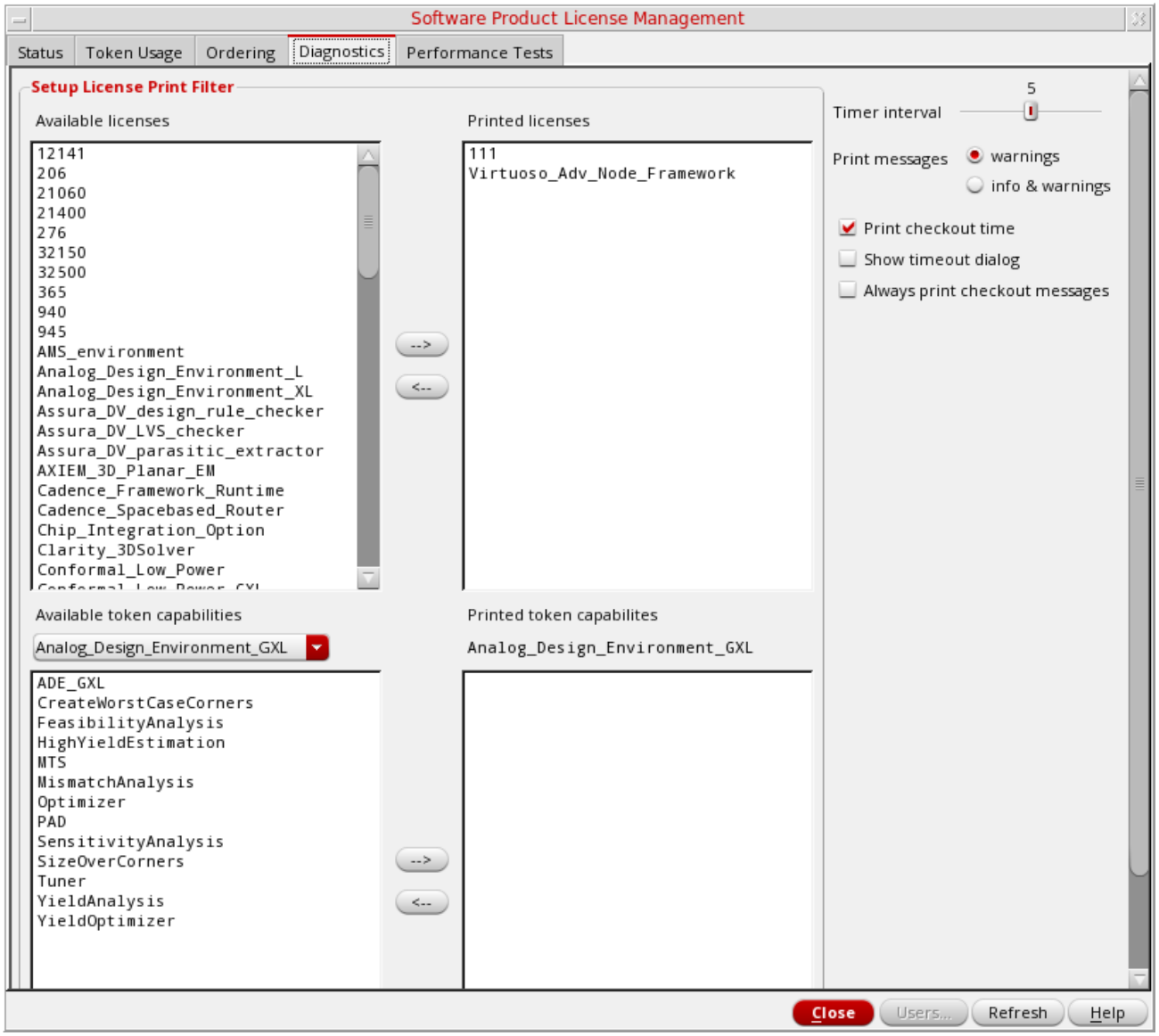


2. Click *Save to File* to save the selected options in the `.cdsenv` file. You can also edit your preferences in the `.cdsenv` file.

Software Product License Management Form: Diagnostics Tab

The *Diagnostics* tab provides an interface for setting the parameters needed for license-checkout reporting. For example, you can set these parameters to report whether a license check is taking too long or if a particular license or token has been checked out. By using the options available in this tab, you can set the values of the license diagnostic variables without needing to remember the shell variable names.

1. Click the *Diagnostics* tab in the Software Product License Management form to view and use the displayed panes and controls.



The Diagnostics Tab Panes

- *Available licenses*: Lists all available licenses.
- *Printed licenses*: Displays a list of licenses selected for the detailed license-checkout report. When a license listed in the *Printed licenses* pane is checked out, a warning message will be output to the CIW.

  You can select an appropriate license name from *Available License* pane and click the right arrow (-->) to move it to the *Printed License*s pane.

  Note: License feature “111” is printed by default in the *Printed licenses* pane as “111” is the required license for Virtuoso applications.
- *Available token capabilities*: Displays the list of available token capabilities for the selected license type. You can list the token capabilities of a particular license by selecting the license name from the drop-down list box provided above this pane.
- *Printed token capabilities:* Displays a list of token capabilities selected for the detailed license-checkout report. When a token capability listed in the *Printed token capabilities* pane is checked out, a warning message will be output to the CIW.

  You can select an appropriate token capability from the *Available token capabilities* pane and click the right arrow (-->) to move it to the *Printed token capabilities* pane.

  Note: The selected printed licenses and token capabilities are used to set the value of the `CDS_LIC_PRINT_FILTER` shell variable.

The Diagnostics Tab Controls

This tab also contains the following controls:

- *Timer interval (Shell variable –* `CDS_LIC_TIMER_INTERVAL`*)*: Use this slider to set the timeout interval (in number of seconds) for license search.
- Print messages (Shell variable – `CDS_LIC_PRINT_TYPE`): Use this option to print all license call-related messages. You can choose to display only warning messages, or both information and warning messages in the CIW.
- *Print checkout time (Shell variable –* `CDS_LIC_PRINT_TIME`*):* Select this check box to print the time lapse during a license check-out process (in seconds).
- *Show timeout dialog (Shell variable –* `CDS_LIC_TIMEOUT_DIALOG`*):* Select this check box to display the timeout message box when the license-checkout time of a license feature exceeds the specified timer interval. (`CDS_LIC_TIMER_INTERVAL`*)*.

- *Always print checkout messages* (Shell variable – *CDS_LIC_PRINT_ALWAYS*): Select this check box to print a checkout message when a particular license feature is checked out.

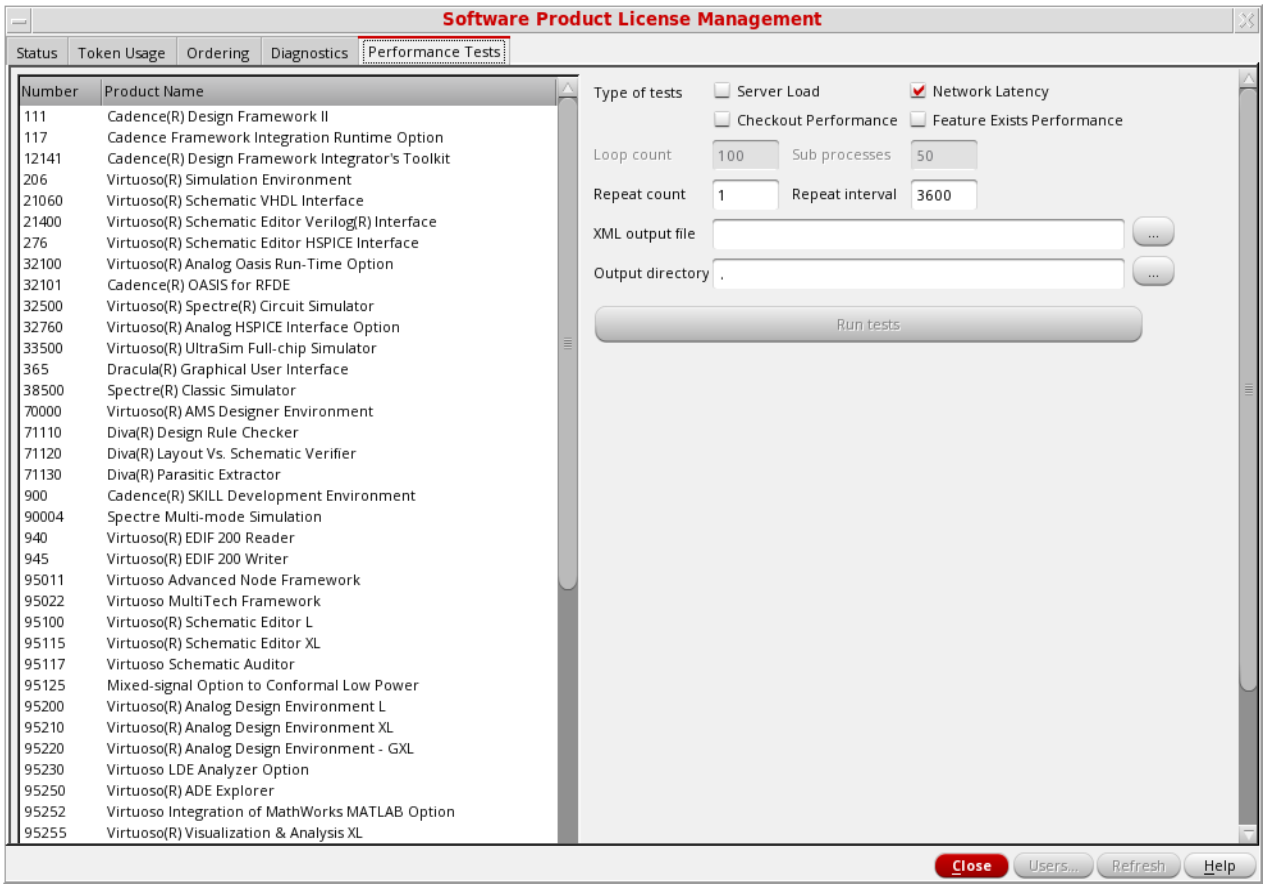
  Note: The settings specified in the above controls update the values of the corresponding shell variables.

Software Product License Management Form: Performance Tests Tab

In an environment where you have licenses served across multiple servers spread over a network, there can be significant delays during the application start up because Cadence applications search all the license servers to acquire a license.

In such situations, you can run a diagnostic test from the *Performance Tests* tab to test the license checkout performance on a feature-by-feature basis and across each server. This tab is an interface to the `perf_test` utility.

Click the *Performance Tests* tab in the Software Product License Management form to use the displayed controls and buttons.



Testing the License Checkout Performance

For testing the network latency, selecting the Cadence Design Framework II license is sufficient. However, if you suspect that a particular license feature is taking more time to check out, you can select the required license feature name for testing.

To test the license-checkout performance of a license feature:

1. Select a license feature name from the left pane. This pane lists all the DFII license features that are available for testing. You can select multiple license features for testing.
2. Select an appropriate check box against *Type of tests* to specify the type of test that you want to perform. The available options are:
   - *Server Load*: To determine the performance of a license server for a given number of clients (or subprocesses).

