

Keysight Engineering Data Management (SOS) Core: Integration with Cadence Virtuoso

Seamless Design Data Management for Analog, RF, and Mixed-Signal IC Design Teams

Table of Contents

Executive Overview 3

SOS Core: Platform Overview 4

The Cadence Virtuoso Integration 5

Visual Design Diff (VDD) 12

Hierarchical Operations 14

Labels, RSO, and Automated Design Handoffs 16

Library-Level Diff Reports 17

Audit Trail 17

Customization and SKILL APIs 18

Real-World Workflow: A Multi-Engineer Design Scenario 19

Why SOS for Virtuoso..... 21

Scaling Your Approach 22

Conclusion 22

Executive Overview

Cadence Virtuoso is a commonplace design environment for analog, RF, and mixed-signal IC design. Tens of thousands of engineers worldwide rely on Virtuoso for schematic capture, simulation setup, layout editing, and physical verification. In team-based design projects, the ability to manage design data reliably within Virtuoso is not a convenience; it is a prerequisite for predictable tapeout schedules, design integrity, and engineering productivity.

Keysight Engineering Data Management (SOS) Core provides a native, deeply integrated data management layer within the Cadence Virtuoso environment. Rather than requiring engineers to switch between their design tool and a separate version control application, the SOS Virtuoso integration embeds all core data management operations directly into the Virtuoso Library Manager and editor windows. Checkout, check-in, update, diff, tagging, hierarchical operations, and audit trail are all accessible from within the menus and context menus that Virtuoso designers already use.

This application note provides a comprehensive overview of SOS Core, followed by a detailed examination of its integration with Cadence Virtuoso. It covers the integration architecture, day-to-day designer workflows, advanced capabilities such as Visual Design Diff (VDD) and hierarchical operations, and the customization mechanisms that allow CAD teams to tailor the integration to their specific process requirements.

SOS Core: Platform Overview

SOS Core is Keysight's design data management platform built for small to mid-sized engineering teams that need performance, reliability, and simplicity. It provides high-performance version control of design data, seamless collaboration across teams and workflows, and referencing mechanisms for IP reuse-driven design.

SOS Core delivers enterprise-grade design data management in a streamlined, easy-to-deploy package. It empowers small to mid-sized design teams with fast, reliable version control, seamless EDA integration, and metadata-driven IP reuse.

Core Capabilities

Version control and data integrity. Every modification to a managed file creates a new, immutable revision in the central repository. The system maintains a complete history of every design object: who changed it, when, and what was changed. Status icons in the SOS GUI and within supported EDA environments indicate whether each object is current, out of date, checked out by you, or locked by another user.

Check-in, check-out, and preemptive locking. SOS Core implements a check-out/check-in workflow with preemptive locking, the standard concurrency model for hardware design data where concurrent modification of binary files is impractical. When an engineer checks out a design object, they receive write access while a lock prevents anyone else from modifying it until the object is checked back in.

Labels and Revision Search Order (RSO). Labels are human-readable aliases assigned to specific revisions (e.g., "design_done", "Gold"). The Revision Search Order (RSO) is a priority list of labels associated with each user workarea that controls which revision of each object is retrieved during a populate or update operation. This mechanism automates design handoffs between functional roles without manual coordination.

Snapshots. Point-in-time captures of the entire project state. If subsequent changes introduce problems, the team can revert the entire project to a snapshot state. Snapshots serve as a reliable baseline for tapeout preparation, milestone reviews, or risk mitigation.

Links-to-Cache architecture. User workareas contain symbolic links that point to the single copy of each file stored in a local Cache, rather than full physical copies. A workarea for a 100 GB project may consume only a few megabytes. This delivers dramatic storage savings and eliminates time-consuming workspace population steps.

IP reuse through references. Rather than copying IP between projects, SOS Core creates managed references that preserve provenance, traceability, and update control.

EDA tool integrations. SOS Core maintains the broadest portfolio of EDA integrations in the design data management category, including native integrations with Cadence Virtuoso, Synopsys Custom Compiler, Siemens Pyxis and Tanner, Keysight ADS, Silvaco Gateway + Expert, MathWorks MATLAB, and Empyrean Aether.

The Cadence Virtuoso Integration

The SOS integration with Cadence Virtuoso is the most mature and feature-rich EDA integration in the SOS portfolio. It transforms the Virtuoso Library Manager and editor windows into a fully data-managed design environment, providing version control, collaboration, and traceability without requiring engineers to leave their primary design tool.

Integration Architecture

The SOS Virtuoso integration is loaded at Virtuoso startup through configuration files that extend the Cadence environment. Four files must be configured: `.cdsinit` (SKILL initialization), `cdsLibMgr.il` (Library Manager extensions), `cdsinfo.tag` (library type declaration), and `cds.lib` (library path definitions). Once configured, the integration injects the Design Manager menu into the Library Manager and editor windows, registers status icon callbacks, and connects the Virtuoso session to the SOS server.

The integration communicates with the SOS server through the SOS client libraries, which connect either to a local Cache Repository or directly to the Primary Repository depending on the deployment topology. For multi-site teams, the Cache architecture ensures that Virtuoso operations resolve against a local data store, delivering local-speed performance regardless of where the Primary Repository is hosted.

Library management under SOS is controlled by the `cdsinfo.tag` file within each library directory. The line `"DMTYPE sos"` (case-sensitive, lowercase) tells Virtuoso that SOS manages the library. This declaration enables all SOS operations for that library within the Virtuoso interface. Libraries without this declaration remain unmanaged and behave as standard Virtuoso libraries.

User Interface Overview

The Design Manager Menu

The Design Manager menu appears in both the Library Manager window and in Virtuoso editor windows (schematic editor, layout editor). It provides access to all SOS operations a designer needs during daily work [figure 1]. The menu is organized into logical groups covering basic operations (Check In, Check Out, Cancel Checkout, Tag, Merge), status and history queries (Show Status, Version Info, History, Visual Design Diff, List Revision Diffs, Show Labeled), hierarchical operations (Manage Hierarchy submenu), the SOS Design Manager alternative interface, workspace operations (Populate, Update, Delete, Rename), and administration and customization (Audit Trail, Preferences, Modify Attribute, Open SOS).

