

# DATA

## ENOVIA DesignSync Data Manager



ENOVIA® Synchronicity® DesignSync® Data Manager is used by semiconductor companies to manage the hardware and software data in their products. Data can be managed at both the detailed file/directory level, and at a “modular” level of abstraction. As such, design data contributed by individual teams can be seamlessly integrated into higher level designs.

### Key Benefits

- Connect and manage your entire design chain with a unified DDM system.
- Significantly boost design productivity for a rapid payback and strong ROI.
- Maximize your ability to reuse existing designs and embedded software.
- Manage your design hierarchy as part of the design process.
- Utilize an intuitive built-in Submit, Integrate, Test, and Release (SITaR) workflow.
- Reduce time-to-market by increasing collaboration efficiency.
- Win first-to-market advantage.
- Manage complex data types from a variety of EDA tool vendors.
- Manage software projects using the Microsoft Visual Studio and Eclipse plug-ins.

# Product Overview

The creation of complex electronic products is not an easy proposition, and is becoming increasingly complex with the proliferation of globally dispersed teams. Since 1998, integrated circuit (IC) design teams have relied on ENOVIA® Synchronicity® DesignSync® Data Manager to help manage the hardware and software data in their products. Today, over 120 development organizations, including 13 of the top 15 semiconductor companies, take advantage of ENOVIA Synchronicity DesignSync Data Manager to boost design productivity. ENOVIA Synchronicity DesignSync Data Manager was designed specifically for DDM (Design Data Management) of complex IC design, and continues to evolve as challenges facing the semiconductor industry evolve as well.

## Local Optimization Versus Global Efficiency

Today, designing an IC often requires integrating a multitude of datasets contributed by multiple, and often geographically dispersed, design teams. As ICs become more complex, design teams are required to specialize because no one individual, or even a team of individuals, can do it all. Specialization results in the segregation of design tasks along specific lines such as analog or digital design – and then within those teams, further segregation occurs. For example, in analog design, schematic capture and physical layout are often performed by different design teams. In digital design, specialties exist in the areas of RTL design, design compilation, physical place and route, simulation and verification. ICs also include ever-increasing amounts of embedded software, which is yet another area of specialization. All components contributed by the different design teams need to come together into one chip design. This daunting task is managed by an integration team.

While individual teams are focused on the details of the design data for which they are responsible, and are thus focused at the file level, the integration team needs to be able to manipulate data sets at a higher level of abstraction. Glue logic at the top-level instantiates configurations, or releases, of lower-level modules. This sounds straightforward, but in many cases is not. This is because individual design teams are often using independent and unconnected DDM systems that require that the integration team extracts data sets for different component parts from different systems. Extremely high rates of change for individual components make the maintenance of a consistent amalgamation at the chip level very difficult. While the various Software Configuration

Management (SCM) and DDM tools in use by the individual design teams may satisfy local requirements, design data management breaks down at the integration level. As a result, manual procedures are used which lead to errors and inefficiencies. For this reason, many companies are now attempting to plot a roadmap that leads from a policy of local optimization into the realm of global design efficiency. ENOVIA Synchronicity DesignSync Data Manager is one solution that can help.

## Designing with Modules

ENOVIA Synchronicity DesignSync Data Manager enables design teams to manage data at both the detailed file/directory level, and at a “modular” level of abstraction. It is the ability to efficiently manage design data at both of these levels of abstraction which differentiates ENOVIA Synchronicity DesignSync Data Manager from other DDM systems. Design data contributed by individual teams can be seamlessly integrated into higher level designs.

At the heart of the architecture is the “module.” A module represents a single coherent and consistent collection of files and folders. While the revision history of individual files is maintained at the file level, a revision history of the context, or structure, in which the files exist is maintained at the “module” level. A module version is stored in a “manifest” which represents a “bill of materials.” The manifest for a module version stores a list of individual file versions along with the associated directory structure, and, if hierarchy is included, references to sub-modules. The revision history of a module records a complete genealogy, making it possible to resurrect the design as it existed at any previous point in time. Storage of module versions in a manifest also eliminates the need to tag individual files in order to manage and maintain consistent data configurations.

The architecture under which module versions are stored in an ENOVIA Synchronicity DesignSync Data Manager server affords many advantages:

## Change Set Processing

Because “manifests” are stored for each module version, update operations employ a “change-set” paradigm, eliminating the need to enumerate over all objects in a directory structure to determine which subset needs updating.

## High Performance

Change set processing also enables high performance, and high performance substantially increases the probability of the successful deployment of a DDM tool, which in turn encourages the use of best practices.

