cadence™

Allegro® X User Guide: Getting Started with Symphony Team Design

Product Version 23.1
September 2023

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

in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

Introduction

This document introduces Allegro® PCB Symphony Team Design as a new product option, which is a concurrent design environment where multiple users can connect to a common database for collaborative design activities. With this option, you need not create separate databases or partitions for each user that requires merging of their work into a master database.

This user guide describes features and functionality for the following layout editors:

- Allegro PCB Designer
- Allegro Venture PCB Designer
- Allegro Enterprise PCB Designer (Symphony Team Design Option included in bundle)
- Allegro Physical Viewer Plus
- Allegro Package Designer+
- SiP Layout XL
- Allegro Sigrity SI

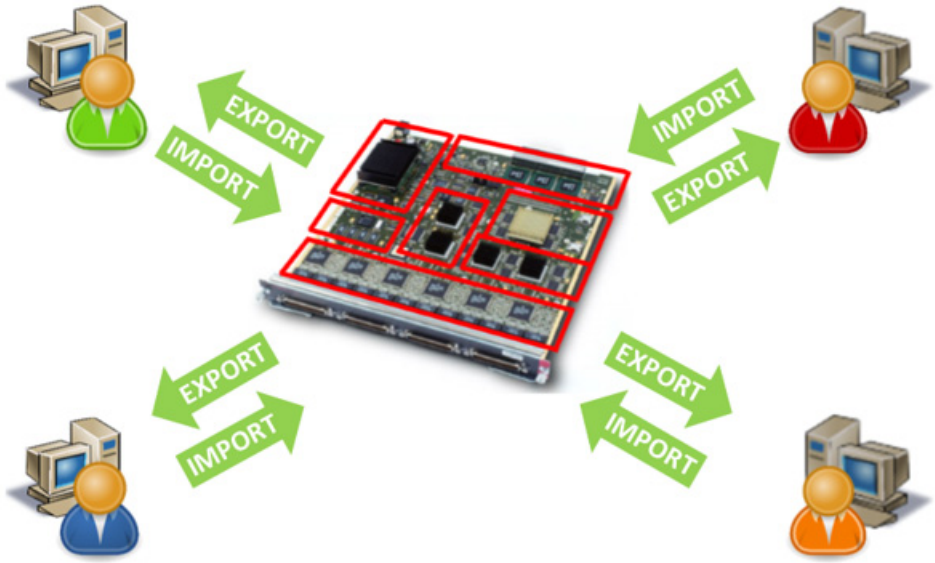
Note: The information provided in this document is based on Cadence® Allegro® release 23.1.

Symphony Team Design Overview

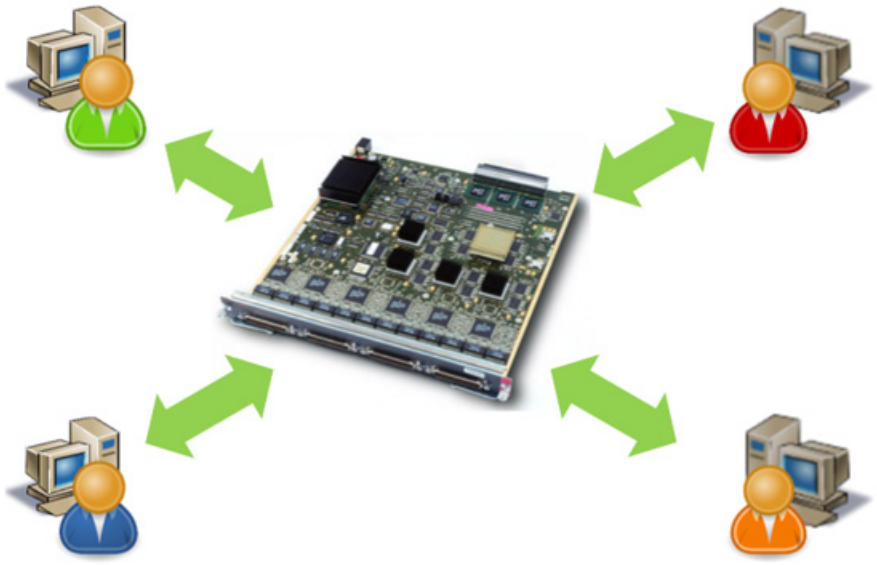
Concurrent team design (PCB Team Design Option) has been part of Allegro® PCB Editor for some time. This solution requires the partitioning of the master design into separate sections so that each team member can perform design activities. These sections are exported from the master database as partition databases so that each team member can

work independently on their assigned section. After work is completed, the individual design partitions are imported into the master database to get a clear picture of the complete design.

Concurrent team design is a productive way of allowing multiple designers to work on one design, but with each team member working on a separate partition. It requires some disruption in the design activities for an import process.



In the 17.4 release, a new concurrent design solution, *Allegro® PCB Symphony Team Design* option has been introduced. The *Allegro Symphony Team Design* option allows users to connect to a common database to perform collaborative design activities such that, each team member sees the design updates in real time. Using this common database approach, there is no need to generate or import design partitions to see the other team members design work. Whether a formal project team is formed or at a moment's notice, designers can simply share their current design and invite other designers to join for assistance.



The architecture of this concurrent solution is summarized in figure 2-1 that shows a server process controlling a PCB board or a package file and layout editor clients connecting to it using TCP-Sockets.