     Note: You can select multiple license features for the server load test. However, the test will be performed for the first license feature only.
   - *Network Latency*: To determine the license-checkout performance on a feature-by-feature basis and across each server.
   - *Checkout Performance*: To display the loop of checkout calls checking out the same feature with additional (upgrade) license for each iteration of the loop.

     Note: The server maintains one job with a number of licenses over the course of this test. The first license server will be used if there are more than one servers specified in the license path.
   - *Feature Exists Performance*: To test if the feature exists on the license server(s) specified in the license path (`CDS_LIC_FILE`).
3. Specify the loop count for checkout calls in the *Loop Count* field. This value is used for the checkout performance and server load tests; the default is 100.
4. Specify the number of sub processes for the server in the *Sub processes* field. This value is used for the server load test; the default value is 50.
5. Specify the test repeat count to rerun the tests <count> number of times in the *Repeat count* field.The default value is 1.
6. Specify the test repeat interval in the *Repeat interval* field. Valid only if *Repeat count* is specified. Reruns the tests every <interval> seconds. The default value is 3600 seconds (1hour).
7. Specify the XML file name in which you want to save the output results in the *XML output file* field.

8. Specify the directory name where you want to save the output results in the *Output directory* field.

   Note: If the specified directory does not exist, it is automatically created. When the output directory is specified, the XML file is saved in this directory.

9. Click *Run tests* to start the license-checkout performance test. After test completion the PerfTest Output window displays the test results.

   Note: While the performance test is running, license feature names in the left pane are temporarily disabled. These licenses become available only after the previous test run is complete.

Product Tier Features

The following table shows the feature details included in each application.

Feature	VSE L	VSE XL	VLS XL	VLS EXL
New Common Environment	X	X	X	X
New Icon Style	X	X	X	X
Multi-Tab Support	X	X	X	X
Updated Pulldown Menus	X	X	X	X
Bookmarks & History	X	X	X	X
Window Config Support	X	X	X	X
World View Assistant	X	X	X	X
Search Assistant	X	X	X	X
Property Editor Assistant	X	X	X	X
Navigator Assistant	X	X	X	X
Task Assistant	X	X	X	X
Basic Polygon Editing			X	X
Automatic Abstract Generation			X	X
DRD Editing			X	X
Constraint Browser		X	X	X
SystemVerilog Support		X		
Schematic Model Generator (wreal)		X		
Native interactive wire editing			X	X
Point-to-point routing			X	X
Finish wire			X	X
Push in interactive wire editing			X	X
Advanced DRD rule checks in interactive routing/wire editing			X	X

Feature	VSE L	VSE XL	VLS XL	VLS EXL
Advanced rule checks in batch checking			X	X
Interactive bus routing			X	X
Interactive special signal wire editing			X	X
A & D Device Placer			X*	X
Modgens			X*	X
Cell Planning			X*	X
Floorplanning			X*	X*
Custom Digital Placer			X*	X*
Spaced-Based Auto Router			X*	X*
* Base product plus options.				

ADE Product Family Features

The following table lists the features available in the ADE product family:

Feature	ADE Explorer	ADE Assembler	95265 (VVO)
Single Test-bench	X	X	
Simple Parametric Analysis	X	X	
Scripting support with OCEAN	X	X	
Fully waveform calculator and display environment	X	X	
Full Cadence MMSIM Support	X	X	

Feature	ADE Explorer	ADE Assembler	95265 (VVO)
3rd Party Simulator Support w/ OASIS	X*		MMSIM only
Distributed simulation (LSF, SGE, etc.)	X	X	
Parasitic Resimulation w/ Extracted View	X	X	
Dependent Expressions	X	X	
Support for Matlab Simulink	X	X	
Reliability Analysis (RelXpert)	X	X	
Configurable Assistants	X	X	
New Assertions and Device Checks Support (Spectre ONLY)	X	X	
Device Checks Support over Sweeps/Corners (Spectre ONLY)	X	X	
Instance and Global Variable Support	X	X	
Corners Analysis	X	X	
Monte Carlo Analysis	X	X	
Generate Documents	X	X	
HTML Specification Sheets	X	X	
Corners from Statistical Runs	X	X	
Data View/History Assistant		X	
Parameterized config views		X	

Feature	ADE Explorer	ADE Assembler	95265 (VVO)
Support for pre-calibration scripting		X	
Compare Histories, Specs and Points		X	
Multiple Test-bench support		X	
Layout Dependent Effect Analysis		X + 95510	
Partial Layout device extraction/EAD resimulation		X + 95510	
Manual Parasitic Estimation & Stitching	X	X	
Standard Sensitivity Analysis	X	X	
Multiple Technology Sim Support (MMSIM only)	X	X	
Manual Circuit Tuning		X	
Local Circuit Optimization		X	
Global Circuit Optimization		X	
Design Migration for Schematics		X	
Auto-sizing over Many Corners		X	
Worst Case Corners Analysis		X	
High-sigma Yield Estimation (4, 5, 6)	X + 95265	X + 95265	X

Feature	ADE Explorer	ADE Assembler	95265 (VVO)
MC Auto-Stop: Fast Yield Verification w/ sample reordering	X + 95265	X + 95265	X
MC Auto-Stop: Sensitivity Accuracy	X + 95265	X + 95265	X
MC Auto-Stop: K-Sigma Corners	X + 95265	X + 95265	X
K-Sigma Corners from MC results	X + 95265	X + 95265	X
Statisitical Mismatch Contribution	X + 95265	X + 95265	X
Assisted Yield Improvement Flow		X + 95265	X
Statistical Sensitivity Analysis		X + 95265	X
New Virtuoso ADE Suite Features ONLY			
Interactive Circuit Tuning with Spectre w/schematic based assistants	X		
Export/import variables to CSV	X	X	
Measurements Across Sweeps and Corners	X	X	
Sweeps over Monte Carlo	X	X	
Run Preview with Run Point Selection	X	X	

Feature	ADE Explorer	ADE Assembler	95265 (VVO)
Extensive filtering capabilities in outputs setup, corners setup and results, dataview/setup assistant filtering	X	X	
Disk check & suspend/resume simulation	X	X	
On schematic waveform thumbnails/info balloons	X	X	
New storage view that is faster and easier to version with DM software	X	X	
Expression builder to simplify data analysis	X	X	
New simpler regression scripting (vs OCEAN)	X	X	
Support for Plotting Templates in ViVA	X	X	
Task Driven UI for easier statistical set-up	X	X	X
Feeds the Virtuoso ADE Verifier tool test set-ups		X	
Report Identical Histories (RIH) to aid in knowing what needs updating for incremental resimulation (supports Verifier well)		X	
Local Variable Sweep Support per Test		X	
Run Plans for mini-verification runs		X	

Feature	**ADE Explorer**	**ADE Assembler**	**95265 (VVO)**
Waveform specifications to autocheck wave responses		X	
Common library of test set-ups, bulk file/model updates (SPACE)		X****	
Access to Spectre Interactive Environment (SIE)		X	
Legato Reliability Solution			
Support for electro-thermal simulation	X***	X***	
Support for Spectre's new reliability simulation	X	X	
Support for fault simulation		X	
Statistical Mismatch Tuning	X + 95265	X + 95265	X
Launch Mathworks Matlab for data post-processing/script development	X**	X**	X**
Mathworks Matlab expression evaluation in results	X**	X**	X**
Notes			
X*	Not all features within the GUI are available to OASIS integrated (3rd party) simulators. Please work with Cadence CPG Marketing about individual situations.		

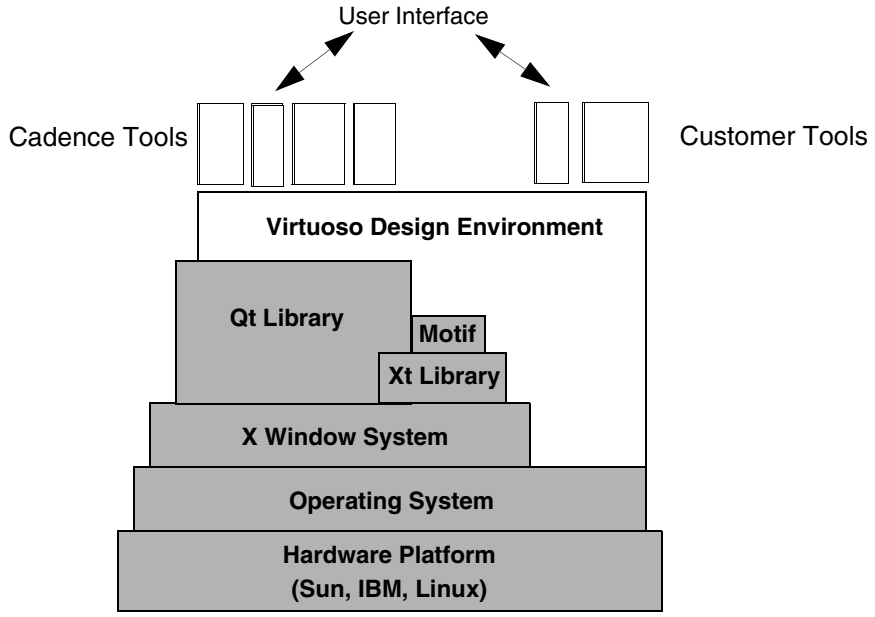
Feature	ADE Explorer	ADE Assembler	95265 (VVO)
X**	To enable the new Matlab capabilities, the customer needs both a Mathworks Matlab license plus the new Cadence product 95252 Virtuoso Matlab Integration Option		
X***	Post-618 FCS. Work with your account team to get the latest information.		
X****	Although the functionality is accessible in Assembler, it requires a Verifier (32570) license to access the feature.		

Hardware and Software Requirements

For hardware and software requirements needed to run the Virtuoso design environment software, see *Hardware and Software Requirements* in the *Virtuoso Design Environment User Guide*.