## Atomic Operation

Operations such as check-in are “atomic” meaning that transactions involving multiple files are not committed to the data repository unless all the individual files have been checked in successfully. Sophisticated recovery mechanisms allow an interrupted check-in to proceed where it left off, without having to start over.

## Directory Versioning

Creation, deletion or renaming of directories and files is captured in the module version history, providing a complete genealogy of changes to the design.

## Module Views

Because designers often work on a subset of project data, workspaces can be populated with specific module “views” of the design. RTL designers could populate their workspaces with the “RTL View”, containing only RTL data, whereas documentation writers could fetch only files associated with documentation.

## Distributed Data Repositories

Design data can be distributed over multiple repositories, thus maximizing local efficiencies.

## Managing a Design Hierarchy

ENOVIA Synchronicity DesignSync Data Manager enables the management of a design hierarchy of modules. In addition to file versions organized in folders, a module version may contain one or more “hierarchical references” (hrefs) to other module versions that may be stored in the same server or in another server located anywhere across the globe. Hrefs are processed when design data is fetched into a workspace. If a module version that contains an href is fetched, the href is resolved and the referenced module version is fetched as well. In this manner, a hierarchical design is assembled in the workspace. By establishing containment relationships (hrefs) to other modules in modules themselves, ENOVIA Synchronicity DesignSync Data Manager captures the design hierarchy and actively manages it throughout the evolution of the design. Thus, all of the individual design elements are managed, and interrelationships are maintained in a consistent manner, whether the entire dataset is stored in a single server, or distributed across servers around the world.

Using ENOVIA Synchronicity DesignSync Data Manager an entire design hierarchy that is distributed across multiple servers can be fetched with a single command. Without using a DDM tool that can efficiently manage a hierarchy of constituent components in a complex IC, many companies are struggling with capturing the “hierarchy” of a design even after it has been completed. If the components come from disparate and unconnected DDM systems, it can be very difficult to determine which components even ended up in a given design release.

Because ENOVIA Synchronicity DesignSync Data Manager allows users to manage a design hierarchy, a company gains advantages including:

### Task-based Workspace Creation

Workspaces can be tailored to particular tasks such as verification of an immutably-released design hierarchy, a hierarchy containing work in process, or a mixture of both.

### Single Command Builds or Updates an Entire Hierarchy

There is no need to fetch individual components of a design; ENOVIA Synchronicity DesignSync Data Manager will automatically fetch an entire hierarchy.

### Overlapping Module Data

A module hierarchy can be created to allow data from different modules to be fetched into the same directory in a workspace.

### Filters

Sophisticated filtering can be applied to files, directories or hrefs to provide fine-grained control over what data is fetched or operated upon. Filters allow designers to construct workspaces comprised of consistent subsets of data, eliminating the need to fetch large amounts of data unnecessary to tasks at hand.

### File-based designs can also be included in a design hierarchy

ENOVIA Synchronicity DesignSync Data Manager still includes the file-based storage mechanisms which existed prior to the introduction of module-based storage. Design configurations based on tags, or Hierarchical Configuration Manager (HCM) releases, can also be referenced into a module. (HCM is the predecessor to “modules,” but is file-based.) This provides a smooth transition to a module-based integration methodology because there is no requirement that ALL components be designed using module-based storage.

Design teams with well-established file-based methodologies can continue to use such, but their contributions can be easily integrated into higher level systems.

### Where Used

It is often important to be able to answer the question “where has this block of design data been used in other design blocks?” A common scenario is when a bug has been discovered in a design block which has already been included in delivered product. Determining which versions of which products contain the faulty block then becomes vitally important. If design hierarchies are managed using ENOVIA Synchronicity DesignSync Data Manager hrefs, the “whereused” command provides the answers. Given a module version, HCM release, or tagged configuration of files, the “whereused” command will trace hrefs upwards, reporting all design hierarchies in which the block has been included. Because design reuse is common, a single component might find its way into many other designs or finished products. Countless hours can be saved determining where a given design block has been used by leveraging this capability to extract the answers automatically.

### Submit, Integrate, Test, and Release – SITaR

ENOVIA Synchronicity DesignSync Data Manager provides an intuitive built-in workflow called SITaR (Submit, Integrate, Test, and Release). SITaR consists of a set of commands that leverage the power of module-based design in an environment consisting of multiple design modules aggregated together by an integrator into a higher level system. Though individual design blocks (modules) may be contributed by different teams, design at the block level cannot occur in a vacuum. Simulations must include interactions with the other blocks in the design.