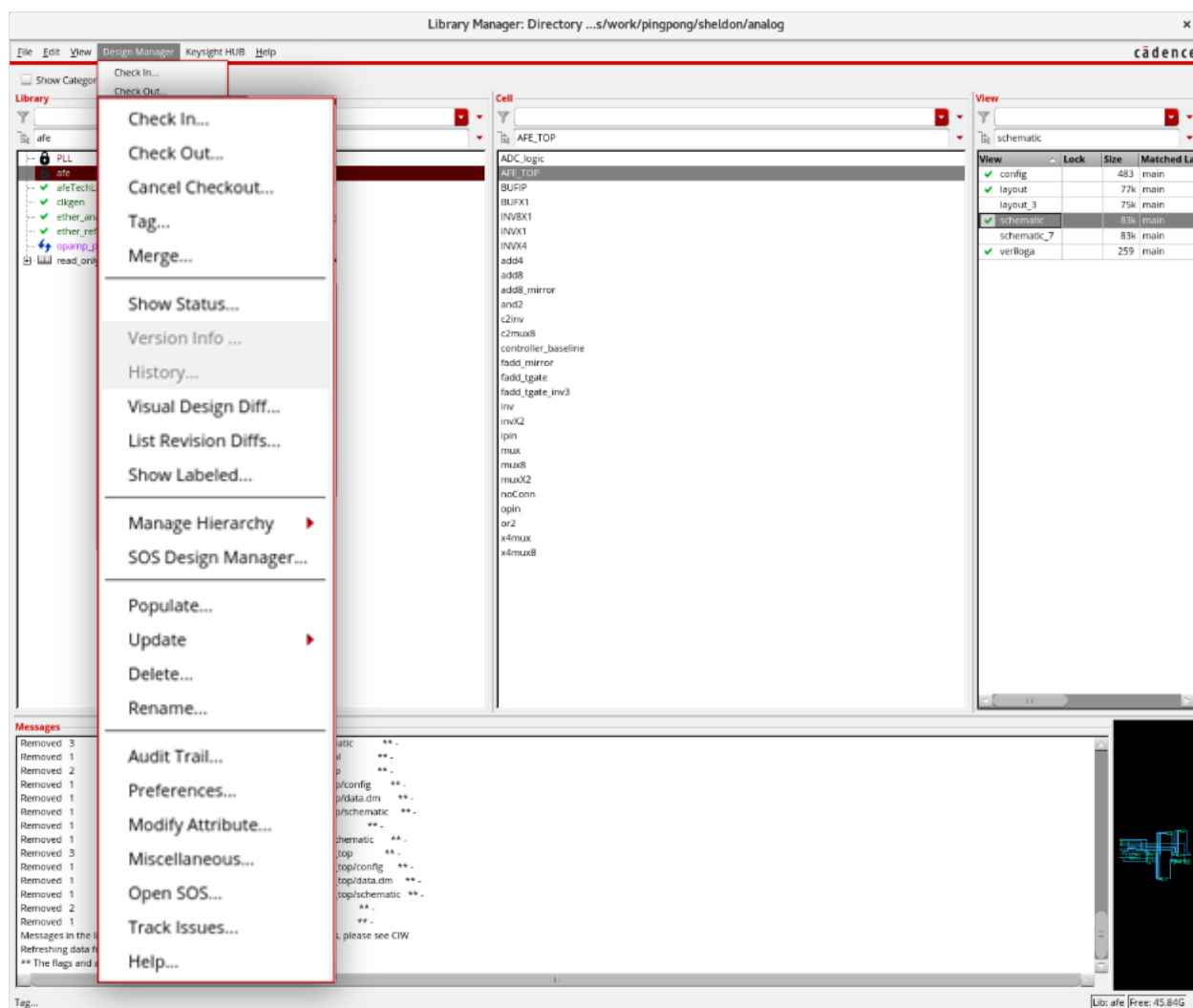


Figure 1. The Design Manager menu provides access to all SOS operations

Status Icons

SOS communicates the version control status of every managed cellview through color-coded icons that appear directly in the Virtuoso Library Manager's View panel. These icons provide at-a-glance awareness of the project state without requiring any additional tool interaction.

Icon	Status	Meaning
	Checked in	The cellview is up to date and available for checkout.
	Checked out by you	You have the cellview locked for editing. Other users cannot modify it

	Checked out by another user	Another engineer has the cellview locked. You can open it read-only.
	Out of date	A newer version exists in the repository. You should update your work area

Hovering over any cellview in the Library Manager displays a tooltip with additional detail, such as who has the object checked out or why an update is needed [figure 2]. For Virtuoso versions prior to 6.1.5, the Design Manager > Show Status command provides the same information in a tabular dialog.

View	Lock	Size
✓ extracted		41k
 layout		31k
✓ schematic		40k
✓ s		24k

Name: afe/c2mux8/layout
DM state: checked out: leslie
Version: 3
Size: 30752

Figure 2. Checking the version control status of every managed cellview

Right-Click Context Menu

The most frequently used SOS operations are also available through the right-click context menu on libraries, cells, and views in the Library Manager [figure 3]. The context menu is scope-sensitive: right-clicking a cellview offers operations at the cellview level, while right-clicking a library applies operations across all objects in that library. This context menu is fully customizable through the SOS Preferences dialog (RMB Customization tab).