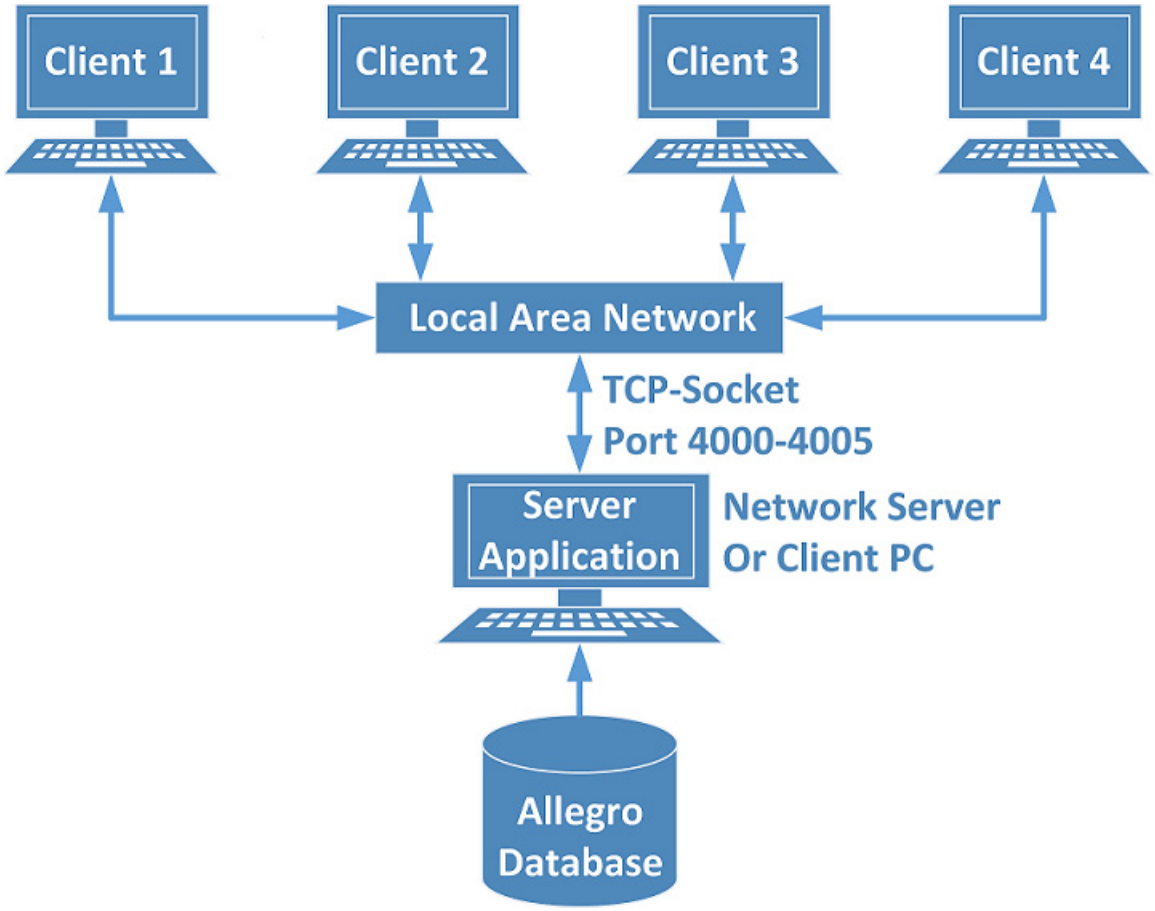


Figure 2-1 Symphony Team Design Architecture

In the figure2-1, four clients are connected to a Symphony server application that has the master database open. Each client pulls the database from the server in order to work on the design and dispatches any changes back to the server. The server then integrates the changes in the master database and dispatches to other clients. The Symphony server is a new process that has minimum user interface and only some of the layout editor database libraries to be able to handle integrating design changes.

System Requirements

Symphony server application is actually a layout editor with the main display replaced with a minimal graphics interface for server controls, and the ability to open and save databases.

Therefore, for Symphony server application to run, you need to have working installation for Cadence layout editor.

- No special hardware configuration or requirements beyond the standard layout editor. For more information, refer to the latest *Allegro Platform System Requirements* documentation.
- The server application can run on a virtual machine with or without a graphical interface.
- OpenGL is recommended on the client-side to properly see color-coded lock displays.
- Clients must have access to the *Symphony Team Design* product option license in order to connect a database being served by the Symphony server.

Terminology

Symphony Server	Symphony Server Application (`muserver.exe`)
Symphony Server Manager	Symphony Server Manager Application (`muservermgr.exe`)
TCP-Port	Transmission Control Protocol Port Address
TCP-Socket	Transmission Control Protocol Socket is used to send and receive data

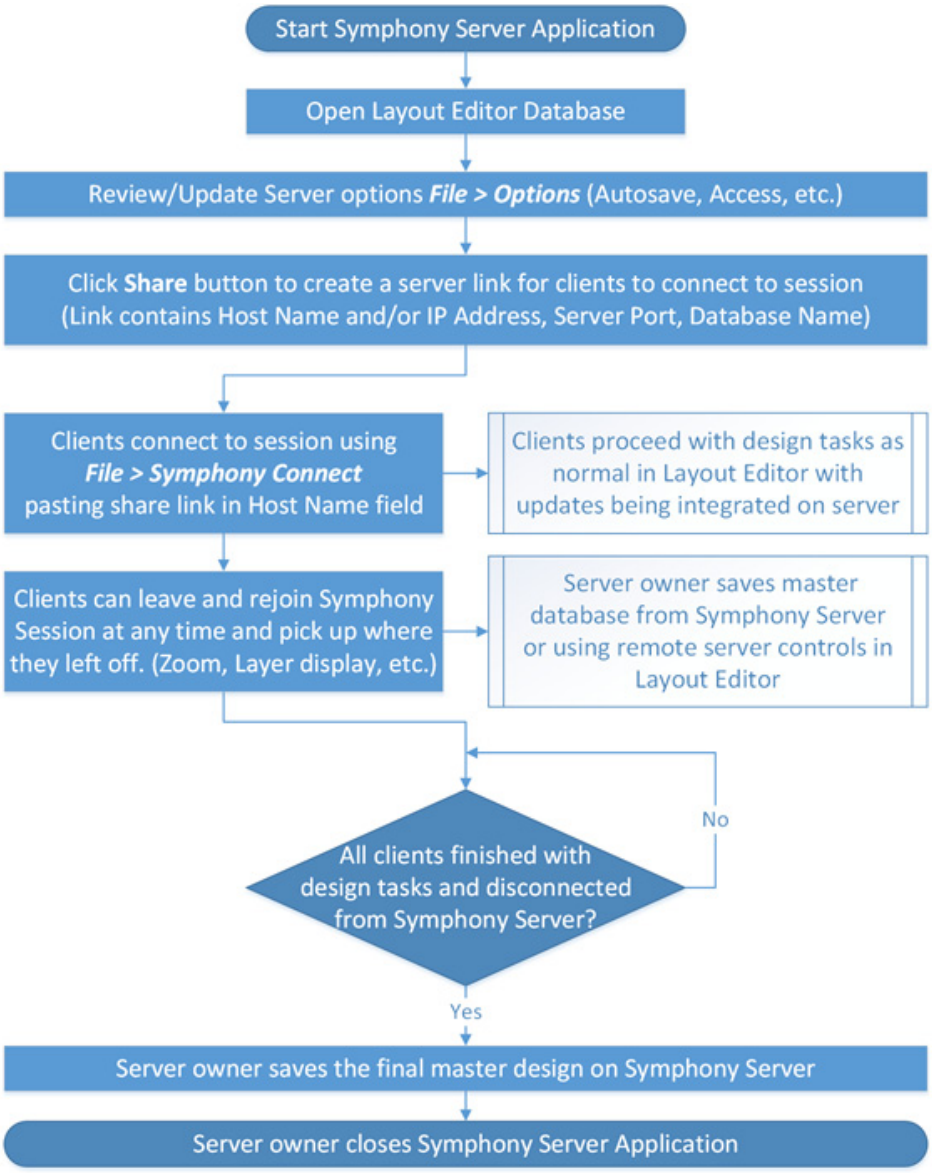
Basic Operation and Flows

Depending on whether you are connecting to a Symphony server application, or you are making your database available to other users as server, there are six different work- flows that can be used in the Symphony team design environment.