SITaR is based on the notion that there are two fundamental “roles” in play in such a design: a “designer” is someone who is contributing at the block level. An “integrator” is responsible for integrating blocks together into a top-level design, testing the system, and releasing the stable “baseline” (a system level configuration of blocks) from which all subsequent block level development occurs.

Working within such a flow, a designer would fetch the current stable baseline into a workspace. The block the designer is working on would then be put in an “edit” mode, and design activities proceed. All simulation takes place in the context of the baseline consisting of all the other blocks. The key here is that design work and simulation is NOT done with work-in-process configurations of other blocks, but only in the context of the stable baseline.

When work is complete, the designer “submits” his module for possible integration into a newer baseline. The integrator monitors the submission queue, which could contain submissions for multiple blocks, and is able to build a workspace containing any mixture of submitted blocks. This is the “Integrate” step in SITaR. Regression tests are performed against the newly integrated set of blocks (the “Test” step in SITaR), and if deemed stable by the integrator, can be released as a new stable baseline (the “Release” step in SITaR). Designer workspaces could then be updated, fetching the baseline for all blocks for which editing activity is not occurring.

Thus, ALL design work at the block level is performed in the context of a stable baseline of the rest of the blocks in the design. All this activity is performed using simple and intuitive commands such as “sitr submit,” or “sitr integrate,” alleviating the need to educate the contributing teams in the use of the more fundamental module design command set, or with complicated handoff and tagging schemes. SITaR commands wrap the underlying module commands such that the power of the module-based DDM architecture is leveraged in the context of this well-defined use model.

### EDA Data Awareness and Third-party Tool Integrations

Because of the tight focus on integrated circuit design, ENOVIA Synchronicity DesignSync Data Manager has been the industry leader in integrating DDM capabilities into the highly specialized design environments in which design work is performed from leading vendors such as Cadence® and Synopsys. EDA integrations are comprised of two components: An “EDA data recognition” capability, and a plug-in for the EDA tools graphic environment.

EDA data recognition enables ENOVIA Synchronicity DesignSync Data Manager to manage complex data types, such as the collections of files and directories which might represent an object such as a schematic diagram, as “atomic” objects.

For example, the end- user would checkout a version of a schematic diagram, causing ENOVIA Synchronicity DesignSync Data Manager to fetch the appropriate co-managed collection of files and directories which define the schematic on disk.

Integrations for EDA tools provided by Cadence and Synopsys are available as add-ons to ENOVIA Synchronicity DesignSync Data Manager. For other EDA tools, ENOVIA Synchronicity DesignSync Data Manager supports a Custom Types System (CTS) as an add-on product. It provides an Application Programming Interface (API) for creation of custom complex data types. Using the CTS, one can enable EDA data awareness for any EDA tool, whether developed in house, or commercially available.

### **Sophisticated Tools for Software and RTL Design**

Sophisticated tools facilitate workflows based on ASCII files such as RTL or software designs. A “Changed Object Browser” makes it easy to locate and operate on files in a workspace which need to be updated with a newer version, have been modified in the workspace and need to be checked in, have been modified and need to be merged with a newer version, or have been modified and merged with a newer version but with resulting conflicts.

Two- and three-way graphical “diff” viewers make it easy to see what the differences are between two files, or what has changed between different versions of a file. The three-way viewer is invoked when different versions are derived from a common ancestor, providing an intuitive interface for performing conflict resolution after merge operations.

### **A Unified DDM System is a Major Competitive Advantage**

The majority of data management problems associated with integrating large designs and the associated software can be eliminated if all the data is managed in a single unified DDM system, which could exist at a single design center, or could be distributed around the world. ENOVIA Synchronicity DesignSync Data Manager is the DDM system of choice to make such a vision a reality. In 13 of the top 15 global semiconductor companies and in hundreds of other organizations, ENOVIA Synchronicity DesignSync Data Manager is the standard for management of complex EDA data created by hardware design tools from companies such as Cadence, Synopsys and Mentor Graphics. But, ENOVIA Synchronicity DesignSync Data Manager is also uniquely suited for the management of RTL Verilog or VHDL design, cell library development, or even documentation development. Plug-ins for the Microsoft Visual Studio and Eclipse IDEs (Integrated Development Environment) are included with ENOVIA Synchronicity DesignSync Data Manager for use by software designers.

ENOVIA Synchronicity DesignSync Data Manager enables individual design teams to independently release intellectual property (IP) modules, while the integration of multiple modules can be managed at a higher level of abstraction. A single command can fetch an entire design hierarchy, and another single command can create an immutable release of the same hierarchy. Inefficient and error prone manual integration procedures can be completely eliminated. Imagine what that would mean in practice at your company!