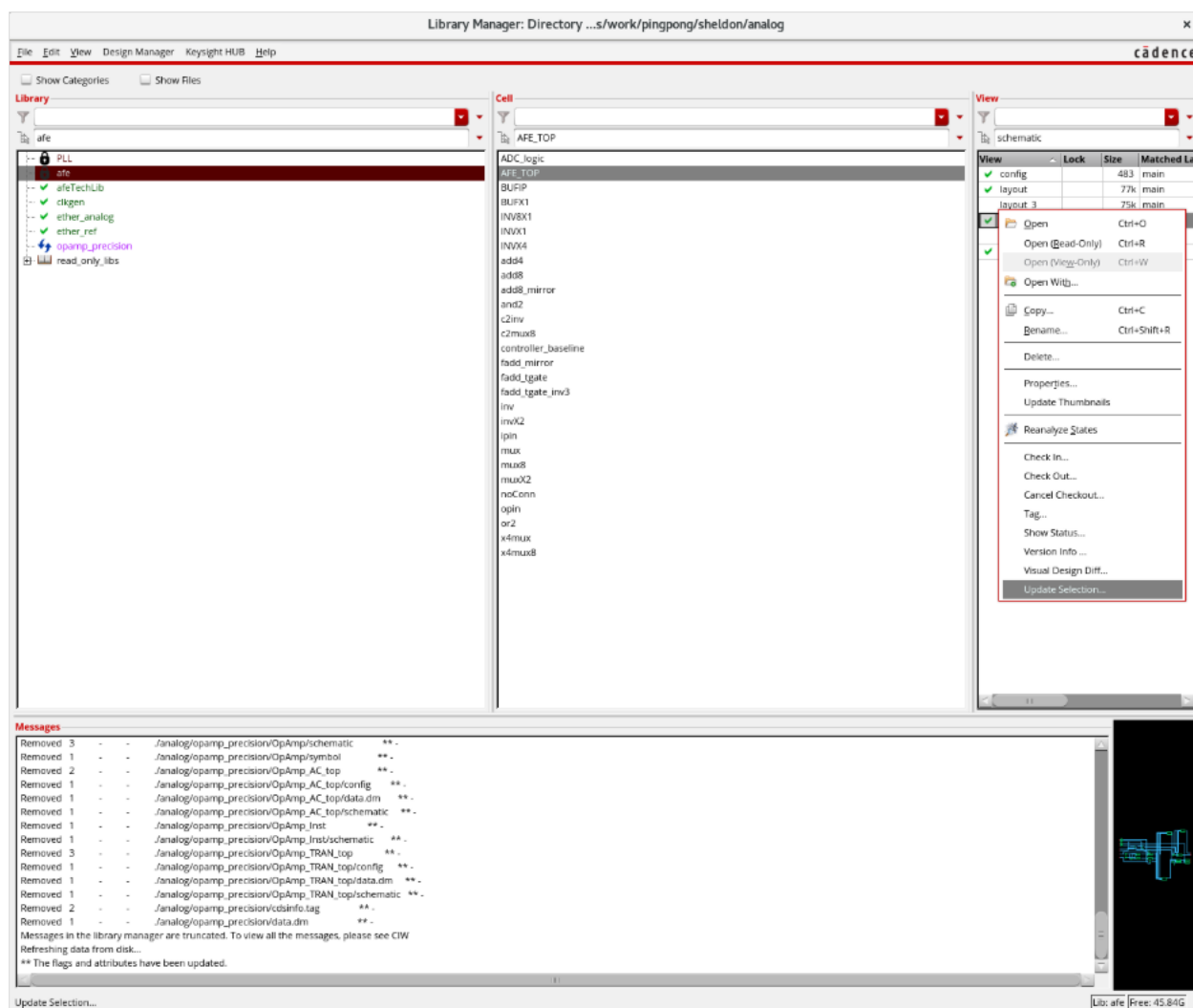


Figure 3. Right-click context menu gives quick access to frequently used SOS operations

The SOS Design Manager

For power users and CAD managers who need to operate on large sets of objects, SOS provides the SOS Design Manager as an alternative to the Cadence Library Manager [figure 4]. The SOS Design Manager supports multi-selection across libraries, cells, and views, which the standard Cadence Library Manager does not. It features a tree browser showing the library hierarchy with propagated status icons, a tabular view with sortable columns and advanced filters, summary statistics at the bottom of the window (selected count, unmanaged count, checked-in count, checked-out count, out-of-date count), and a "where used" report that identifies which cellviews reference a selected object.

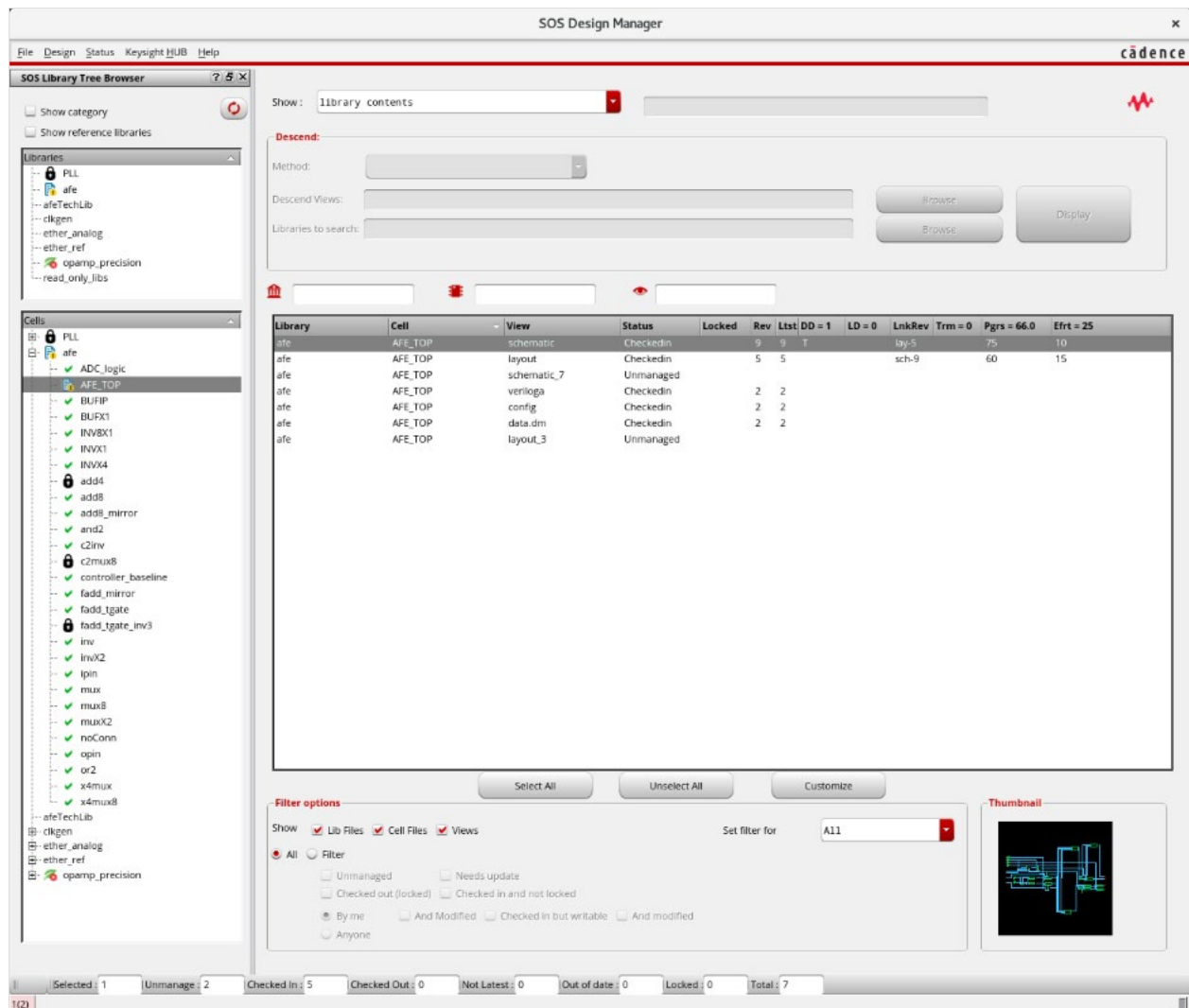


Figure 4. SOS Design Manager can be used as an alternative to the Cadence Library Manager

Core Designer Workflows

The SOS Virtuoso integration is designed to feel natural within the standard Virtuoso design flow. Engineers open, edit, save, and close cellviews with the same Virtuoso commands they are accustomed to. SOS intercepts these operations transparently: opening a cellview triggers an auto-checkout prompt, and closing or saving a cellview can trigger check-in workflows.