Software Layers

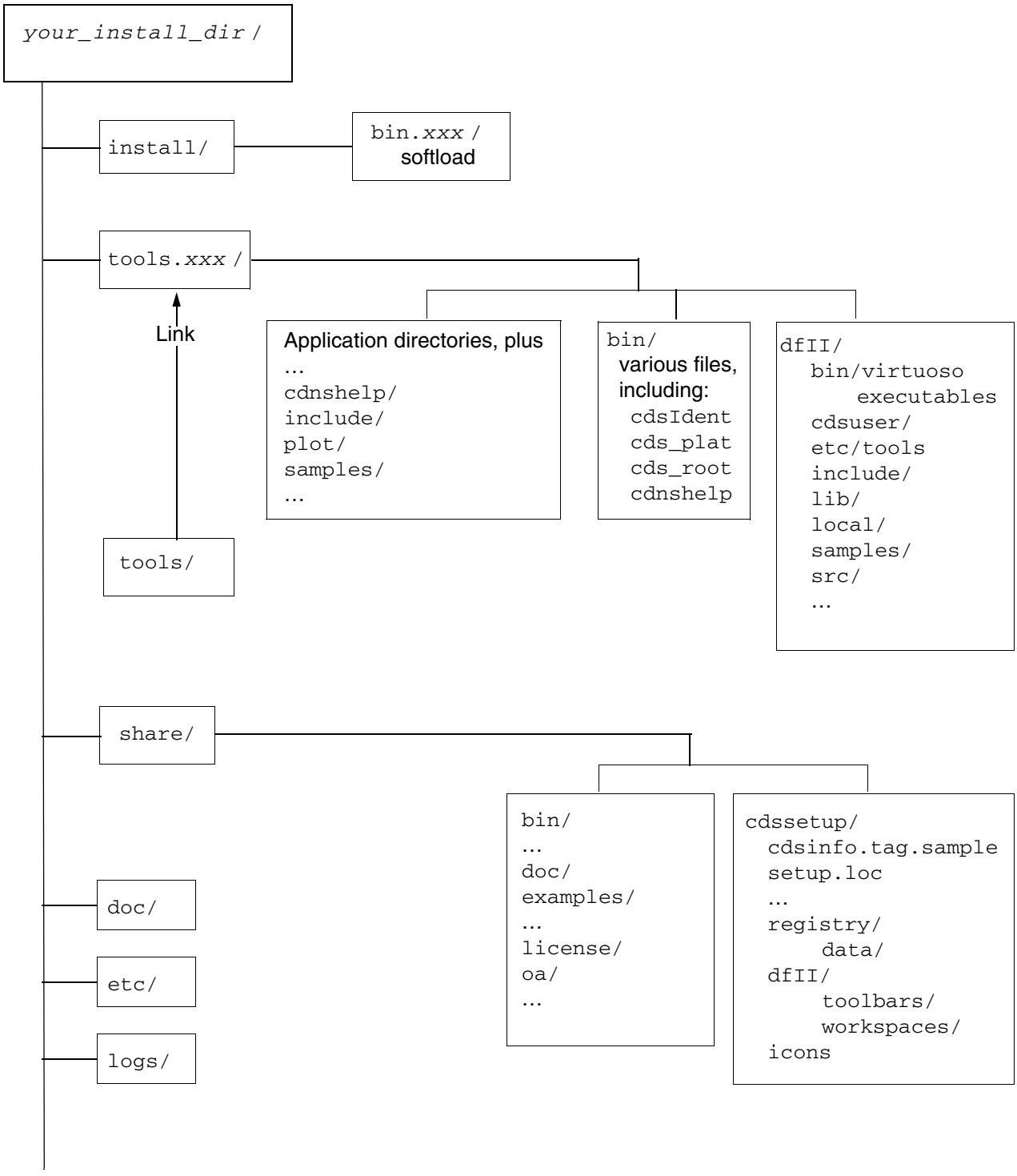
The figure below shows how the Virtuoso design environment software resides on top of several layers of software (bottom layer is your hardware platform).



For More Information	Refer to
Installation	*Cadence Installation Guide*
Licensing	*Cadence License Manager*
File locking	*Cadence Application Infrastructure User Guide*
Plotting	*Plotter Configuration User Guide*
Design flow	Your Cadence applications engineer (AE)

Virtuoso Design Environment Hierarchy

The software uses the following directory hierarchy to store executables and data files.



The *your_install_dir*/tools/dfII directory includes the directories listed below. The files in these directories depend on which Cadence® products are installed.

bin	Virtuoso design environment software executables for your platform.
bin.*xxx*	Link to bin directory.
cdsuser	Sample home directory environment setup files such as .Xdefaults, .cdsinit, .cshrc, .mwmrc, .simrc, .xinitrc, and arrowKeys.map.
etc/tools	Directories corresponding to particular applications or core pieces of infrastructure such as layout, schematic, and cdba. Each directory contains a default .cdsenv file and might also contain other sample files.
include	C header files to go with licensed Cadence products.
local	CAD group customization files and site-specific configuration files. **Note:** Installing new software does not overwrite this directory.
lib	Programming libraries.
samples	Sample user files. Some of the samples include the following: .cdsenv — Sample environment variable file local — Sample customization files plot — Schematic plotting files techfile — Technology files and libraries tutorials — Tutorials for various products
script	Configuration scripts.
src	More customization files.

Virtuoso Design Environment Executables

From the IC 6.1.4 release, all Virtuoso software executables have been integrated into one executable called `virtuoso`.

32- and 64-Bit Platforms

32- and 64-bit platform variants exist for the `virtuoso` workbench.

dbAccess Command-Line Executable

`dbAccess` is a non-graphical UNIX command-line executable for using the core SKILL functionality and accessing DFII OpenAccess design and technology file data. You can also use this executable to debug Pcells and verify that they run in other environments and have no dependencies on product packages or licenses. `dbAccess` supports `db`, `dd`, `rod`, and `tech` functions. Since it does not have a graphical user interface, it does not support `hi` and `ge` functions.

Command Switches

```
dbAccess [{-32 | -64 | -32only | -64only | -3264 | -6432}] [-quiet3264][-debug3264]
     [-plat <platform>] [-v3264] [-help3264] <[-load <file-name>]>
```

Description

`-32`	Select the 32-bit version; if the 32-bit version is not available, print a warning message and try to launch the 64-bit version.
`-64`	Select the 64-bit version; if the 64-bit version is not available, print a warning message and try to launch the 32-bit version.
`-32only`	Select the 32-bit version; if the 32-bit version is not available, print an error message and exit.
`-64only`	Select the 64-bit version; if the 64-bit version is not available, print an error message and exit.
`-3264`	Select the 32-bit version; if the 32-bit version is not available, print an info message and try to launch the 64-bit version.

`-6432`	Select the 64-bit version; if the 64-bit version is not available, print an info message and try to launch the 32-bit version.
`-quiet3264`	Suppress the warning/error/info messages of the -32/-32only/-3264 and -64/-64only/-6432 options. -debug3264 Print the environment updated by the wrapper and the command launched.
`-plat <platform>`	Override the default platform selection when the tool is launched from the `<install_root>/bin` directory.
`-v3264`	Print the wrapper's version string
`-help3264`	Display detailed help of `dbAccess` generic options.
`-load`	Loads and executes the specified SKILL file.

Note: If you do not provide a command-line option, `dbAccess` switches to interactive mode and accepts commands from the keyboard or `stdin`.

In addition, `dbAccess` accepts the Virtuoso command-line options, such as:

`-V`	Displays Cadence release version.
`-W`	Displays Cadence release subversion.

However, it ignores options that require a graphical user interface.

Example

Suppose you have a SKILL file, `listLibs.il`, which includes commands to display the libraries specified in the `cds.lib` file. If a `cds.lib` file is not present, the default `cds.lib` from the installation directory is used.

The file contents are as follows:

```
foreach(lib ddGetLibList() printf("%s\n" lib~>name))
```

Now, if you run the `dbAccess` command and loads this SKILL file, the following output is displayed on a terminal window:

```
dbAccess -load listLibs.il
Virtuoso Framework License (111) was checked out successfully. Total checkout
time was 0.64s.
cdsDefTechLib
basic
```

```
US_8ths
rfLib
rfExamples
ahdlLib
analogLib
functional
```

Note: The above output is generated using the default `cds.lib` from the installation directory.

Virtuoso Design Environment Licensing Setup

The key licensing characteristics of the Virtuoso design environment are outlined in the following table.

Licensing Characteristic	Virtuoso Design Environment
Licensing software	SoftShare
Licensing DAEMON	`cdslmd`
License file	`license.`*`HOSTID`*
Path to DFII	*`your_install_dir`*`/tools/dfII`
Path to the license directory	*`your_install_dir`*`/share/license`
Variable for license file	See the *Cadence License Manager*
File locking	CLS

Note: Licenses from several license servers can be used if there are multiple independent license servers configured.

Important

All Cadence software requires Cadence licenses. To specify your license file, set the `CDS_LIC_FILE`. If a license cannot be found on any of the servers specified in the `CDS_LIC_FILE`, the license search will continue to look through the `LM_LICENSE_FILE`. If you do not want a license search to continue beyond the `CDS_LIC_FILE`, you need to set the `CDS_LIC_ONLY` environment variable which will instruct Virtuoso to ignore the `LM_LICENSE_FILE`.

For more information about license files and variables, see the *Cadence Installation Guide* and *Cadence License Manager*.

The following table answers frequently asked licensing questions:

Question	Answer
In interactive mode, when are licenses checked in?	Licenses are checked in automatically when you exit an application.
Do all Cadence products support global `TIMEOUT`?	Yes. `TIMEOUT` sets the time after which an inactive license is reclaimed by the vendor daemon. You can amend the default `TIMEOUT` setting in the options file to specify how long it is before a *product* license times out. The value entered will be valid as long as it exceeds the minimum setting (3600 secs). **Note:** `TIMEOUTALL`, which applies a timeout to all features is also supported in DFII products. For example, `TIMEOUT` *featurename seconds* `TIMEOUTALL` *seconds*
What happens to my work and any open windows when products lose their licenses?	When you execute the next command, it will re-check out the license automatically. **Note:** You will only notice an issue if the license you need (the license that timed out) is no longer available.
After startup, how often does a product check for a valid license?	The license-checking interval depends on the product. In general, interactive products (for editing) check at the start of every editing command and batch-mode products check every five minutes.
Can I wait for a license?	No.

Question	Answer
If prolonged batch jobs require multiple licenses, are all licenses reserved or checked out up front?	This depends on the application.

Preface

The *Virtuoso Software Licensing and Configuration User Guide* describes how to configure the Virtuoso® design environment software as well as provides the following information about the environment:

- Executables needed to run the software - see "Launching the Virtuoso Product Family Applications" on page 21 and "Setting the Installation Path" on page 88 for more information
- Operating system requirements - see "Product Tier Features" on page 67 and "Verifying Your System Configuration" on page 102 for more information
- Directory hierarchy for storing executables and data files - see "Virtuoso Design Environment Hierarchy" on page 76 and "Setting the Installation Path" on page 88 for more information
- Key licensing characteristics - see "Virtuoso Design Environment Licensing Setup" on page 81 for more information

This user guide is aimed at developers and designers of integrated circuits and assumes that you are familiar with:

- The Virtuoso design environment and application infrastructure mechanisms designed to support consistent operations between all Cadence® tools.
- The applications used to design and develop integrated circuits in the Virtuoso design environment, notably, the Virtuoso Layout Suite, and Virtuoso Schematic Editor.
- The Virtuoso design environment technology file.