# Product Highlights

## Client/Server Architecture

The Client/Server architecture allows for design work to proceed without connection to the server. Communication occurs for data management operations or status reporting only. The architecture is uniquely suited to support geographically dispersed design teams. Servers may be globally distributed, and accessible from client applications anywhere.

## Multisite Version Control

A “single source of the truth” is maintained, and made available to designers regardless of their physical location.

## Distributed Data Storage

Data repositories (SyncServers) can be hosted at any design site for maximum local efficiencies. Data from multiple servers can be automatically aggregated in a workspace.

## Transfer Diffs

A transfer mode allows only “diffs” (i.e. file deltas) to be transferred during check-in and checkout operations for both ASCII and binary data files. This capability provides maximum benefit in environments in which geographically dispersed design teams collaborate over high-latency networks. File transfer options are configurable between client and server combinations.

## Data Replication

Sophisticated caching mechanisms support data replication and minimize disk space usage by creating workspaces which share read-only data files which are stored in cache directories. Static and dynamic caching mechanisms are supported.

If a workspace is constructed using a static cache, updates are controlled by the user. If a dynamic cache is used, updates occur automatically, so the workspace is always up to date.

Caches make efficient use of disk space, because workspaces consist of links to read-only copies of shared files. If the cache directory is located on the same file system partition as the workspace, UNIX hard links are used to share the file copy. If the cache is on a different mounted file system, soft links are used. Physical copies are only fetched into a workspace for edit operations. The mirroring system can be configured to automatically generate mirrors as data is released and populate the data into the local shared cache so data is replicated where it is needed ahead of time.

## Module Linkages via Mcaches

The Mcache is an efficient data sharing mechanism that enables copies of modules to be shared by local teams. This greatly reduces file storage requirements as well as providing a tremendous savings in terms of development time over fetching individual copies. The Mcache provides version context directly in the user’s local workspace, making it easier and quicker to determine that one is working with the correct module versions.

## Foreign Configuration Management Modules

In many organizations, circumstances mandate that design data is managed in a variety of systems. ENOVIA Synchronicity DesignSync Data Manager provides the ability to integrate data from other Configuration Management systems and allows developers to easily retrieve a complete design hierarchy across all the systems managing data for project-wide functions such as system test or tape-out.

## Security

Whether transferring data internally or externally, security is ensured using commercial grade 128 bit SSL encryption.

## Internet Based Transfer Protocols

Data transactions use standard internet protocols and work seamlessly with existing firewalls.

## Audit Trails

Detailed revision control activity is captured in a database which may be queried using a standard Web browser.

## Sophisticated Access Controls

Protection of valuable design data is ensured by setting configurable access controls. Data access is controlled in the server, and does not rely on UNIX permissions. Access to data can be controlled based on a user’s identity, the data the user wishes to access, and the command the user wishes to run to operate on the data. An intuitive graphical user interface is provided for definition and administration of access controls.

## Sophisticated Workspace Management

Multiple methods are provided for the creation and maintenance of designer workspaces supporting differing work styles. For example, workspaces can be configured to incorporate changes made by others either on demand, or automatically.



## Configurable Use Models

Both the “locking” model and the “non-locking” model are supported to suit either design team or individual preferences. An intuitive built-in SITaR workflow facilitates design collaboration, and increases quality by ensuring that all design work is performed in the context of a stable project, or system, baseline.

## Module Centric View in DesignSync User Interface

The DesignSync interface includes both a directory-centric and a module-centric workspace view. The module-centric workspace view makes it easy to determine which data is associated with which module, especially when data from multiple modules is overlapped in the same workspace directory structure.

## Tag Module Hierarchy

The tag command supports the application of a tag to all the module versions included in a specific module hierarchy. Examination of tags associated with lower level modules provides valuable “where used” information. For example, ALU module version 1.137 could be tagged CHIP1\_\_R3, indicating that version 1.137 of the ALU module is included in the R3 release of CHIP1.

## Built-in SITaR Workflow

A standard workflow in which all design work proceeds in the context of a stable system baseline is included. SITaR facilitates project setup and collaboration by providing a simplified and intuitive command set for use by both designers and integrators. SITaR enables design teams to leverage the power of modular design, while minimizing the learning curve. This is because SITaR wraps underlying DM commands in the context of a well-defined work flow.

## Where Used Capability

A design hierarchy can be traced from the bottom up, providing answers to the question: “where has this design block been used in other design blocks or products?”

## Version History Reporting

Brief or detailed reports of the version history of an entire module, or an individual design object, provide a complete genealogical record.