- Network Server-Side Team Design Flow on page 13
- Informal First-Client Driven Team Design Flow on page 14
- Network Client-Side Team Design Flow on page 15
- Network Server Manager Team Design Flow on page 16
- Client-Side Pause Session Flow on page 17
- Client-Side Constraint Edit Mode Flow on page 18

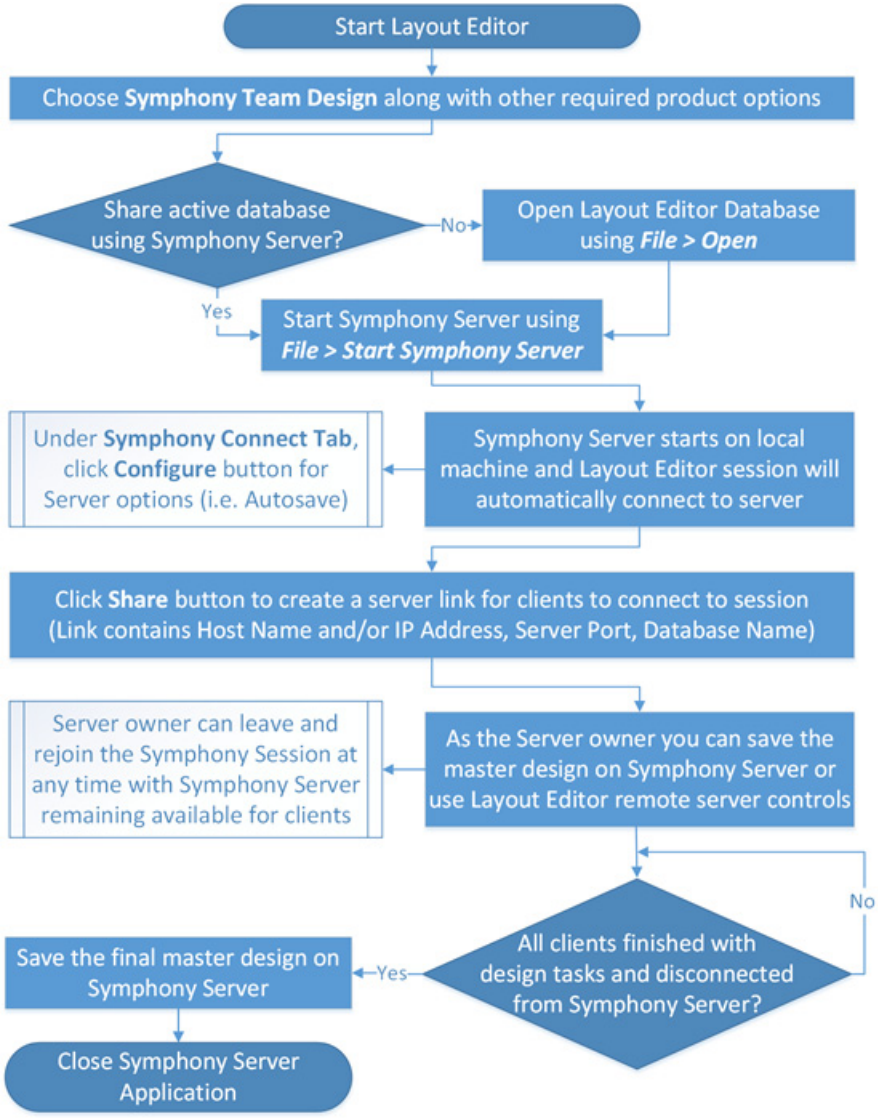
Network Server-Side Team Design Flow

Starts database server as a standalone separate application to which, clients can connect.



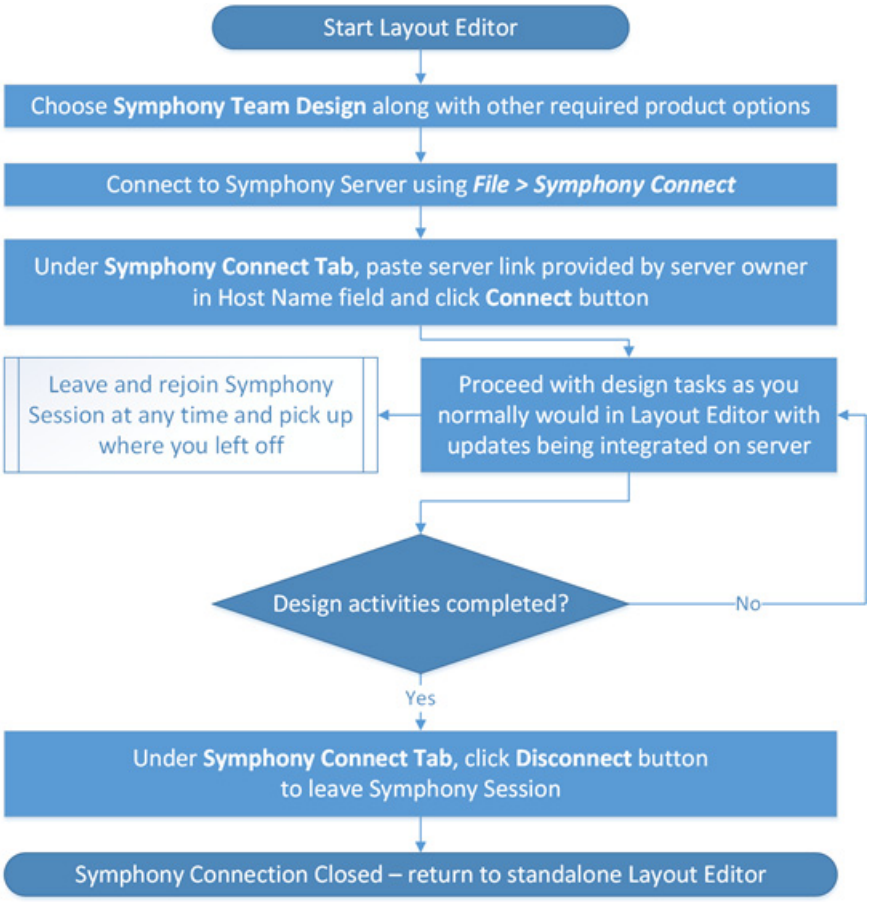
Informal First-Client Driven Team Design Flow

When the master database is on your system, and you need to make the database available for clients and work on the database.