This preface contains the following topics:

- Scope
- Licensing Requirements
- Related Documentation
- Additional Learning Resources
- Customer Support
- Feedback about Documentation

- Typographic and Syntax Conventions

Scope

Unless otherwise noted, the functionality described in this guide can be used in both mature node (for example, IC6.1.8) and advanced node (for example, ICADVM20.1) releases

Label	Meaning
(ICADVM20.1 Only)	Features supported only in the ICADVM20.1 advanced nodes and advanced methodologies release.
(ICADVM20.1 EAD Only)	Features supported only in the ICADVM20.1 release and which require the Virtuoso_Layout_Suite_EAD license (95600).
(IC6.1.8 Only)	Features supported only in mature node releases.
(ICADVM20.1 Virtuoso RF Solution Only)	Features supported only in the ICADVM20.1 release releases and which require the Virtuoso_RF_Option (95560) license.
(ICADVM20.1 Virtuoso MultiTech Framework Only)	Features supported only in the ICADVM20.1 release and which require the Virtuoso_MultiTech_Framework (95022) license.
(ICADVM20.1 Photonics Only)	Features supported only in the ICADVM20.1 release and which require the Virtuoso_Photonics_Option license (95550).

Licensing Requirements

In the ICADVM20.1 release, product license 111 and 95011 (`Virtuoso_Adv_Node_Framework`) must be checked out at Virtuoso startup. For more information on the licensing changes in the ICADVM20.1 release, see License Usage When Using Interactive Commands on page 23.

Related Documentation

What's New and KPNS

- *Virtuoso Software Licensing and Configuration What's New*
- *Virtuoso Software Licensing and Configuration Known Problems and Solutions*

Installation, Environment, and Infrastructure

- *Cadence Installation Guide*
- *Cadence License Manager*.
- *Virtuoso Design Environment User Guide*
- *Virtuoso Design Environment SKILL Reference*
- *Cadence Application Infrastructure User Guide*

Technology Information

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

Virtuoso Tools

IC6.1.8 Only

- *Virtuoso Layout Suite L User Guide*
- *Virtuoso Layout Suite XL User Guide*
- *Virtuoso Layout Suite GXL Reference*

ICADVM20.1 Only

- *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 Concurrent Layout Editing User Guide*
- *Virtuoso Design Planner User Guide*
- *Virtuoso Multi-Patterning Technology User Guide*
- *Virtuoso Placer User Guide*
- *Virtuoso Simulation Driven Interactive Routing User Guide*
- *Virtuoso Width Spacing Patterns User Guide*
- *Virtuoso RF Flow Guide*
- *Virtuoso Electromagnetic Solver Assistant User Guide*

IC6.1.8 and ICADVM20.1

- *Virtuoso Abstract Generator User Guide*
- *Virtuoso Custom Digital Placer User Guide*
- *Virtuoso Design Rule Driven Editing User Guide*
- *Virtuoso Electrically Aware Design Flow Guide*
- *Virtuoso Floorplanner User Guide*
- *Virtuoso Fluid Guard Ring User Guide*
- *Virtuoso Interactive and Assisted Routing User Guide*
- *Virtuoso Layout Suite SKILL Reference*
- *Virtuoso Module Generator User Guide*
- *Virtuoso Parameterized Cell Reference*
- *Virtuoso Pegasus Interactive User Guide*
- *Virtuoso Space-based Router User Guide*

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 the related features 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.

Setting Up the Virtuoso Software

This chapter contains the following startup information:

- Quick Start: Linux and Unix Environments on page 86
- Configuration and Startup Procedures on page 88

Each time the application software runs, the application loads variables in this order:

1. Resources in the resource database

   The resources are taken directly from the `.Xdefaults` file only if the `res.db` file does not exist. It is created automatically by common desktop environment (CDE) after login and incorporates your `.Xdefaults` file at that time.

2. Variables in the `app_defaults` file
3. Variables set by the application
4. Variables set by the *`your_install_dir`*`/tools/`*`app_dir`*`/local/.cdsenv` file
   - *`your_install_dir`* is the directory that contains the software
   - *`app_dir`* is the directory containing the Cadence® application files
5. Environment variables
6. Program loads all `.cdsenv` files in the following order if the `CDS_LOAD_ENV` environment variable is not set:
   - *`your_install_dir`*`/tools/dfII/etc/tools/`*`application`*`/.cdsenv`
   - *`your_install_dir`*`/tools/dfII/local/.cdsenv`
   - `$HOME/.cdsenv`
7. Program searches the following locations in order and loads variables from the first `.cdsinit` file encountered:
   - *`your_install_dir`*`/tools/dfII/local`
   - current working directory: .
   - your home directory

Individual user settings override the variables set by the application and any group customization. You can copy the `.cdsenv` site file to your local directory and customize it or choose *Options – Save Defaults* in the Command Interpreter Window (CIW) to save variable settings to a `.cdsenv` file (see "Saving New Default Values" in the *Virtuoso Design Environment User Guide* for more information).

Quick Start: Linux and Unix Environments

This section outlines the quick start procedure for Linux and Unix environments running X Windows graphics software.

Important

You may need to use a `root` login to modify files or change permissions.

1. Set up a user account.

   Note: You can refer to "Setting Up a User Account" on page 90 for more details.

2. Modify and apply your search path as follows:

If you are running the csh shell:

Modify your search path in your `.cshrc` as follows:

```
set path = (. your_install_dir/tools/bin \
    your_install_dir/tools/dfII/bin $path)
```

Then, either log out and back in again or type the following commands:

```
source .cshrc
rehash
```

If you are running bash or ksh shell:

Modify your search path in your `.bashrc` or `.kshrc`, respectively.

```
PATH=your_install_dir/tools/bin:your_install_dir/tools/dfII/bin:$PATH;
export PATH
```

Then, either log out and back in again or type one of the following commands:

```
source .bashrc
```

Or

```
source .kshrc
```

If you are running sh shell:

Modify your search path in your `profile` file.

```
PATH=your_install_dir/tools/bin:your_install_dir/tools/dfII/bin:$PATH;
export PATH
```

Then, either log out and back in again or type the following command:

```
.<profile file>
```

Note: If an incorrect path message appears, you need to correct any errors in your path statement (`.cshrc`, `bashrc`, `.kshrc`, or `.profile`) and re-apply the changes.

```
no programName in path path
```

3. Start the software by typing

   ```
   executable &
   ```

   where *executable* is the executable you want to run (for example, `virtuoso`).

   The CIW appears after a short initialization period.

Note: You must set up plotters before you can plot a design (see the Plotter Configuration User Guide for details).

Configuration and Startup Procedures

The tasks below provide a guide to the full configuration and startup procedure for your software. This document assumes you have copies of the default initialization files from *your_install_dir*/tools/dfII/cdsuser/.* in a local directory and that you are logging in through dtlogin (perhaps from /usr/dt/bin).

- Setting the Installation Path on page 88
- Setting Up a User Account on page 90
- Setting the Focus Correctly on page 90
- Specifying Cadence Environment Variables on page 92
- Modifying the .rhosts File on page 97
- Modifying the .Xdefaults or Equivalent File on page 98
- Modifying the .cdsinit File on page 101
- Verifying Your System Configuration on page 102
- Distributing the User Files on page 103
- Configuring Remote Displays on page 103
- Library and Tool Issues on page 105

Setting the Installation Path

Before starting the software, set a path to the executables and components.

For example,

```
setenv CDSHOME /yourInstallDirPath
set path = ( $CDSHOME/tools/bin $CDSHOME/tools/dfII/bin )
```

Note: The $CDS_INST_DIR environment variable is no longer used by Cadence applications. For more information, see “$CDS_INST_DIR” under “Search Mechanism” in Chapter 3 of the *Cadence Application Infrastructure User Guide*.

Setting an environment variable (such as CDSHOME) allows consistency when setting paths to additional components or libraries. However, you do not need to set library hierarchy directories using the LD_LIBRARY_PATH environment variable or to set library paths using an environment variable in your library definitions file. See the *Virtuoso Design Environment User Guide* for more information.

The `LD_LIBRARY_PATH` UNIX environment variable is used for special library modules (`.so`) that tools might need. It is not necessary to add hierarchy library directories to the `LD_LIBRARY_PATH` because the executable scripts add the appropriate information.

The library definitions file (an ASCII text file that defines which libraries are accessible and where they are located in the installation hierarchy) uses root expressions to refer to installation paths. For information about how to refer to the root of Cadence installation hierarchies in your library definitions file without using environment variables, see "Installation Root Expressions" in Chapter 5 of the *Cadence Application Infrastructure User Guide*.

Cadence tools are typically installed in their own directory hierarchy. You should use different UNIX environment variables to represent each release hierarchy. The following example shows how to set installation paths in a `.cshrc` file (you must modify these paths to match your installation).

Note: For information regarding setting environment variables for specific tools, refer to the documentation for that particular tool.

```
#*********************************************************************#
# Set path to DFII install directory
set CDSHOME = /cdsIntall/cds/IC541
#
# Set path to ICC install directory
setenv ICCHOME /cdsInstall/cds/iccraft112
#
# Set path to Assura install directory
setenv ASSURAHOME /cdsInstall/cds/Assura312
#
# Set path to Innovus install directory
setenv INVSHOME /dsmInstall/INVS191
#
#*********************************************************************#
#
# Set paths to Cadence executables
#(In this example, variable names are assigned for each tool path,
# then set in the set path statement)
#
set dfiiPath = ( $CDSHOME/tools/bin $CDSHOME/tools/dfII/bin )
set iccPath = ( $ICCHOME/tools/iccraft/bin )
set assuraPath = ( $ASSURAHOME/tools/bin $ASSURAHOME/tools/dfII/bin )
set invsPath = ( $INVSHOME/tools/bin $INVSHOME/tools/fe/bin
)
#
set path=( ~/bin \
/usr/bin /usr/sbin /bin /etc /usr/etc \
$dfiiPath \
$iccPath \
$assuraPath \
$invsPath \
    )
#*********************************************************************#
# Set paths to license files
setenv LM_LICENSE_FILE $CDSHOME/cdsbin/lmtools/license.dat/license.dat
#*********************************************************************#
```

```
# END of the .cshrc file
#*****************************************************************#
```

Setting Up a User Account

In order to set up accounts for new users, you must already have created a login and home directory for each user. The *your_install_dir*/tools/dfII/cdsuser directory contains sample home directory environment setup files (see "Virtuoso Design Environment" on page 16).

Next, follow the steps outlined in "Copying Sample Files to Your Home Directory" on page 90.

Once you have set up a user account, you can modify your search path and run the software as described in "Quick Start: Linux and Unix Environments" on page 86.

Copying Sample Files to Your Home Directory

1. Copy the sample files to your home directory.

   You can use commands similar to the following:

   ```
   cd
   cp your_install_dir/tools/dfII/samples/.??* .
   cp your_install_dir/tools/dfII/cdsuser/.??* .
   ```

Caution

The copy command (cp) will overwrite any existing files of the same name in your directory.

2. Go through each section of the sample files and customize them for your site. Be sure to replace /cdsdir with *your_install_dir*.

Setting the Focus Correctly

On KDE, Red Hat Enterprise Linux, and Gnome, a newly opened window might pop up under a window or on the desktop and is not immediately visible. You can correct this as follows.

In the KDE window manager:

1. Open KDE Control Center or type /usr/bin/kcontrol on LINUX command prompt to start the KDE Control Center.
2. Select *Desktop — Window Behavior*.