## Vault Browser

A sophisticated vault browser provides a graphical display of the history of a managed object.

## Annotations

An ASCII file can be graphically annotated with information about who last changed each line the file, and when.

## Diff and Merge Tools

Both graphical and command line diff and merge tools are provided, including TkDiff.

## Comparison Utilities

Sophisticated utilities allow for the comparison of module versions, releases, hierarchies and workspaces. A workspace can be compared with a known configuration, or even with another workspace. Comparison of file content makes use of checksums so that files with identical content, but stored as different versions, are reported as being identical.

## Graphical User Interface

The ENOVIA Synchronicity DesignSync Data Manager GUI allows for the navigation and manipulation of data at both the detailed file level and at the more abstract module level. Comparison utilities, diff tools and a command line interface are included in the GUI.

## Command Line Interfaces

Two command line interfaces are provided. The “dssc” shell runs ENOVIA Synchronicity DesignSync Data Manager commands. The “stcl” shell is a Tcl interpreter, into which the ENOVIA Synchronicity DesignSync Data Manager commands have been linked, providing programmatic capabilities and access to other utilities.

## Extensible Architecture

The command set can be easily extended by creating aliases or autoloaded Tcl procedures.

## Client-Side Triggers

You can easily introduce process automation to increase efficiency and decrease errors by creating Tcl procedures which are registered to intercept operations and perform other operations. For example, if a layout object is checked in, a Design Rule Check procedure could be automatically run, and if clean, the check-in operation is allowed to proceed.

## C-API

ENOVIA Synchronicity DesignSync Data Manager can be easily integrated with other tools using a fully documented C-API.

### **Plug-in for Source Code Control for Software Components**

ENOVIA Synchronicity DesignSync Data Manager includes a plugin for the Microsoft Visual Studio IDE (Integrated Development Environment.)

### **Integrations with EDA Design Tools**

Integrations for EDA tools provided by Cadence® and Synopsys are available as add-ons to ENOVIA Synchronicity DesignSync Data Manager. For detailed information, please see product information for ENOVIA® Synchronicity® for DFII (Cadence integration), ENOVIA® Synchronicity® for Milkyway (Synopsys Milkyway integration), and ENOVIA® Synchronicity® for Synopsys CD (Synopsys Custom Designer integration).

### **Integrations with Software Development Environments**

Integrations for the popular Eclipse and Microsoft Visual Studio SW development environments are provided as part of ENOVIA DesignSync Data Manager. These integrations allow developers to quickly and easily interact with ENOVIA DesignSync® to manage design data without leaving their editing environment.

### **EDA Tool integration API**

ENOVIA Synchronicity for CTS (Custom Types System) provides an API for creation of custom complex data types. ENOVIA Synchronicity for CTS is used to enable EDA data awareness for arbitrary data types in ENOVIA Synchronicity DesignSync Data Manager. Using ENOVIA Synchronicity for CTS, one can enable EDA data awareness for any EDA tool, whether developed in house, or commercially available.

### **The Role of ENOVIA V6 and PLM 2.0**

ENOVIA Synchronicity DesignSync Data Manager supports PLM 2.0, product lifecycle management online for everyone, and the ENOVIA V6 values, which are:

- Global collaboration innovation
- Single PLM platform for intellectual property (IP) management
- Online creation and collaboration
- Ready to use PLM business processes
- Lower cost of ownership.





## Delivering Best-in-Class Products



Virtual Product



Information Intelligence



3D Design



Virtual Planet



Realistic Simulation



Dashboard Intelligence



Digital Manufacturing



Social Innovation



Collaborative Innovation



3D Communication

---

Dassault Systèmes, the **3DEXPERIENCE** Company, provides business and people with virtual universes to imagine sustainable innovations. Its world-leading solutions transform the way products are designed, produced, and supported. Dassault Systèmes' collaborative solutions foster social innovation, expanding possibilities for the virtual world to improve the real world. The group brings value to over 150,000 customers of all sizes, in all industries, in more than 80 countries. For more information, visit [www.3ds.com](http://www.3ds.com).

---

### Europe/Middle East/Africa

Dassault Systèmes  
10, rue Marcel Dassault  
CS 40501  
78946 Vélizy-Villacoublay Cedex  
France

### Asia-Pacific

Dassault Systèmes  
Pier City Shibaura Bldg 10F  
3-18-1 Kaigan, Minato-Ku  
Tokyo 108-002  
Japan

### Americas

Dassault Systèmes  
175 Wyman Street  
Waltham, Massachusetts  
02451-1223  
USA

Visit us at  
**3DS.COM/ENOVIA**

---