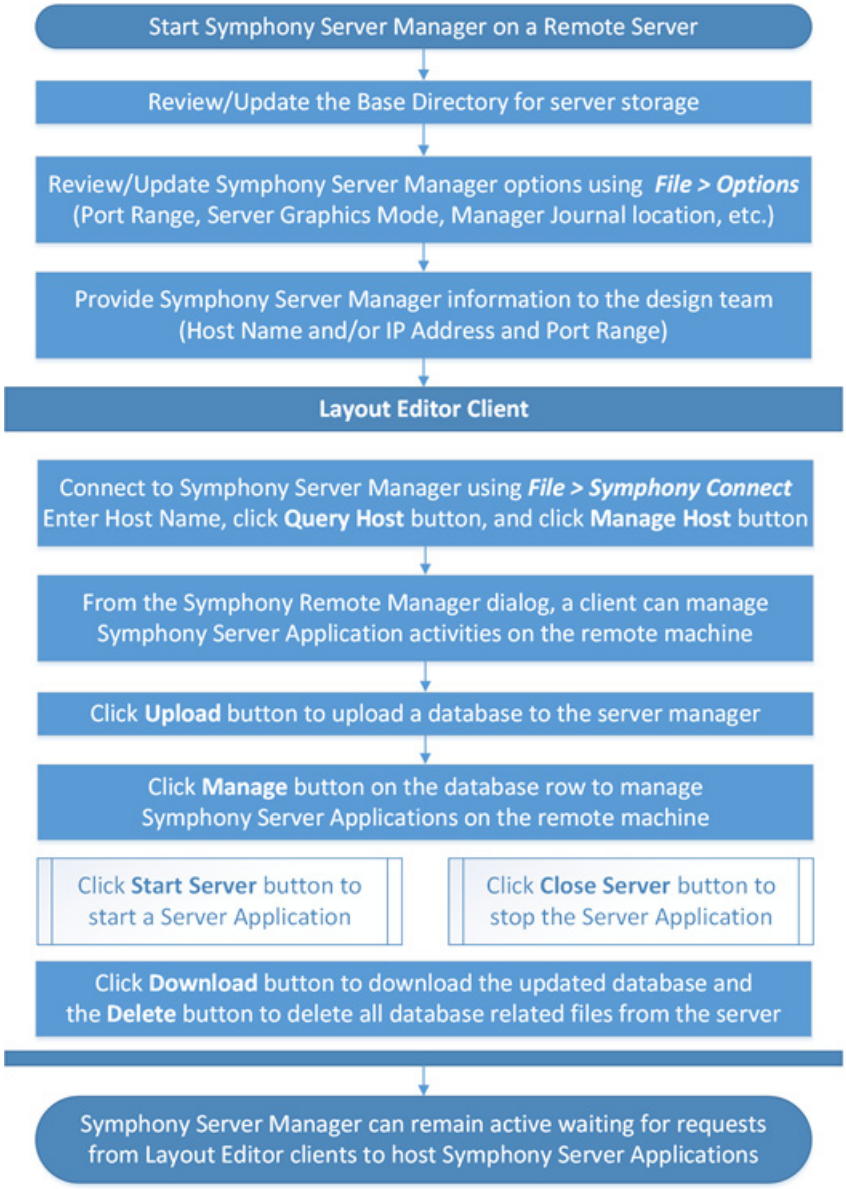
Network Client-Side Team Design Flow

Use this flow when you are a client that needs to work on a server database.