Typical Daily Task Flow

A typical daily workflow for a Virtuoso designer using SOS follows a five-step cycle.

Step 1: Check out a cellview. When the designer opens a cellview for editing, Virtuoso presents an Auto Checkout dialog. Clicking OK locks the cellview in the repository and provides the designer with a writable copy in their workarea. While checked out, no other user at any site can modify the same cellview.

Step 2: Edit the cellview and save changes. The designer works normally in the Virtuoso schematic or layout editor, saving changes to disk periodically. All saves are local to the designer's workarea and do not affect other users.

Step 3: Verify the changes. The designer runs simulations, DRC, LVS, or other verification steps on the modified cellview before committing changes to the repository.

Step 4: Check in the cellview. From the Library Manager, the designer selects Design Manager > Check In. The Check In form displays the objects being committed, allows the designer to enter a description of the changes, and optionally apply a tag. Clicking OK creates a new revision in the repository and releases the lock, making the updated cellview available to the rest of the team.

Step 5: Update the workarea. The designer periodically synchronizes their workarea to pull in changes from other engineers. The Design Manager > Update menu provides two options: Update Selection (for specific libraries, cells, or views) and Update Workarea (for the entire workspace). A Preview button allows the designer to review pending changes before applying them.

Designers may repeat this cycle once or several times per day, depending on the scope of changes in each iteration.

Canceling a Checkout

If a designer decides to discard their edits, they can cancel the checkout from the Library Manager right-click menu. The Cancel Checkout dialog offers multiple scope options: discard by the current user in the current workarea, discard by the current user in any workarea, or (for administrators) discard checkouts by any user. The "Do not discard if modified" safety option is enabled by default and must be explicitly deselected to confirm the discard.

Adding New Cellviews to a Project

Cellviews that do not yet belong to an SOS project are labeled "Unmanaged" in the Check In dialog. Designers can add new cellviews to the project either directly from an open editor window (Design Manager > Check In) or in bulk from the Library Manager by selecting the parent library and using the Check In command with the Unmanaged filter enabled. When adding an entirely new library to a project, the designer must ensure the cdsinfo.tag file contains the "DMTYPE sos" declaration.

Version History, Revert, and Recovery

SOS maintains the complete revision history of every managed cellview. The Version Info dialog, accessible by right-clicking a cellview in the Library Manager, displays every revision with its author, timestamp, change description, and associated tags or snapshots [figure 5].

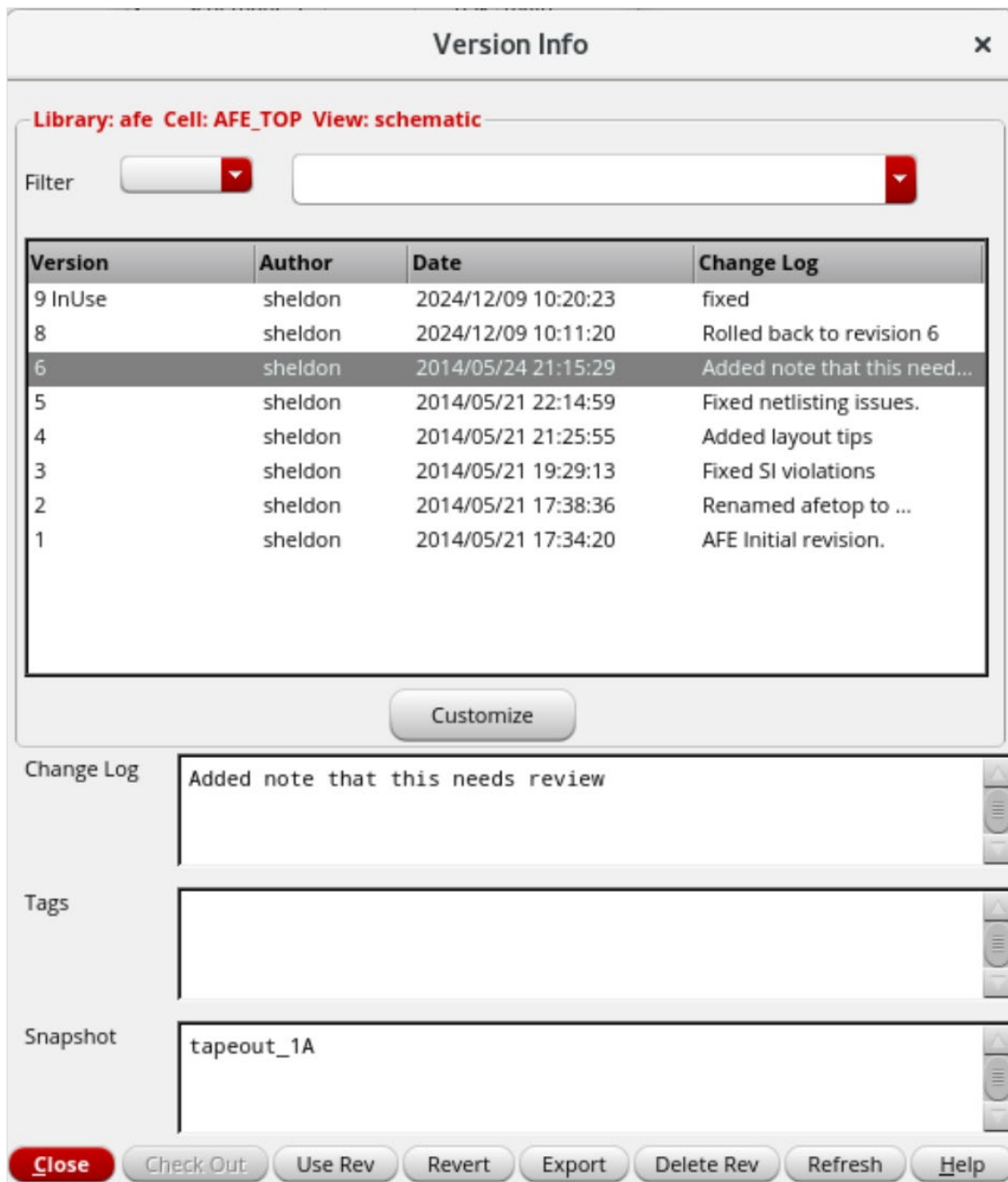


Figure 5. Every revision, such as author, timestamp, change description, and associated tags or snapshots can be accessed through the Version Info dialog – history at a glance.