3. Select the *Advanced* tab.
4. Set *Focus stealing prevention level* to `None` or `Low`.

On RHEL6.X:

1. Open the *System Settings* form by running the following command:

   ```
   LINUX> /usr/bin/systemsettings
   ```

2. Click the *General* tab and then *Window Behavior*.
3. Set *Focus stealing prevention level* to `None` or `Low`.

On Gnome:

1. Open the Gnome Configuration Editor window by entering the `gconf-editor` command at the terminal.
2. Navigate to *apps — metacity — general*.
3. Select `new_windows_always_on_top`.
4. Set the *`focus_new_window`* variable to `strict`.

Specifying Cadence Environment Variables

You can specify any number of the following environment variables.

Cadence Environment Variables

Variable	What You Can Specify
CDS_DEFAULT_BROWSER	Sets the initial value of the `ui .cdsenv` variable `webBrowser` to this value. If set, this is overridden by the value of `ui.webBrowser` in the `.cdsenv` file.
CDS_NOGRAPH_DISPLAY	Defines the display to be used for `nograph` mode instead of the `nograph cdsXvnc` server.
CDS_XVNC_OFFSET	Sets the least significant digit of the display number to try first for the `-nograph` display. The default is `0`. For example, if set to `5` and `TENBASE` is not set, Virtuoso will first try to connect to `display:95`. This will be overridden by the command line argument `-nographN` if used. Defining `CDS_NOGRAPH_DISPLAY` will also override this variable setting.
CDS_XVNC_TENBASE	Sets the tens digit of the display number to use for the `nograph` display. Valid values are `1` to `9`. For example, if set to `5` Virtuoso will attempt to use a `nograph` display in the range `50` through `59` (starting display depends on the offset value). The special value `+` can also be used to use the display range `400` through `409`, if displays in the range `10` through `99` are not available. This will be overridden by the command line argument `-vncTenbase`, if used. Defining `CDS_NOGRAPH_DISPLAY` will also override this variable setting.
CDS_LOG_PATH	A colon-separated path of directories (ordered by preference) in which to put the log file. If none of these directories exist or they are all not writable, the default `$HOME` directory is used.

Cadence Environment Variables, *continued*

Variable	What You Can Specify
CDS_LOAD_ENV	A customized search order for .cdsenv using one of the following keywords: ■ False loads neither ~/.cdsenv or CWD/.cdsenv ■ CWD loads only CWD/.cdsenv ■ addCWD loads ~/.cdsenv and then loads CWD/.cdsenv ■ CWDElseHome loads CWD/.cdsenv if it exists, or ~/.cdsenv, if it exists ■ CSF uses the Cadence Setup Search File mechanism to find the .cdsenv files to load For more information, see "Specifying a Search Order for .cdsenv" in the *Virtuoso Design Environment User Guide.*
CDS_LOG_TIMESTAMPS	Set to 1, True, Yes, t, or y for default timestamp logging This is same as the command line option -logtime. Set to 2 or rel or relative for relative timestamp logging. This is same as the command line option -logtimerel. **Note:** The options are case insensitive.
CDS_USE_LOCAL_TIMESTAMP	If local timestamp is defined and the value does not start with 0, f, or n, all timestamps in the CDS.log file will use local time instead of Universal Time (UTC) (also known as GMT). **Note:** The options are case insensitive.
CDS_FILTER_CDSENV_WARNINGS	If defined and the value starts with t or y, warnings for unregistered and type mismatched variables will not be thrown while loading values from the .cdsenv file.

Cadence Environment Variables, *continued*

Variable	What You Can Specify
`CDS_PROMPT_CKOUT`	Whether the Auto Checkout form automatically appears when you open data files using a product that has a graphical user interface (GUI) and automatic check-out capability using one of the following keywords: `all` prompts for all *autocheckout* requests and is the default. Causes all tools that have a GUI and perform *autocheckout* operations to prompt you to check out data before performing an *autocheckout*. A prompt is issued regardless of the values of the other variables controlling check in/out behavior. This action lets you set up an environment that prompts for checkout wherever possible. Tools that cannot prompt cannot automatically check out data unless you also set the `CDS_AUTO_CKOUT` variable. ■ `none` does not prompt for any *autocheckout* requests. ■ `views` prompts for *autocheckout* requests of cellview data. ■ `files` prompts for *autocheckout* requests of non-cellview data.

Cadence Environment Variables, *continued*

Variable	What You Can Specify
`CDS_AUTO_CKOUT`	Whether the software automatically checks out a file when you open it with a product using one of the following keywords: ■ `all` checks out all *autocheckout* requests and is the default. ■ `none` does not check out any *autocheckout* requests. ■ `views` checks out *autocheckout* requests of cellview data. ■ `files` checks out *autocheckout* requests of non-cellview data. If a tool has a GUI and `CDS_PROMPT_CKOUT` is set to `all`, `CDS_AUTO_CKOUT` has no effect. Otherwise, when set to `all`, it causes all tools that have an *autocheckout* capability to automatically check out a file as needed.

Cadence Environment Variables, *continued*

Variable	What You Can Specify
CDS_PROMPT_CKIN	Whether the Auto Checkin form appears when you close properties or files that were automatically checked out. Or if you try to exit a session without closing properties or files that were automatically checked out, while using a Cadence product that has both a GUI and automatic check-in capability using one of the following keywords: ■ files prompts you before performing an *autocheckin* of non-cellview data (default). It overrides CDS_AUTO_CKIN and applies to interactive tools only. all prompts for all *autocheckin* requests. Causes all tools that have a GUI and perform check in operations to prompt you whether or not to check in data before performing an *autocheckin*. Tools that do not have a GUI are not affected by this variable. ■ none does not prompt for any *autocheckin* requests. ■ views prompts for *autocheckin* requests of cellview data.
CDS_AUTO_CKIN	Whether the software automatically checks in data files when you close properties or files that were automatically checked out or when you exit a session without closing properties or files that were automatically checked out using one of the following keywords: ■ files file is *autocheckin* if it was automatically checked out during this session and is the default. ■ all checks in all *autocheckin* requests. ■ none does not check in any *autocheckin* requests. ■ views checks in *autocheckin* requests of cellview data. If the tool has a GUI and CDS_PROMPT_CKIN is set to all, CDS_AUTO_CKIN has no effect. When this variable is set to all, all tools automatically check in any file they automatically checked out.

Cadence Environment Variables, *continued*

Variable	What You Can Specify
CDS_IBM_FULLDUMP	Whether to dump the core with full traceback and data sections (IBM only). When an IBM workstation dumps its core, the core contains a full traceback section but no data section. Debugging typically requires the data section.
CDS_LICENSE_DIR	Alternate SoftShare license directory used.
CDS_LOG_VERSION	Naming convention for preserving multiple log files using one of the following keywords: ■ sequential adds a sequential number to the name of the log file, such as CDS.log.1 or CDS.log.2. ■ pid adds the number of the UNIX process to the name of the log file, such as CDS.log.1719 or CDS.log.2250. **Note:** If you do not specify CDS_LOG_VERSION, each session overwrites the log file from the previous one.
CDS_SHM_ADDR	The address to use for shared memory. **Note:** Set this variable only when you do not have shared memory. Setting this variable turns off VO fasttime. If you set CDS_SHM_ADDR to a non-zero hex number, the software uses that address instead of the default address for shared memory.
CDS_AUTO_64BIT	Which applications to run using the 64-bit version (see "Running 64-Bit Versions of Applications" on page 111).

Modifying the .rhosts File

Note: Modifying the .rhosts file is optional.

If you are going to run a remote simulation or remote shmd, your network must be properly configured. You must create a .rhosts file in your home directory or edit the rhosts.equiv file in the /etc directory of each workstation on the network for which you want to run a remote simulation or shmd process. If you do not create a .rhosts file, any

attempt to run a simulation or `shmd` on the remote workstation gives you a “permission denied” error.

Modifying the .Xdefaults or Equivalent File

The *your_install_dir*`/tools/dfII/cdsuser` directory contains a `.Xdefaults` file. The X Window System normally uses the resources in the X resources database that is loaded upon login from your local `.Xdefaults` file.

You can copy the Cadence-provided `.Xdefaults` file to your local directory and modify it to contain the settings you want.

1. Copy Cadence `.Xdefaults` file to your local directory:

   ```
   cd
   cp your_install_dir/tools/dfII/cdsuser/.Xdefaults .
   ```

2. Modify the settings you want and save the file.

The software reads in X resources when you start your application. If you make changes to your `.Xdefaults` file after the software is running, these changes do not take effect until you read the resources into the X server and restart your application. You must force the server and window manager to see the file changes.

1. Read in the `.Xdefaults` file.

   ```
   xrdb -merge .Xdefaults
   ```

   Note: The `-merge` option prevents deleting desired settings that are not in the `.Xdefaults` file.

2. Log out and login again.

Cadence Resources

The following table identifies some of the resources that you can modify in your local `.Xdefaults` file. You can find examples of these resource settings in *your_install_dir*`/tools/dfII/cdsuser/.Xdefaults`.

Cadence resources use tight bindings with a period (`.`) instead of the usual loose bindings with an asterisk (`*`). The period cannot be replaced by an asterisk. Wildcard resources produce unpredictable effects on dialog boxes and menus.

Cadence recommends that you restrict your resources to the list shown in the resource list above to prevent X resources from clashing with SKILL descriptions for forms and menus.

Cadence Resource	Description
`Opus.activeBannerColor`	Background color of the window number label displayed at the bottom-left corner of the current window.
`Opus.beepDialog`	Beeps when warning or dialog boxes appear.
`Opus.flashColor`	Color of flashing box drawn around error fields in forms.
`Opus.formPlacement`	Location of form (`top`, `bottom`, `left`, `right`, or `center`).
`Opus.formRelativeTo`	Location of form relative to screen, window (currentWindow), or CIW.
`Opus.geometry`	Location and size of CIW (overrides `Opus.x`, `Opus.y`, `Opus.height`, and `opus.width`).
`Opus.height`	Height of CIW (`Opus.geometry` overrides this setting).
`Opus.optionFormPlacement`	Placement of option form (*top*, *bottom*, l*eft*, *right*, or *center*).
`Opus.optionFormRelativeTo`	Placement of option form relative to *screen*, current window (currentWindow), or CIW.
`Opus.selectionColor`	Controls the highlight that appears around selected objects when `Opus.selectionPlane` is `True`, otherwise it uses the grid color.
`Opus.showScrollBars`	Sets the initial value of the `ui` cdsenv variable `showScrollBars`. If set, this will be overridden by the value of `ui.showScrollBars` in the `.cdsenv` file.
`Opus.textEditor`	Text editor.
`Opus.warpPointer`	Automatically moves cursor to *OK* in dialog boxes and to current value on pop-up menus.
`Opus.width`	Width of the CIW (`Opus.geometry` overrides this setting).

Cadence Resource	Description
`Opus.x`	Horizontal position (X axis) of the CIW (`Opus.geometry` overrides this setting).
`Opus.y`	Vertical position (Y axis) of the CIW (`Opus.geometry` overrides this setting).