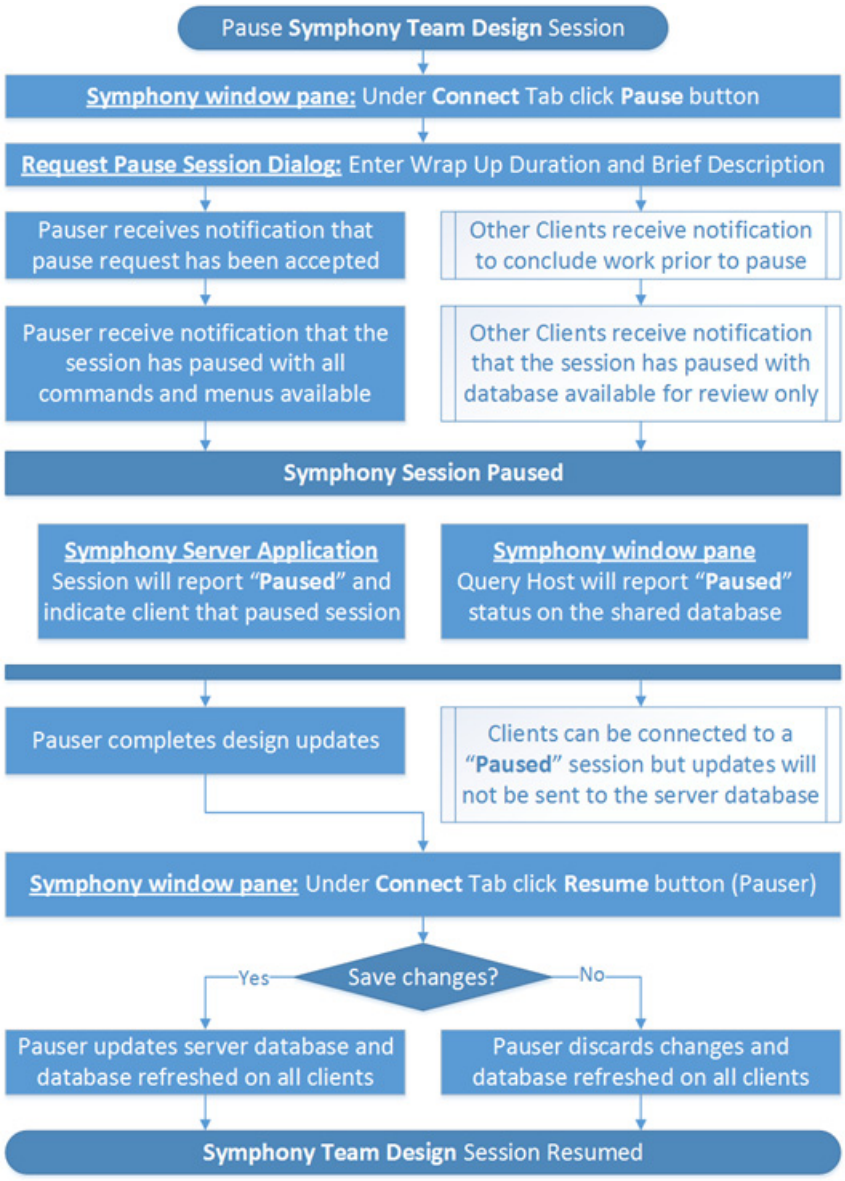
Network Server Manager Team Design Flow

Use this flow when you are a client that needs to utilize a server manager.



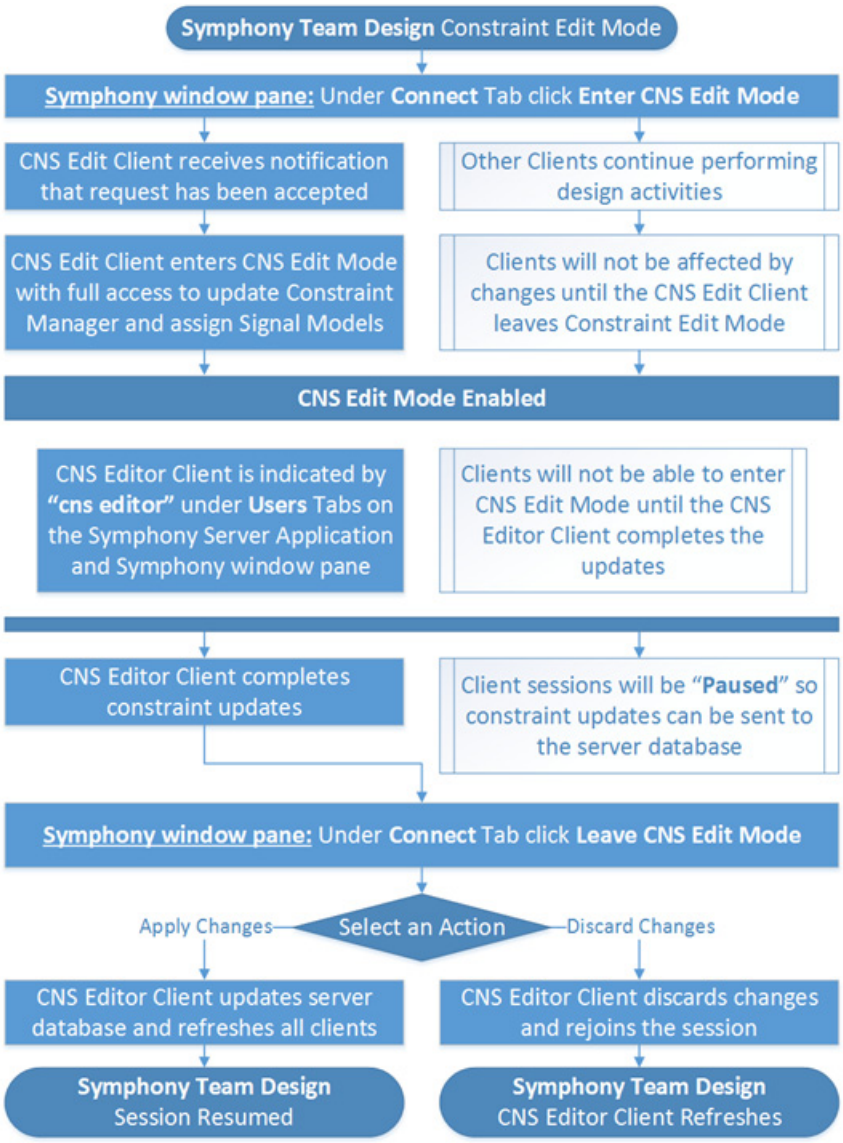
Client-Side Pause Session Flow

Use this flow when you are a client that needs exclusive access to the server database.



Client-Side Constraint Edit Mode Flow

Use this flow when you are a client that needs to update design constraints on the server database.



Getting Started with Symphony Team Design

Modes of Operation

Symphony server application can operate in following modes:

Network Server-Side Team Design Mode

Design lead starts the server application from a network server machine and all the users join and communicate with network server machine. Server manager is available to manage server applications on a network server machine.

Note: For more information, see Network Server-Side Team Design Mode on page 19.

Informal First-Client Driven Team Design Mode

Design owner starts the standalone Symphony server application from within a layout editor to share the open database from the local system of the database owner. Once started the owner's session automatically connects to the Symphony server application. All others join and communicate with owner's machine.

Note: For more information, see Informal First-Client Driven Team Design Mode on page 35.

In both the modes, the master database is controlled by the server and the owner is responsible to open and save the database through the remote controls available in their layout editor session or from the Symphony server user interface.

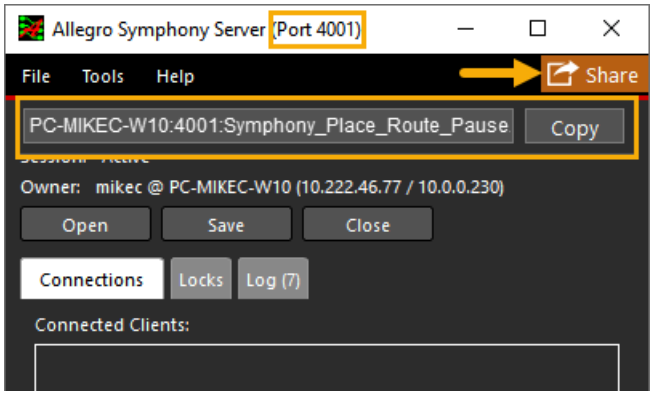
Note: Auto-save can be configured through the Symphony server but is not required on the client-side to save the database, as all changes are integrated into the master database.