From the Version Info dialog, a designer can perform three recovery operations. "Use Rev" temporarily switches the designer's workarea to use an earlier revision of the cellview without affecting other users. "Revert" creates a new revision in the repository that is identical to a selected earlier revision, effectively rolling back the object for all users. "Export" extracts a specific revision as an unmanaged, editable copy outside of SOS control.

Deleted cellviews that were previously managed by SOS can always be recovered through the SOS GUI. Only unmanaged cellviews that were never checked in are deleted permanently.

Visual Design Diff (VDD)

Visual Design Diff is one of the most valued capabilities in the SOS Virtuoso integration. Some customers adopt it as a standalone tool independent of the broader SOS platform. VDD enables side-by-side visual comparison of two cellview revisions, or two entirely different cellviews, directly on the schematic or layout canvas.

How VDD Works

To launch VDD, a designer selects Design Manager > Visual Design Diff (or Design Manager > Diff from an open editor window). The Version Compare dialog allows selection of the two revisions to compare [figure 6]. For SOS-managed cellviews, the revision list is populated automatically from the repository history. Non-managed cellviews can also be compared.

Once the comparison executes, VDD presents its results in two coordinated views. The Differences Browser is a structured tree listing every detected change: added, deleted, and modified objects, categorized by type (nets, instances, labels, user properties). The Design Canvas displays the cellview with color-coded halos highlighting each difference: green for added objects, magenta for deleted objects, and yellow for modified objects. Clicking any entry in the Differences Browser zooms the canvas to center on the corresponding object.

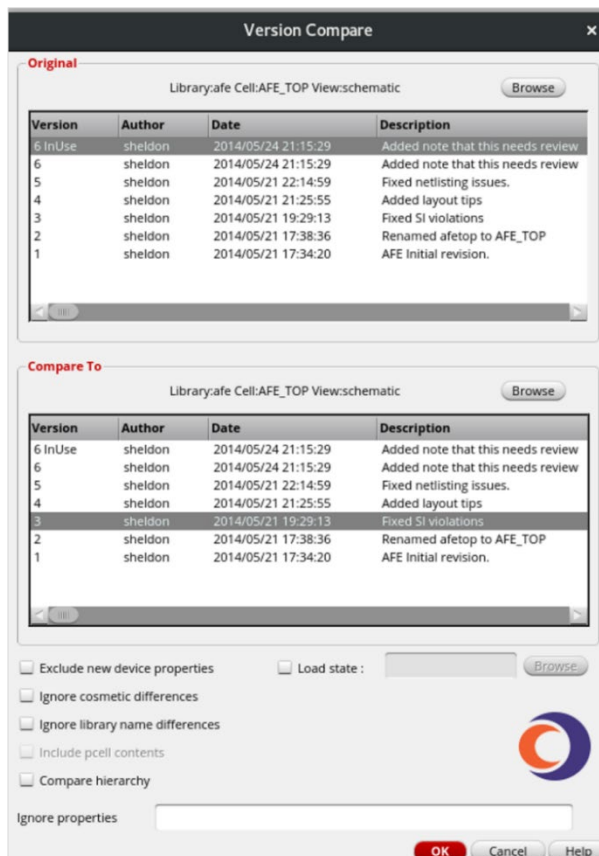


Figure 6. Compare 2 cellviews using Visual Design Diff (VDD)

Schematic Diff

For schematic cellviews, VDD is connectivity-aware. It compares the netlist-level semantics of the two versions, not merely geometric positions of wires and symbols [figure 7]. This means that a cosmetic repositioning of a wire that preserves the same electrical connectivity is not reported as a functional change. The Differences Browser reports added/deleted nets, added/deleted/modified instances, changes to instance properties, and modifications to labels and user-defined attributes.

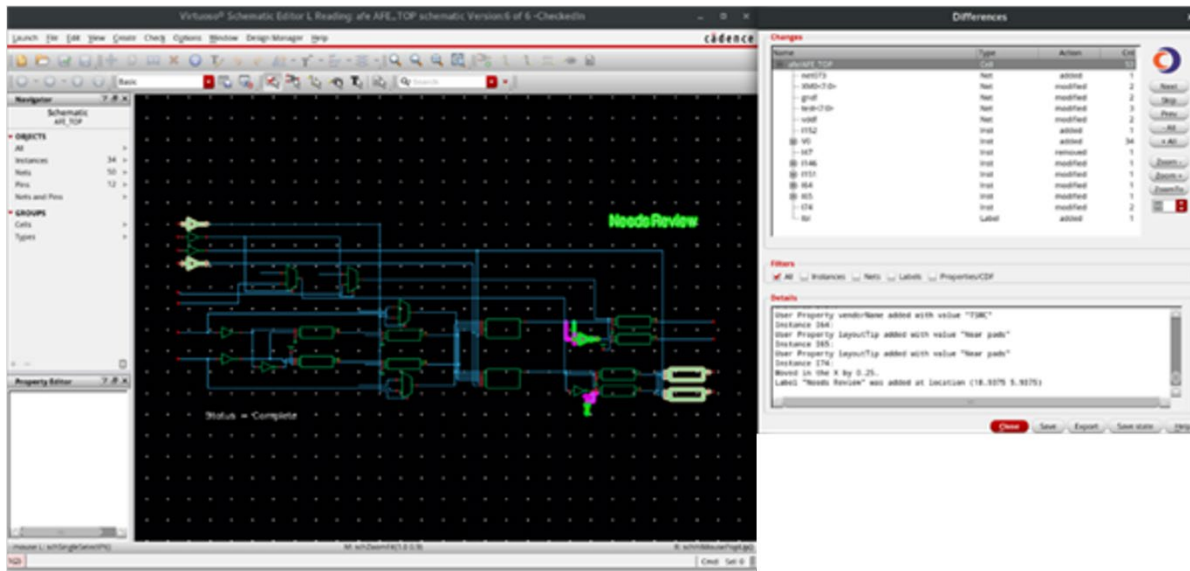


Figure 7. Schematic diffing

Layout Diff

For layout cellviews, VDD performs an XOR comparison across layers to determine geometric differences. The operation runs directly on the OpenAccess data, without requiring a GDSII export step, making it significantly faster than traditional export-and-compare workflows [figure 8].