Desktop and Window Manager Resources

Follow these steps for setting the *Focus stealing prevention level*:

1. Open KDE Control Center. Or type `/usr/bin/kcontrol` on LINUX command prompt to invoke KDE Control Center.
2. Select *Desktop — Window Behavior*. Select the *Advanced* tab.
3. Set *Focus stealing prevention level* to `None` or `Low`.

This ensures that new dialog box or windows or forms appear in the front and not in background under some window.

On RHEL6.X the same can be implemented by invoking `systemsettings` command:

```
LINUX> /usr/bin/systemsettings
```

1. On the *System Settings* form, *General* tab, click *Window Behavior*.
2. Set *Focus stealing prevention level* to `None` or `Low`.

On KDE running on RHEL6.X and later, follow these steps to prevent double-click actions to be invoked with a single click:

1. From the main menu, select *Settings — System Settings* or run the `systemsettings` command.
2. Click *Keyboard & Mouse* and then, select *Mouse*.
3. Select the *Double-click to open files and folders* option.

   Note: The default value of *Single-click to open files and folders* causes double-click actions to be invoked with a single click.

For information about setting the font options, see *Viewing the Font List* in the *Virtuoso Design Environment User Guide*.

Modifying the .cdsinit File

Before you set up users, decide whether you want them to use the same `.cdsinit` file. The program uses the first `.cdsinit` file that it finds in the following order:

1. *your_install_dir*`/tools/dfII/local/.cdsinit`
2. Current directory (`.`)
3. Your home directory (`~`)

You can customize the `.cdsinit` file after you copy it into the current directory or the user home directory. Use the following guidelines when modifying the `.cdsinit` file:

Tip

You can find a sample `.cdsinit` file at *your_install_dir*`/tools/dfII/samples/local/cdsinit`.

- Group the information in the file according to the application and clearly label the application name using comment lines.
- For each setting, use comment lines to specify what the normal defaults are and to what they might be changed. Specify what you normally use.
- Warn about any settings that lead to undesirable results.
- Set the option to your normal value or to the normal default value.
- Read several files if necessary, such as
  - *your_install_dir*`/tools/dfII/local/.cdsinit`
  - A project's `./.cdsinit`
  - Each user's personal `~/.cdsinit`

To create a site-specific file, follow these steps:

1. Copy the default file to another directory.

   Type the following to copy the default file to a local directory:

   ```
   cp your_install_dir/tools/dfII/cdsuser/.cdsinit
   your_install_dir/tools/dfII/local
   ```

2. Edit the last line of the (system) `.cdsinit` file to read.

   ```
   if(isFile("./.cdsinit")
   then load("./.cdsinit")
   else when(isFile("~/.cdsinit")
   ```

```
            load("~/.cdsinit)
        )
)
```

3. Save and close the file.

This allows you to customize their own `.cdsinit` files. Some customizations may include the following:

- You can specify the library search path for the site-specific libraries. See the *Virtuoso Design Environment User Guide* for more information.
- You can define key bindings.
- You can set up the environment for SKILL programming (see the *Cadence SKILL Language Reference* and the *Cadence SKILL Language User Guide*).
  - Set the log filter to display user and program results in the output area of the CIW:

    ```
    hiSetFilterOptions( t t t t t t t )
    ```

  - Turn the `writeProtect` switch off:

    ```
    sstatus( writeProtect nil )
    ```

    This switch affects only procedures loaded after the software starts running.

  - Turn on the debugger:

    ```
    installDebugger()
    ```

- If you have your own SKILL files, you can include one or more locations in the SKILL search path by adding a `setSkillPath` function with a list of space-separated path strings such as one of the following:

  ```
  setSkillPath(". skill_path")
  setSkillPath(strcat(". your_install_dir/etc skill_path"))
  setSkillPath(strcat(". " prependInstallPath("/etc ") "skill_path"))
  ```

Verifying Your System Configuration

The system configuration checker (*your_install_dir*`/tools/bin/checkSysConf`) verifies that the operating system level, system configuration, and patch level of your machines match Cadence software requirements. To run the system configuration checker, do the following:

1. In an xterm window, type the following command to get a list of valid release names:

   ```
   checkSysConf -r
   ```

   Valid release names appear on your screen.

2. Run the system configuration checker using a valid release name as follows:

```
checkSysConf validReleaseName
```

For example,

```
checkSysConf IC5.2 | more
```

The system configuration checker reports system information (such as *Host Name* and *Hostid*) and verifies whether system requirements are met (such as *MEMORY*, *SWAP*, *DISPLAY*, *PACKAGE*, and *PATCH* requirements). Any failures to comply with system requirements for the specified release stream appear at the end of the report. For example, you might need to find out which products require a missing patch by running `checkSysConf` with the `-p` option:

```
checkSysConf IC5.2 -p patch
```

For more information about the system configuration checker, refer to the *Cadence Online Support* web site: http://support.cadence.com/docs/files/releases/sys_conf_check/welcome.html.

Distributing the User Files

Do not overwrite existing user files.

➤ Copy the files to the new user home directory.

Include the following default or site-specific files:

`.cshrc` or `.profile`
`.Xdefaults`

Configuring Remote Displays

Normally, only specified workstations can access the X server, but you can change the access by typing one of the following commands:

Important

Contact your system administrator before using this command as it overrides more stringent security schemes.

Command	Description
`xhost +`	All hosts can access the X server.
`xhost +` *`hostname`*	Add *`hostname`* to access list.
`xhost -` *`hostname`*	Remove *`hostname`* from access list.
`xhost -`	Only those workstations listed in the access list, `/etc/X*.hosts`, can access the X server.

Library and Tool Issues

You might want to also consider the following library and tool issues:

- Converting existing libraries (see the *Conversion Tool Box User Guide)*
- Creating new libraries
- Integrating other tools
- Creating menus for tools (see the *Virtuoso Design Environment User Guide*)
- Locale Settings

Virtuoso Design Environment Executables

From the IC 6.1.4 release, all Virtuoso software executables have been integrated into one executable called `virtuoso`.

32- and 64-Bit Platforms

32- and 64-bit platform variants exist for the `virtuoso` workbench.

dbAccess Command-Line Executable

`dbAccess` is a non-graphical UNIX command-line executable for using the core SKILL functionality and accessing DFII OpenAccess design and technology file data. You can also use this executable to debug Pcells and verify that they run in other environments and have no dependencies on product packages or licenses. `dbAccess` supports `db`, `dd`, `rod`, and `tech` functions. Since it does not have a graphical user interface, it does not support `hi` and `ge` functions.

Command Switches

```
dbAccess [{-32 | -64 | -32only | -64only | -3264 | -6432}] [-quiet3264][-debug3264]
     [-plat <platform>] [-v3264] [-help3264] <[-load <file-name>]>
```

Description

`-32`	Select the 32-bit version; if the 32-bit version is not available, print a warning message and try to launch the 64-bit version.
`-64`	Select the 64-bit version; if the 64-bit version is not available, print a warning message and try to launch the 32-bit version.
`-32only`	Select the 32-bit version; if the 32-bit version is not available, print an error message and exit.
`-64only`	Select the 64-bit version; if the 64-bit version is not available, print an error message and exit.
`-3264`	Select the 32-bit version; if the 32-bit version is not available, print an info message and try to launch the 64-bit version.

`-6432`	Select the 64-bit version; if the 64-bit version is not available, print an info message and try to launch the 32-bit version.
`-quiet3264`	Suppress the warning/error/info messages of the -32/-32only/-3264 and -64/-64only/-6432 options. -debug3264 Print the environment updated by the wrapper and the command launched.
`-plat <platform>`	Override the default platform selection when the tool is launched from the `<install_root>/bin` directory.
`-v3264`	Print the wrapper's version string
`-help3264`	Display detailed help of `dbAccess` generic options.
`-load`	Loads and executes the specified SKILL file.

Note: If you do not provide a command-line option, `dbAccess` switches to interactive mode and accepts commands from the keyboard or `stdin`.

In addition, `dbAccess` accepts the Virtuoso command-line options, such as:

`-V`	Displays Cadence release version.
`-W`	Displays Cadence release subversion.

However, it ignores options that require a graphical user interface.

Example

Suppose you have a SKILL file, `listLibs.il`, which includes commands to display the libraries specified in the `cds.lib` file. If a `cds.lib` file is not present, the default `cds.lib` from the installation directory is used.

The file contents are as follows:

```
foreach(lib ddGetLibList() printf("%s\n" lib~>name))
```

Now, if you run the `dbAccess` command and loads this SKILL file, the following output is displayed on a terminal window:

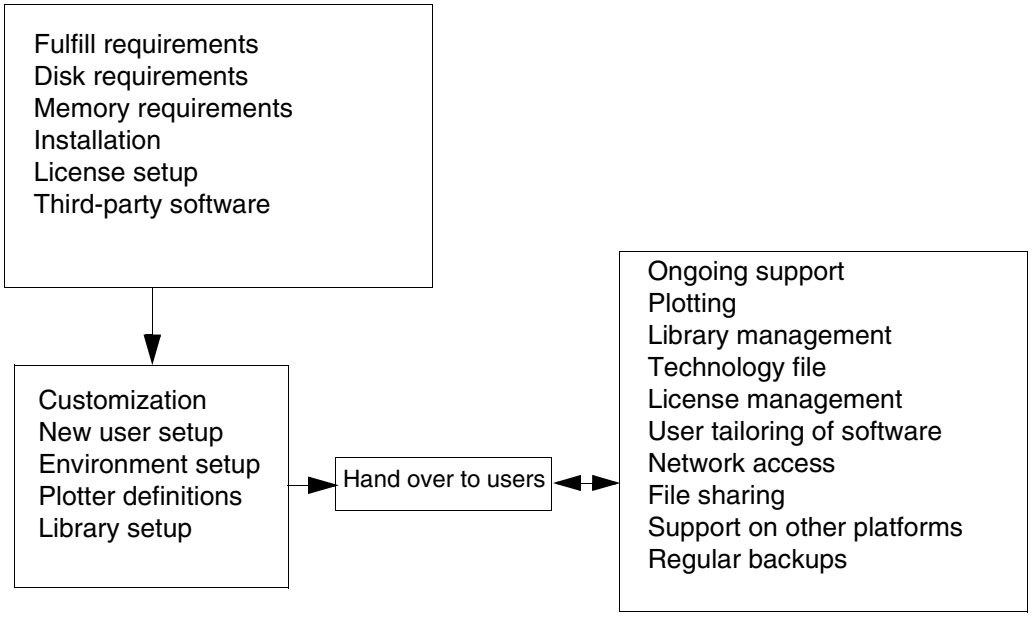
```
dbAccess -load listLibs.il
Virtuoso Framework License (111) was checked out successfully. Total checkout
time was 0.64s.
cdsDefTechLib
basic
```

```
US_8ths
rfLib
rfExamples
ahdlLib
analogLib
functional
```

Note: The above output is generated using the default `cds.lib` from the installation directory.