Network Server-Side Team Design Mode

A separate Symphony server application (`muserver.exe`) is used to open a database so that multiple clients can connect to the common database and perform design activities in a

concurrent environment. The server process has a minimum user interface that is used to open and save a board or package file, setup some settings, and monitor activities.

The following figure shows the main user interface that has the current open database; three convenient buttons to *Open*, *Close*, and *Save* the design; and three additional tabs to monitor who is connected to the database, what objects are locked by team members, and a general log of server activities.

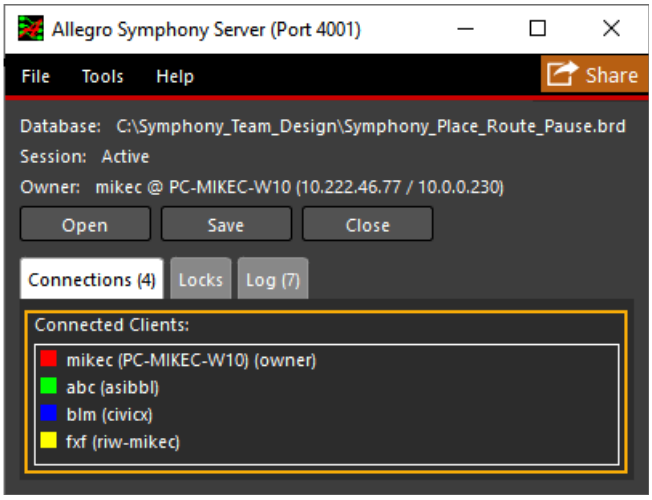


Click *Open* to browse and open a database. The database is shared on port 4001 of the machine where Symphony server is running. Clients can now connect to the database.

Click *Share* to generate a server link that can be sent to team members so they can connect to the database being shared by the server.

As clients connect to the database their login IDs starts displaying in the *List of Connections*.

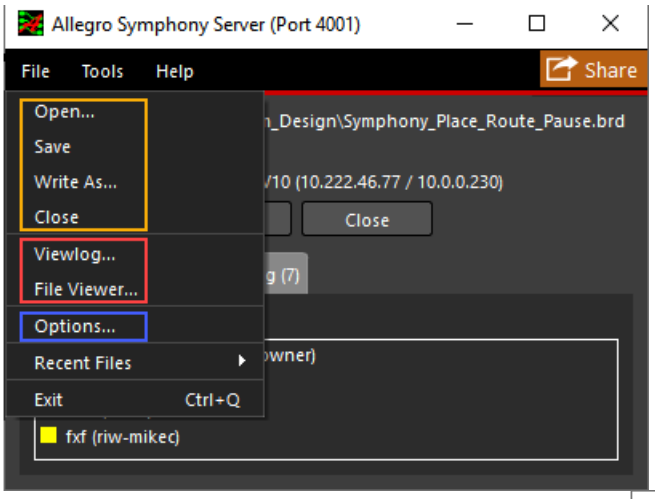
Note: Although clients can connect from different platforms, it is recommended to have the same Cadence software patch level on clients and server.

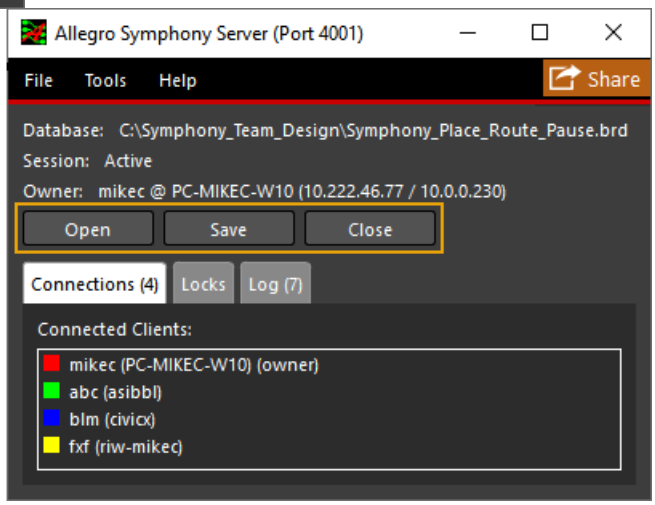


Each client connection is listed by user name, machine name, unique ID, and color for easy identification. Color coding is visible on the canvas when you create multi-user object locks.

The *File* menu contains standard database operations (*Open*, *Close*, and *Save*) while *Write As* provides the ability to save the database to a new name without changing the name of the active database currently being served to layout editor clients.

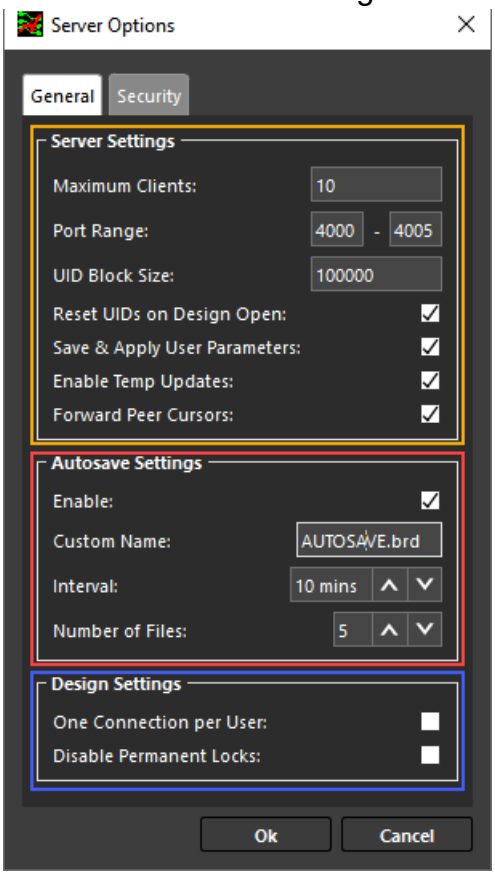
Note: *Close* disconnects all clients with a prompt on the server side to save the database prior to exiting the Symphony server application.