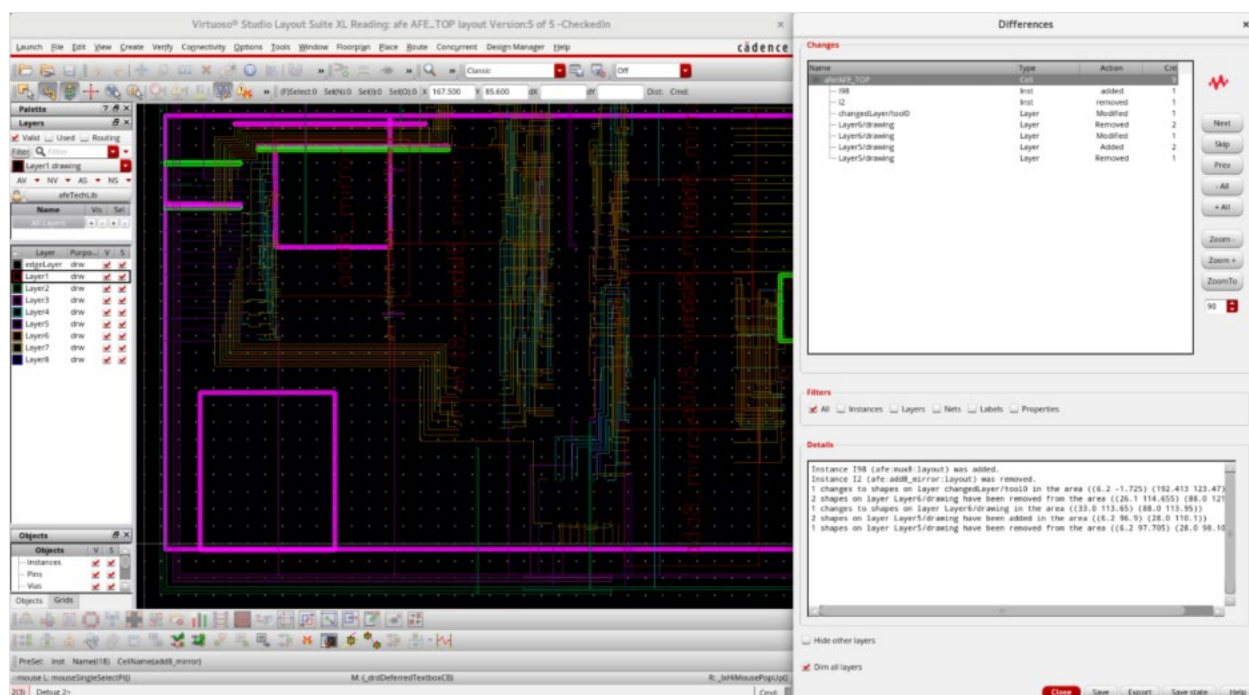


Figure 8. Layout diffing, using OpenAccess data directly, without requiring a GDSII export

Hierarchical Comparison

VDD supports hierarchical comparison across design hierarchies. The Version Compare dialog includes options to "Compare hierarchy," "Ignore cosmetic differences," "Ignore library name differences," and "Include pcell/via contents." These filters allow designers to focus on electrically significant changes while suppressing noise from cosmetic modifications or library reorganizations.

Hierarchical Operations

In complex analog and mixed-signal designs, individual cellviews do not exist in isolation. A top-level schematic references sub-blocks, which in turn reference leaf cells, forming a deep design hierarchy. SOS provides hierarchical operations that traverse this hierarchy and apply a data management operation to the entire design tree in a single step.

The Design Manager > Manage Hierarchy submenu exposes hierarchical variants of all core SOS operations: Check In, Check Out, Cancel Checkout, Tag, Snapshot, Merge, Show Status, Update, Show Labeled, List Revision Diffs, and Miscellaneous.

Hierarchy Traversal

When a hierarchical operation is invoked, SOS presents the Descend Hierarchy dialog. The designer selects the traversal mechanism, which determines how SOS discovers the objects in the hierarchy. Three mechanisms are available: traversal by view type (e.g., include all schematic,

layout, and symbol views), traversal by view name (space-separated list of specific view names), and traversal using a config view (follows the hierarchy defined by a Cadence config view). The designer also selects which libraries to include in the traversal.

After traversal, SOS presents the standard operation dialog (e.g., Tag, Check Out) pre-populated with all discovered objects. The designer can refine the selection using filters before confirming the operation.

Practical Example: Hierarchical Tagging

Consider a design engineer who has completed schematic design for an AFE (Analog Front End) block with 15 sub-cells. Rather than manually tagging each sub-cell individually, the engineer selects the top-level AFE cellview, invokes Design Manager > Manage Hierarchy > Tag, selects "schematic" as the view type, and clicks OK [figure 9]. SOS traverses the hierarchy, discovers all 15 schematic views, and presents them in the Tag dialog. The engineer selects the "design_done" tag, clicks OK, and the entire schematic hierarchy is tagged in a single operation. A layout engineer in a different timezone can then configure their RSO to pull the "design_done" versions automatically.