Additional Virtuoso Software Licensing and Configuration Information

The administration of the software is summarized below.



Note: This document assumes that your operating system and the X Window System are installed and that each user has an account (user name, home directory, `.cshrc`, and `.login` file or `.profile` file). The *Virtuoso Design Environment User Guide* provides more information about customizing these files.

The following information is described in this chapter:

- X Window System on page 110
- Running 64-Bit Versions of Applications on page 111
- TrueColor Visuals on page 114
- Locale Settings on page 116

X Window System

You can find server-dependent directories and files in the following location:

X_install_dir/etc/*

where *X_install_dir* is the directory where your X Window system software is installed.

You can start X in one of the following ways:

1. The workstation runs X all the time (X is running even before you log in). You are using `dtlogin`.

   Note: This document assumes that you are using `dtlogin`.

   Important

   The `dtterm -C` option on some platforms makes the dtterm window act as the console, whereas on other platforms, `dtterm -C` quits with an error.

2. The workstation comes up in text or a proprietary graphics mode. You start X after you log in (without `xdm`).

Running 64-Bit Versions of Applications

Most Cadence® applications have both 32- and 64-bit versions. The 64-bit versions of applications are installed in the same tools hierarchy as the 32-bit versions. A wrapper for each application determines which version of the application is run.

Note: The default setting is to run applications in 32-bit unless the `CDS_AUTO_64BIT` environment variable is set to `ALL` (see below for more information).

To run the 64-bit version of a Cadence application, do the following:

1. Verify that your operating system supports 64-bit applications.

   Note: You should have a minimum of 4GB of RAM to run Cadence 64-bit applications.

2. Verify that a 64-bit version of the application is installed.

   The 64-bit version of an application is located in the `64bit` directory in the standard installation location of the application.

   For example,
   your_install_dir`/tools/bin/64bit/cdsHierEditor`
   your_install_dir`/tools/dfII/bin/64bit/virtuoso`

3. Set the following environment variable:

```
CDS_AUTO_64BIT { ALL | NONE | list | INCLUDE:list | EXCLUDE:list }
```

`ALL`	All applications are run as 64-bit.
`NONE`	All applications are run as 32-bit.
`list`	Only the applications specified are run as 64-bit. Specify *`list`* as a list of case-sensitive application names, separated by a colon, comma, or semi-colon. If you use a semi-colon, enclose the list in quotation marks.
`INCLUDE:`*`list`*	Only the applications specified are run as 64-bit. Specify *`list`* as a list of case-sensitive application names, separated by a colon, comma, or semi-colon. If you use a semi-colon, enclose the list in quotation marks.
`EXCLUDE:`*`list`*	Only the applications specified are run as 32-bit; all other applications are run as 64-bit. Specify *`list`* as a list of case-sensitive application names, separated by a colon, comma, or semi-colon. If you use a semi-colon, enclose the list in quotation marks.

For example, if you have `virtuoso` and `libManager` installed:

`CDS_AUTO_64BIT` is ...	**The following versions are run ...**
`ALL`	`virtuoso: 64-bit; libManager: 64-bit`
`virtuoso`	`virtuoso: 64-bit; libManager: 32-bit`
`INCLUDE:virtuoso`	`virtuoso: 64-bit; libManager: 32-bit`
`EXCLUDE:virtuoso`	`virtuoso: 32-bit; libManager: 64-bit`
`NONE`	`virtuoso: 32-bit; libManager: 32-bit`

Important

Setting `CDS_AUTO_64BIT` does not guarantee that you will run the 64-bit version of an application. The wrapper runs the 64-bit version of the application only if all the following conditions are true:

- The operating system supports 64-bit applications.
- A 64-bit version of the application is installed.
- You choose to run the 64-bit version by setting the `CDS_AUTO_64BIT` environment variable.

  Note: Applications can override the `CDS_AUTO_64BIT` variable with an application-specific variable. See the application's documentation for more information.

Otherwise, the 32-bit version of the application is run.

4. Start the Cadence application from its standard location.

   For example, for the virtuoso executable,
   your_install_dir`/tools/dfII/bin/`*virtuoso*.

   or for the Cadence Hierarchy Editor

   your_install_dir`/tools/bin/64bit/cdsHierEditor`

   The wrapper for the application assess whether to run the 32- or 64-bit version of the application based on the value of `CDS_AUTO_64BIT`.

   The corresponding versions of the applications reside in subdirectories of the `bin` directory. For example, the 32-bit version of *appName* is in *your_install_dir*`/tools/bin/32bit/`*appName*.

Caution

Do not run the executables in the `32bit` or `64bit` directories directly (always run the application through its wrapper).

Note: See your application documentation for more information. In some cases, you might need to start *appName.exe* instead of *appName* to run the application through its wrapper.

Tip

When you start an application, you can use the `-debug3264` option to assess whether you are running the 32- or 64-bit version. Diagnostic information is displayed before the program starts.

In the following example, the 32-bit version of the libManager executable was run even though `CDS_AUTO_64BIT` was set to `ALL`. The output of the `-debug3264` option

indicates that the 32-bit version was run because the 64-bit version of the application was not installed.

```
% setenv CDS_AUTO_64BIT ALL
% libManager.exe -debug3264
-------- 32/64 bit wrapper diagnostics -------------
App name: libManager.exe
App path: /net/machine/cds/5.0.0/tools/dfII/bin
OS is 64-bit capable.
The user has selected 64-bit operation via the environment variables.
No 64-bit version of the application exists.
A 32-bit version of the application exists.
----------------------------------------------------
PATH            :
/net/ansbk/usr/SoftWindows.solaris/bin:/mnt3/ns/bin:/usr/bin/X11:/usr/openwin
/bin:.:/usr/ucb:/bin:/mnt3/ns/bin:/usr/local/bin:/usr/bin:/usr/etc:/usr/local
:/usr/lang/v3:/net/machine/cds/5.0.0/tools/bin:/net/machine/cds/5.0.0/tools/d
fII/bin:/net/machine/cds/5.0.0/tools/lib:/opt/SUNWdtpcv/bin:/usr/local/pvt
LD_LIBRARY_PATH :
/net/machine/cds/5.0.0/tools/lib:/usr/lib:/usr/openwin/lib:/usr/lib/X11:/usr/
dt/lib
CDS_AUTO_64BIT  : ALL
Launching "/net/machine/cds/5.0.0/tools/dfII/bin/32bit/libManager.exe"
----------------------------------------------------
```

TrueColor Visuals

The software defaults to a 24-planes TrueColor visual. If a 24-planes visual is not available, the software will first search for a 16-planes TrueColor visual, then a 15-planes TrueColor visual.

Note: One of these graphics display visuals must be available in order to run TrueColor visuals.

Finding Available Visuals

To check for available graphics display visuals, do the following.

➤ At the system prompt, type `xdpyinfo`

The following shows the results of `xdpyinfo` for a typical 24-planes TrueColor visual:

```
visual:
visual id:    0x2e
class:    TrueColor
```

```
depth:    24 planes
available colormap entries:    256 per subfield
red, green, blue masks:    0xff, 0xff00, 0xff0000
significant bits in color specification:    8 bits
```

Pseudocolor and TrueColor Visuals

In previous releases, the software supported Pseudocolor visuals. Pseudocolor visuals allowed color mapping (mapping between the pixels used to define the colors and the colors that display). In the IC 6.1 release, only TrueColor visuals are supported. TrueColor provide a full palette of colors which increase with larger numbers of planes (bits). In other words, the incremental difference between each color is smaller with the more planes you have.

A 15-plane TrueColor visual provides 32,768 colors and uses 5 bits each to describe the red component, green component, and blue components. A 16-plane TrueColor visual provides 65,535 colors and uses 6 bits to describe the red component, 5 bits to describe the green component, and 5 bits to describe the blue component. The 24-plane TrueColor visual provides 16,777,216 colors and uses 8 bits to describe each component of the red, green, and blue.

The following table shows how the total number of planes (or bits) are used to describe each of the components; red, green, and blue for TrueColor visuals.

15 Planes Color Depth

Color	r	r	r	r	r	g	g	g	g	g	b	b	b	b	b
Bits	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

16 Planes Color Depth

Color	r	r	r	r	r	r	g	g	g	g	g	b	b	b	b	b
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

24 Planes Color Depth

Color	r	r	r	r	r	r	r	r	g	g	g	g	g	g	g	g	b	b	b	b	b	b	b	b
Bits	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Locale Settings

Virtuoso software requires locale settings to be set to `C` in order to work properly. When you run Virtuoso software on any platform, the software sets the locale to `C` automatically. Your original locale settings are ignored.

This also means that for a script, shell, or other tool that is run from a Virtuoso application (for example, a shell invoked from the CIW), the locale setting that is used is always `C`, regardless of any locale variables set in your environment.

If you want to restore your original locale settings for a script, shell, or tool that is invoked from a Virtuoso application, source one of the following scripts first:

For Bourne shells:

your_install_dir`/tools/dfII/samples/local/cdsOrigLocale.sh`

For C shells:

your_install_dir/tools/dfII/samples/local/cdsOrigLocale.csh

For example,

source *your_install_dir*/tools/dfII/samples/local/cdsOrigLocale.csh

or

. *your_install_dir*/tools/dfII/samples/local/cdsOrigLocale.sh

You can also do this from the `system` command. For example, to run dtpad, you could use the following command:

```
system( strcat( ". " prependInstallPath( "samples/local/cdsOrigLocale.sh" )
            "; /usr/dt/bin/dtpad -standAlone" ) )
```

A

Licensing Environment Variables

This appendix describes the licensing variables that you can use to customize the default license settings. You can set the values of these environment variables in the `.cdsenv` file.

The following example illustrates an environment variable setting in the `.cdsenv` file.

```
license VLSLicenseCheckoutOrder string "XL, GXL, EXL"
```

In each entry, the first column is the tool partition, the second column is the variable, the third column is the data type, and the fourth column contains the value to be used. For more information, see *Specifying Environment Settings* in *Virtuoso Design Environment User Guide*.

You can also set and retrieve the values of these environment variables using the `envSetVal` and `envGetVal` functions in Virtuoso CIW. For example:

```
envSetVal("license" "VLSLicenseCheckoutOrder" 'string "XL, GXL, EXL")
envGetVal("license" "VLSLicenseCheckoutOrder")
```

This section describes the environment variables available in the `license` partition of the `.cdsenv` file:

- CheckoutOrder Variables on page 120
- UseNextLicense Variables on page 126
- Other Licensing Variables on page 131

CheckoutOrder Variables

These variables control the license checkout order of Virtuoso tier-based applications. The values specified in these variables govern the order in which higher-tier licenses are checked out when an attempt to check out a lower-tier license fails.

For each `CheckoutOrder` variable, a set of valid values can be created using a combination of values from the default value list, with commas or spaces as delimiters. However, an empty string is not allowed as a valid value for these variables, except `EADLicenseCheckoutOrder`.