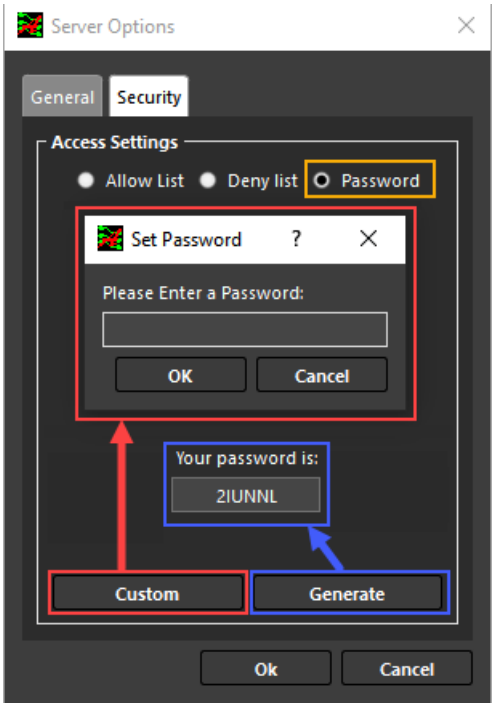
Customizing Server Settings

Choose *File – Options* to customize server settings.



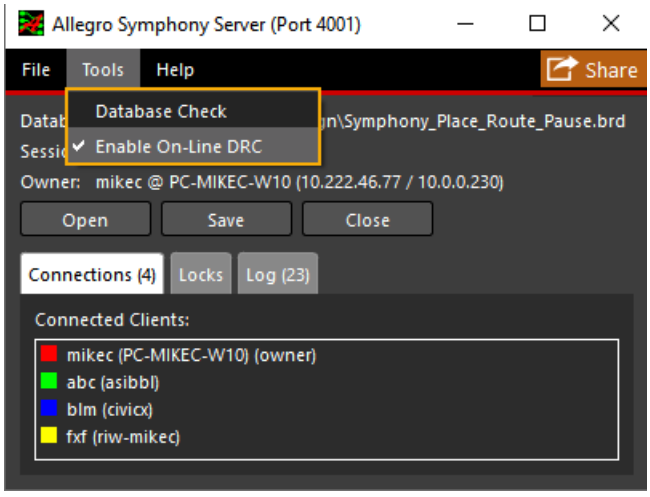
Auto-save options and design settings can also be modified in the *General* tab of the *Server Options* dialog box. For more information, see `muserver` command in the *Allegro PCB and Package Physical Layout Command Reference* guide.

The *Security* tab has options to allow or deny specific users from accessing the database. The password can either be auto-generated or defined manually. It is attached to the database to restrict the access.



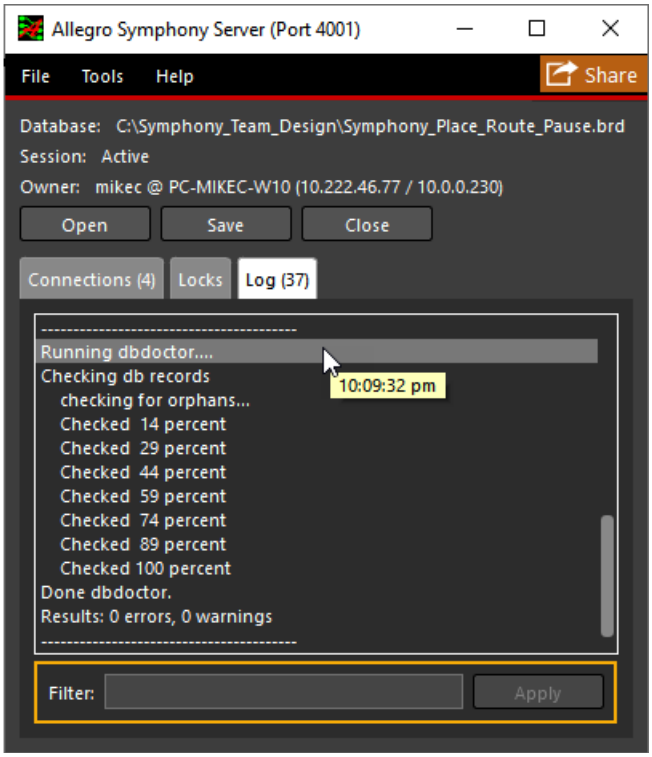
For more information, see `muserver` command in the *Allegro PCB and Package Physical Layout Command Reference* guide.

The *Tools* menu provides quick access to some basic database checks on the open database and display the results in the *Log* tab and disable *On-Line DRC* on the server database.



The database file `dbdoctor.log` can also be viewed by selecting *File – Viewlog*.

To view the time for any activity, hover over a log entry. The time stamp is available in the data tip.



To refine the locking and log details for any user, you can set *Filter* field available at the bottom of the server user interface.

For more information, see `muserver` command in the *Allegro PCB and Package Physical Layout Command Reference* guide.

Starting Symphony Server in No Graphics Mode

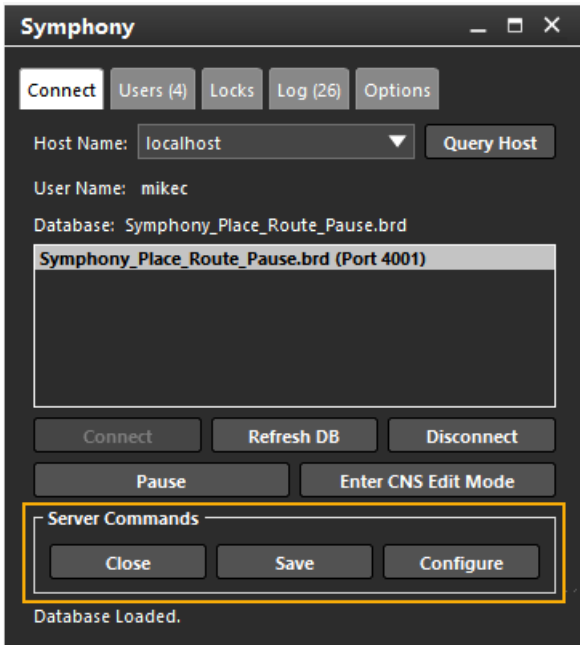
The Symphony server can run on a virtual machine with or without a graphical interface. The server can be executed on the command line with switches to set basic configuration during startup as well as the ability to run in *no graphics* mode. The command arguments for `muserver.exe` are as follows.

`-safe`	Runs without user or site configuration files and settings
`-nographic`	Specifies to invoke server application without user interface
`-port <# or range>`	Specifies port range
`-autosave`	Enables the autosave system
`-autosave_name <name>`	Specifies the custom autosave base name
`-autosave_vers <#>`	Specifies number of autosave versions
`-autosave_int <#>`	Specifies autosave interval

The following example starts Symphony server application in no graphics mode communicating over port 4000.

```
muserver -port 4000 -nographic Symphony_Demo.brd
```

Note: Running Symphony server application in no graphics mode can only be controlled from the layout editor session of the user, who started the server.