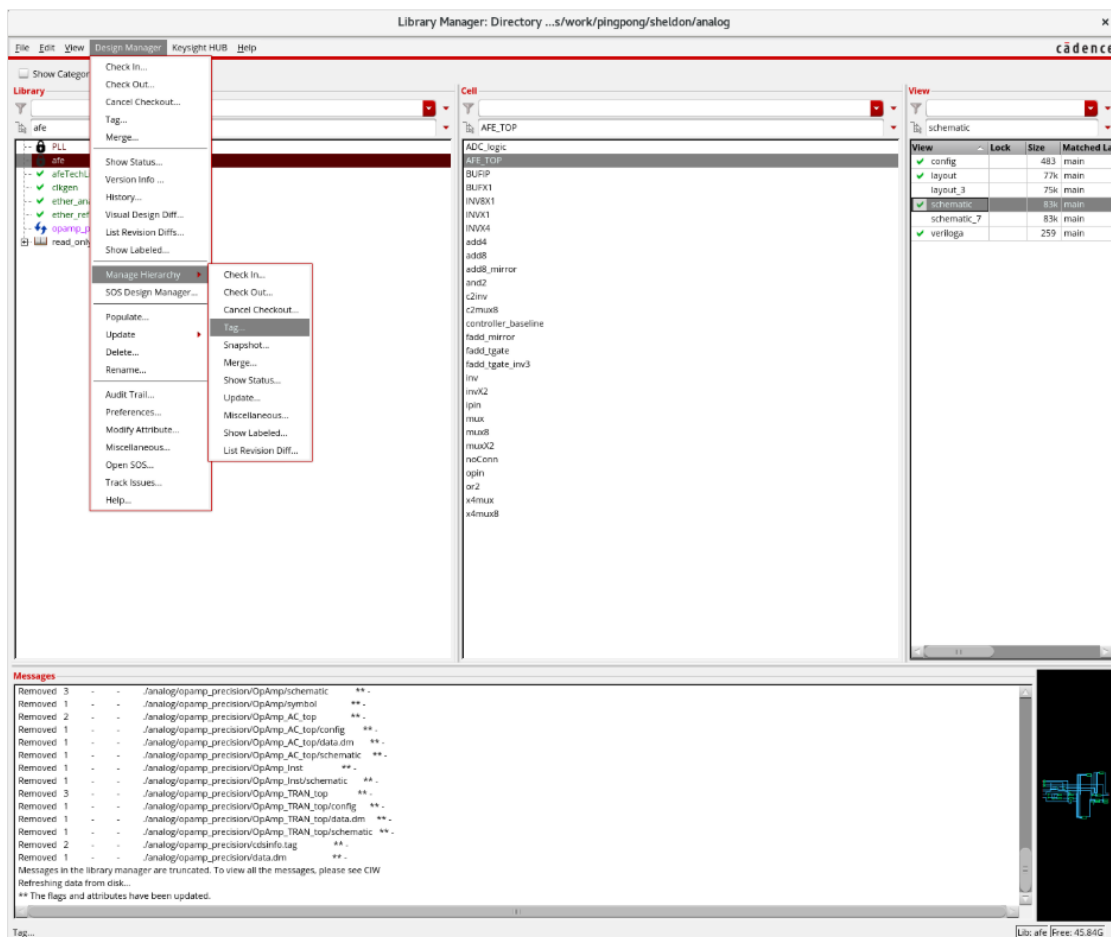


Figure 9. Using Design Manager for hierarchical tagging removes the need for manual, individual tagging

Labels, RSO, and Automated Design Handoffs

The combination of Labels and Revision Search Order (RSO) is the mechanism by which SOS automates design handoffs within the Virtuoso environment. This capability eliminates the need for email-based notifications, manual file transfers, or ad hoc coordination between design, layout, and verification engineers.

How RSO Works in Practice

Each user workarea has an associated RSO, which is an ordered list of labels. When a workarea is updated (via Design Manager > Update), SOS resolves which revision of each object to retrieve by walking the RSO from highest to lowest priority. For a given object, SOS checks each label in priority order: if a revision tagged with that label exists, it is used; otherwise, SOS moves to the next label. The built-in label "main" always resolves to the latest revision.

For example, with an RSO of [design_done, gold, main], SOS first checks whether a revision tagged "design_done" exists for each object. If so, that specific revision is placed in the workarea. If no "design_done" revision exists for a given object, SOS falls back to "gold," and if that is also absent, it retrieves the latest revision. By manipulating the RSO, designers can precisely control which version of every object appears in their workspace.

Workarea Configuration for RSO

The SOS Virtuoso integration supports advanced workarea configuration that allows different RSOs for different view types within the same workarea. For example, a verification engineer might configure their workarea to use "design_done" for schematic views while using "layout_done" for layout views. This is managed through the Change Library RSO dialog (Design Manager > Update > Change Library RSO), which provides a three-panel interface: library selection, view-class-to-RSO mapping (update rules), and label/RSO assignment.

Workarea configurations can be saved to external files or to the project itself (Design Manager > Update > Save Workarea Config / Load Workarea Config). This enables CAD managers to define standard configurations and distribute them across the team, ensuring consistent workspace setup.

Library-Level Diff Reports

Beyond cellview-level comparison, SOS provides library-level diff reports that summarize all changes across an entire library between two RSO states. These reports are accessible from Design Manager > List Revision Diffs (at the library level) or from Manage Hierarchy > List Revision Diffs (across a full design hierarchy).

The List Revision Diffs dialog allows selection of "From" and "To" RSO pairs, time-range filtering, and output format selection (table view, HTML, text, or CSV). The resulting report enumerates every object that differs between the two states, with columns for the object path, "From" revision, "To" revision, check-in author, check-in time, and change description. This capability is particularly valuable for release engineering, where a team needs to understand exactly what changed between two milestones.

Audit Trail

The Audit Trail (Design Manager > Audit Trail) generates a chronologically sorted report of all data management operations executed on project data. It provides a comprehensive, filterable history that answers questions like: who checked in this cellview, when was this tag applied, what operations were performed on this library last week?

The Audit Trail dialog supports filtering by time range, operation type (create + checkin, checkout, snapshot + tag + branch, delete + move + rename, addreference + editreference), user (CSV list), and scope (entire project or selected paths). The report can be displayed in a grouped transaction view or as individual operations.

The Virtuoso-specific Audit Trail dialog adds two capabilities beyond the standard SOS Audit Trail. The Diff button launches VDD directly from a selected transaction in the report, allowing instant visual inspection of the changes introduced by a specific check-in. The Save button exports selected rows to a CSV file for external analysis or archiving.

Customization and SKILL APIs

The SOS Virtuoso integration provides extensive customization capabilities through the SOS Preferences dialog (Design Manager > Preferences) and through SKILL programming interfaces.

SOS Preferences

The SOS Preferences dialog contains multiple tabs for configuring the integration behavior. The General tab controls SOS Design Manager enablement, library iconification, and other global settings. The Dialog box options tab configures default behaviors for Check In, Check Out, and Update dialogs. The Manage hierarchy tab sets defaults for hierarchical operations. The VDD tab configures Visual Design Diff options. The Advanced tab manages SKILL triggers and hooks. The Filter Options tab sets default filters for report tables. The RMB Customization tab allows designers to select which SOS commands appear in the right-click context menu. The View types tab configures which view types are displayed in RSO-related dialogs. The Customize cellview columns tab adds custom attribute columns to the Library Manager's View section.

Preferences can be saved to and loaded from the .sosinit.ile file, enabling consistent configuration across sessions and across team members.