For example, the `adeMaestroCheckoutOrder` variable can have the following valid values:

- "`Classic, Maestro`" looks for an ADE license first. If that is not available, a Maestro license is checked out
- "`Maestro, Classic`" looks for a Maestro license first. If that is not available, an ADE license is checked out.
- "`ADE`" looks only for an ADE license.
- "`Maestro`" looks only for a Maestro license.

The following table lists the `CheckoutOrder` variables available in the `license` partition:.

Variable	Details
`VLSLicenseCheckoutOrder`	
	Syntax: **`license VLSLicenseCheckoutOrder string "`**`XL, GXL, EXL`**`"`** ***Description***: Specifies the license checkout order preference for the Virtuoso Layout Suite (VLS) product family. ***Default Value***: `XL,GXL,EXL`

Variable	Details
VSELicenseCheckoutOrder	
	Syntax: **license VSELicenseCheckoutOrder string "**L, XL**"** ***Description***: Specifies the license checkout order preference for the Virtuoso Schematic Editor (VSE) product family. ***Default Value***: L,XL
ADELicenseCheckoutOrder	
	Syntax: **license ADELicenseCheckoutOrder string "**L, XL, GXL**"** ***Description***: Specifies the license checkout order preference for the Virtuoso Analog Design Environment (ADE) product family. ***Default Value***: L,XL,GXL
VLSAdvOptLicenseCheckoutOrder	
	Syntax: **license VLSAdvOptLicenseCheckoutOrder string "**95511, 95512**"** ***Description***: Specifies the license checkout order preference for Virtuoso advanced node features. ***Default Value***: 95511,95512

Variable	Details
maestroCheckoutOrder	
	Syntax: **license maestroCheckoutOrder string "**Explorer, Assembler**"** ***Description***: Specifies the license checkout order preference for Maestro (Virtuoso ADE Explorer or Virtuoso ADE Assembler) licenses. ***Default Value***: Explorer,Assembler

Variable	Details
`adeMaestroCheckoutOrder`	
	Syntax: **`license adeMaestroCheckoutOrder string`** `"Classic, Maestro"` ***Description***: Specifies the license checkout order preference for Virtuoso ADE and Maestro products. If you do not have ADE L, XL, or GXL licenses, but have the Virtuoso ADE Explorer or Virtuoso ADE Assembler license, then these licenses can also be used to run ADE L, XL, and GXL. To control which licenses are used and in which order to run ADE L, XL, or GXL, you can set this variable to one of the following values: ***Default Value***: `Classic, Maestro` ❑ When set to default, ADE L, XL or GXL licenses will be searched first to run these tools. If these licenses are not available, ADE Explorer and ADE Assembler licenses will be searched and used when found. Other valid values for this variable are: ❑ `ADE`: When you set this variable to ADE, only ADE L, XL, or GXL licenses will be searched and used to run ADE L, XL, or GXL. It's important to note that, with this setting, ADE Explorer or ADE Assembler licenses will never be used to run ADE L, XL, or GXL. ❑ `Maestro, Classic`: When this value is set, ADE Explorer and ADE Assembler licenses will be searched and used first. If they are not available, ADE L, XL, and GXL licenses are searched and used. ❑ `Maestro`: Only Virtuoso ADE Explorer and Virtuoso ADE Assembler license are searched and used. In this case, ADE Explorer or Assembler licenses are always used to run ADE L, XL, and GXL.

Variable	Details
	Note: `Maestro` and `Classic` options honor the `maestroCheckoutOrder` and `ADELicenseCheckoutOrder` variables, respectively.
`AMSEnvLicenseCheckoutOrder`	
	Syntax: **`license AMSEnvLicenseCheckoutOrder string`** `"AMS, ADE"` ***Description***: Specifies the license checkout order to be considered while running the *Check and Save* feature of Text Editor for verilogams and SystemVerilog views. By default, this feature first looks for the `AMS_Environment` license (Product number `70000`). When the `AMS_Environment` license is not available, an ADE license, that is, `Classic` (`L`, `XL`, or `GXL`) or `Maestro` (Virtuoso ADE Explorer or Virtuoso ADE Assembler) license is checked out. ***Default Value***: `AMS, ADE` **Note:** `AMSEnvLicenseCheckoutOrder` replaces the `skipAMSEnvironmentLicCheck`, which will be removed from a future release. Currently, when `skipAMSEnvironmentLicCheck` is set to `t`, AMS is removed from the checkout order.

Variable	Details
VIVALicenseCheckoutOrder	***Syntax***: **license VIVALicenseCheckoutOrder string "**VIVA, ADE**"** ***Description***: Specifies the license checkout order preference for Virtuoso Visualization and Analysis XL. By default, this variable is set to VIVA,ADE. If the VIVA license is not available, an ADE license, that is, Classic (L, XL, or GXL) or Maestro (Virtuoso ADE Explorer or Virtuoso ADE Assembler) license is checked out. **Note:** You must set this environment variable in .cdsinit or .cdsenv file. Do not set this environment variable in the CIW. ***Default Value***: VIVA,ADE ***See Also***: For more information, see VIVALicenseCheckoutOrder in the *Virtuoso Visualization and Analysis XL User Guide*.

UseNextLicense Variables

If a license for a requested application is not available, the Use Next License dialog box is displayed, which lets you check out a higher tier license, subject to availability.

These variables control whether to check out the next available license for Virtuoso tier-based applications, when the license required for a requested application is not available.

Depending on the value of this variable, you are given an opportunity to check out a higher tier license, subject to availability.

These environment variables can have the following values:

Value	Use
prompt	Prompts for confirmation before checking out the next tier license. **Note**: During a Virtuoso session, the Next License dialog appears only once for each application. When `prompt` is set as the current value, the Next License dialog is displayed asking you to confirm before checking out the next tier license.
always	Always tries to check out the next tier license when the requested license is not available.
never	Never attempts to check out the next tier license and displays an error message instead.

Note: If the value of these variables is set as an empty string " " (except for `EADLicenseCheckoutOrder`) or has an incorrect variable value, the license tool prints an appropriate CIW warning message about the specific issue. In both cases, the previous checkout order is restored.

Note: If Virtuoso is currently being run in the `-nograph` mode, the license order mechanism will not display the Use Next License dialog if a specific license is not available, and the value of `*_UseNextLicense` is set to `prompt`. Instead, a warning message will be added to the log file.

The following table lists the `UseNextLicense` variables available in the `license` partition

Variable	Details

`VLSXL_UseNextLicense`	***Syntax***:
	`license VLSXL_UseNextLicense string "prompt"`
	Description:
	Sets the `UseNextLicense` value that controls whether to check out the next available license (Virtuoso_Layout_Suite_GXL) for Virtuoso Layout Suite XL or not.
	Default Value:
	`prompt`
	Valid Values:
	`prompt, always, never`

`VSEL_UseNextLicense`	***Syntax***:
	`license VSEL_UseNextLicense string "prompt"`
	Description:
	Sets the `UseNextLicense` value that controls whether to check out the next available license (Virtuoso_Schematic_Editor_XL) for Virtuoso Schematic Editor L or not.
	Default Value:
	`prompt`
	Valid Values:
	`prompt, always, never`

Variable	Details
ADEL_UseNextLicense	***Syntax***: **license ADEL_UseNextLicense string "**prompt**"** ***Description***: Sets the UseNextLicense value that controls whether to check out the next available license (Analog_Design_Environment_XL) for Virtuoso Analog Design Environment L or not. ***Default Value***: prompt ***Valid Values***: prompt, always, never
ADEXL_UseNextLicense	***Syntax***: **license ADEXL_UseNextLicense string "**prompt**"** ***Description***: Sets the UseNextLicense value that controls whether to check out the next available license (Analog_Design_Environment_GXL) for Virtuoso Analog Design Environment XL or not. ***Default Value***: prompt ***Valid Values***: prompt, always, never

Variable	Details

VLSAdvOpt_UseNextLicense	***Syntax***:
	`license VLSAdvOpt_UseNextLicense string "prompt"`
	Description:
	Sets the `UseNextLicense` value that controls whether to check out the next available license (95512) for Virtuoso advanced node applications or not.
	Default Value:
	`prompt`
	Valid Values:
	`prompt, always, never`

VIVA_UseNextLicense	***Syntax***:
	`license VIVA_UseNextLicense string "prompt"`
	Description:
	Sets the `UseNextLicense` value that controls whether to check out the next available license (Maestro, ADE), as per the specified checkout order for Virtuoso Visualization and Analysis XL or not.
	Default Value:
	`prompt`
	Valid Values:
	`prompt, always, never`

Variable	Details
ADE_UseNextLicense	***Syntax***: **license ADE_UseNextLicense string "**prompt**"** ***Description***: Sets the UseNextLicense value that controls whether to check out the next available license (Maestro: Virtuoso ADE Explorer/Virtuoso ADE Assembler) for Virtuoso Analog Design Environment product tiers or not. ***Default Value***: prompt ***Valid Values***: prompt, always, never ***See Also***: adeMaestroCheckoutOrder
Explorer_UseNextLicense	***Syntax***: **license Explorer_UseNextLicense string "**prompt**"** ***Description***: Sets the UseNextLicense value that controls whether to check out the next available license (Virtuoso ADE Assembler) for Virtuoso ADE Explorer or not. ***Default Value***: prompt ***Valid Values***: prompt, always, never

For applications in product family tiers (such as, Layout, Schematic, and ADE), the checkout behavior is governed by both CheckoutOrder and UseNextLicense variable settings. If the license required for an application as per the CheckoutOrder is not available, the UseNextLicense variable for the product family controls whether the higher-tier license should be checked out or not.

For example, if ADELicenseCheckoutOrder is set to "XL, L, GXL" and *_UseNextLicense is set to "never" and an XL license is not available, then, running ADE

L will fail as `*_UseNextLicense` is set to `never` and `ADELicenseCheckoutOrder` is set to stop at `XL`.

The following table lists the product families and the associated `CheckoutOrder` and `UseNextLicense` variables:

Product Family	CheckoutOrder Variable	UseNextLicense Variable
Layout	`VLSLicenseCheckoutOrder`	`VLSXL_UseNextLicense`
Schematic	`VSELicenseCheckoutOrder`	`VSEL_UseNextLicense`
ADE	`ADELicenseCheckoutOrder`	`ADEL_UseNextLicense`, `ADEXL_UseNextLicense`
Virtuoso Advanced Node	`VLSAdvOptLicenseCheckoutOrder`	`VLSAdvOpt_UseNextLicense`
Maestro	`maestroCheckoutOrder`	`Explorer_UseNextLicense`
ADE/Maestro	`adeMaestroCheckoutOrder`	`ADE_UseNextLicense`
VIVA	`VIVALicenseCheckoutOrder`	`VIVA_UseNextLicense`

Other Licensing Variables

Variable	Details
`skipAMSEnvironmentLicCheck`	
	Syntax: **`license skipAMSEnvironmentLicCheck boolean`** `nil` ***Description***: Skips AMS_environment license check and uses ADE licenses instead. ***Default Value***: `nil` **Note:** This environment variable will be deprecated in a future release. Instead, use `AMSEnvLicenseCheckoutOrder`.