Custom Attributes

SOS supports user-defined custom attributes (integer or string values) that are stored with each object revision. Custom attributes are defined in the project's sos.cfg file and can be surfaced in Virtuoso dialogs using the sosAddPropToHM SKILL function. For example, a team might define a "progress" attribute that designers set during check-in, enabling project managers to track design completion through the SOS Design Manager's column operations and reporting features.

SKILL Variables and Functions

The integration exposes several SKILL variables for controlling behavior. Check \$CLIOSOFT_DIR/scripts/cds_sosviadfil_vars.il for details.

Library Categories and Iconification

The Iconify the libraries option in the General tab of SOS Preferences enables visual categorization of libraries in the Library Manager. The default categories distinguish SOS-managed libraries, SOS reference libraries, and unmanaged libraries. Custom categories can be defined by creating a custom sos_category.lib file.

Real-World Workflow: A Multi-Engineer Design Scenario

The following scenario illustrates how the SOS Virtuoso integration supports a realistic multi-engineer design flow from schematic design through layout completion and release, using fictitious names for the intervenients of the handoffs.

Project Setup

A project lead, Penny, initializes the SOS project from the SOS GUI. The project repository stores all design data centrally, with Cache Repositories deployed at each engineering site. Penny creates project views to give analog engineers visibility into analog libraries and layout engineers visibility into physical implementation data.

Schematic Design (Engineer: Sheldon)

Sheldon, an analog design engineer, launches Virtuoso and opens the Library Manager. He right-clicks the "afe" library and selects Design Manager > Update Selection to synchronize his workarea with the latest project state. The Update dialog identifies two out-of-date cellviews, which Sheldon updates before beginning work.

Sheldon double-clicks the AFE_TOP schematic to open it. The Auto Checkout dialog appears; he confirms and begins editing. After completing his schematic changes, he saves the cellview and runs Spectre simulations to verify functionality.

Satisfied with the simulation results, Sheldon right-clicks the library and selects Design Manager > Check In. He enters a change description ("Fixed netlisting issues") and optionally links the check-in to a Jira issue ID for traceability. SOS creates a new revision and releases the lock.

Sheldon then applies a milestone tag to his entire schematic hierarchy. He selects the AFE_TOP cellview, invokes Design Manager > Manage Hierarchy > Tag, chooses the "schematic" view type, and applies the "design_done" tag to all schematic views in the hierarchy in a single operation.

Layout Design (Engineer: Amy)

Amy, a layout engineer at a remote site, opens the Audit Trail (Design Manager > Audit Trail) to review recent project activity. She sees Sheldon's check-in and tagging operations in the HTML-formatted report.

Amy opens Design Manager > Update > Change Library RSO and moves "design_done" to the top of her RSO priority list. She then updates her workarea. SOS resolves the RSO for each object: schematic views tagged "design_done" are retrieved at the specific revision Sheldon tagged, while other view types fall back to the next label in the RSO.

Amy completes the physical layout, runs DRC and LVS, and checks in her work. She applies the "layout_done" tag to her layout hierarchy using the same Manage Hierarchy > Tag workflow.

Release (Project Lead: Penny)

Penny creates a project snapshot from the SOS GUI (Project > Take Snapshot), entering "fp2" as the snapshot name. The snapshot freezes the exact combination of revisions across all objects at that moment, providing a reliable baseline that can be recreated at any point in the future for tapeout preparation or post-silicon debugging.

Why SOS for Virtuoso

The SOS Virtuoso integration delivers value across multiple dimensions that compound over time as teams grow and projects multiply.

Capability	Value Delivered
Native Virtuoso integration	Engineers never leave their design environment. All data management operations are accessible from the Library Manager menus, editor menus, and right-click context menus. The learning curve is minimal.
Visual Design Diff (VDD)	Side-by-side visual comparison of schematic and layout revisions with connectivity-aware schematic diff and XOR-based layout diff. Eliminates the need for GDSII export for comparison. Valued as a standalone tool by many customers.
Hierarchical operations	Apply check-in, check-out, tag, update, and other operations across an entire design hierarchy in a single step. Critical for complex analog and mixed-signal designs with deep hierarchies.
Labels and RSO	Automate design handoffs between functional roles (design, layout, verification) without email, file transfers, or manual coordination. Each role configures their RSO to pull exactly the revisions they need.
Links-to-Cache workspaces	Lightweight symbolic-link workspaces that consume megabytes instead of duplicating terabytes of project data per engineer. Dramatically reduces storage costs and workspace setup time.
Audit trail and traceability	Complete, chronologically sorted history of all project operations. Integrated issue tracker linking. Essential for tapeout readiness, post-silicon debugging, and regulatory compliance.
Customizability	SKILL APIs, custom attributes, configurable context menus, workarea configurations, and trigger hooks allow CAD teams to tailor the integration to their specific design methodology and process requirements.

Scaling Your Approach

SOS Core is designed as the entry point to the broader Keysight Engineering Data Management platform. As organizations grow in team size, number of sites, regulatory requirements, and data maturity, SOS Core provides a seamless upgrade path to SOS Enterprise, which extends the platform with global, multi-site collaboration with real-time synchronization; enterprise-grade governance including role-based access control, encryption, and automated approval workflows; full lifecycle traceability from schematic through verification and test; AI-ready data infrastructure with structured metadata and APIs for ML pipelines and agentic AI workflows; and IP lifecycle management with catalogs, bills of materials, and standardized reuse processes.

The upgrade preserves all existing project data, configurations, user workflows, and Virtuoso integration settings. There is no migration penalty.

Conclusion

The Keysight SOS Core integration with Cadence Virtuoso transforms Virtuoso from a standalone design environment into a fully data-managed platform. Engineers gain version control, collaboration, and traceability without leaving their primary tool. The integration's Visual Design Diff capability provides unmatched visibility into design changes. Hierarchical operations and the Labels/RSO system automate the cross-functional handoffs that are critical to analog and mixed-signal design flows. And the Links-to-Cache architecture ensures that all of this is delivered with minimal storage overhead and local-speed performance, regardless of team size or geographic distribution.

For semiconductor design teams that rely on Cadence Virtuoso, SOS Core is the most mature, deeply integrated, and widely deployed design data management solution available. Built on the same platform trusted by the world's leading semiconductor companies, it provides a reliable, scalable foundation for design data management in the AI era.

To learn more about Keysight SOS Core and the Virtuoso integration, visit: [EDA Integrations | Keysight](#)



Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.

This information is subject to change without notice. © Keysight Technologies, 2026, Published in USA, April 29, 2026, 3126-1131.EN