Starting Symphony Server Using Symphony Server Manager

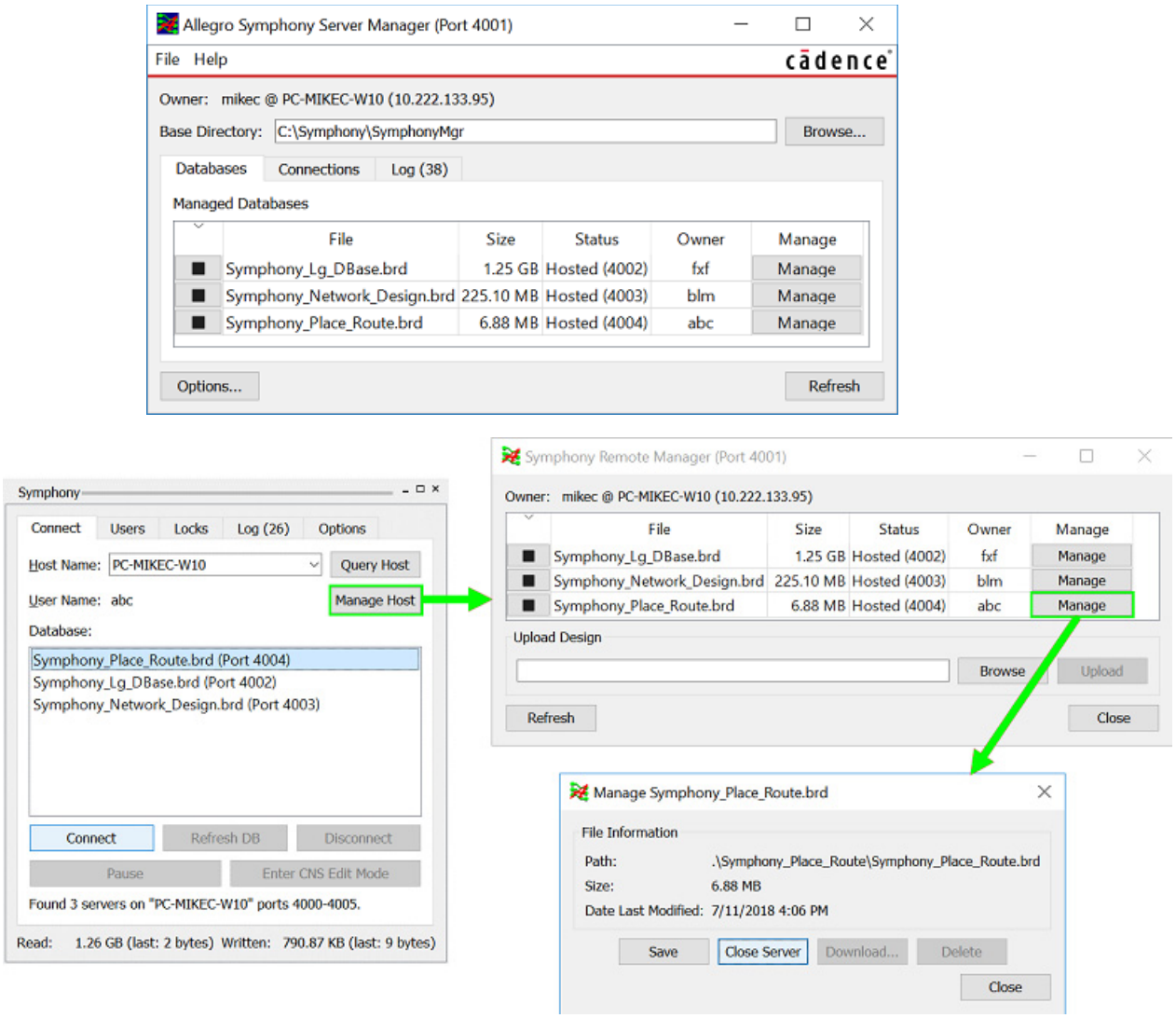
The Symphony Server Manager (`muservermgr.exe`) provides remote management of Symphony server applications on a dedicated hardware server. You can start the server application on a remote server with or without a graphical interface. The remote server application listens requests from layout editor users to start and manage Symphony server applications. No special hardware configuration or requirements are needed except the ability to run standard layout editor on the remote machine.

Note: Only one Symphony Server Manager application is supported per remote server.

Advantages of Symphony Server Manager

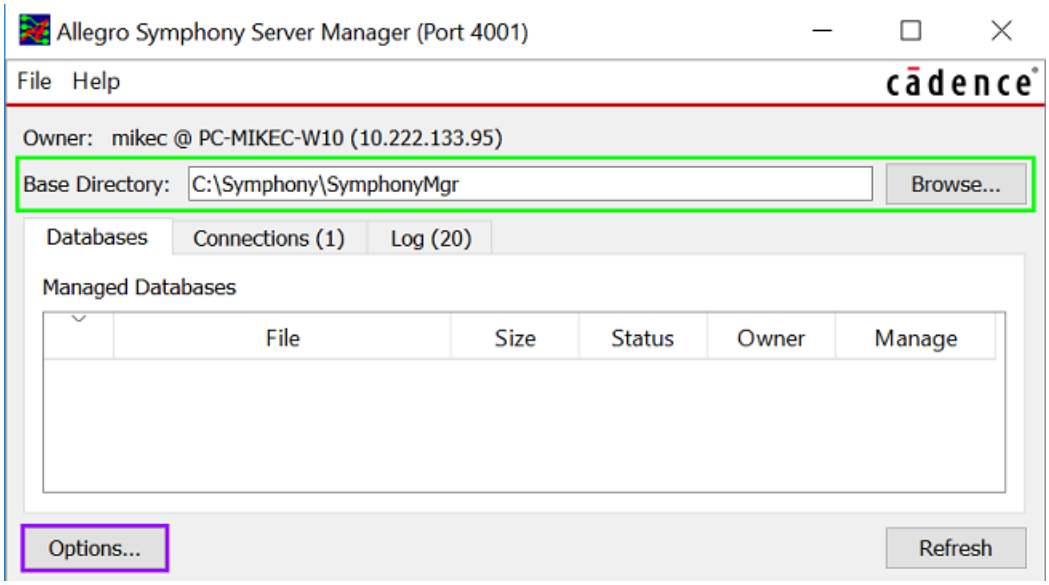
- Eliminates the need to login to the remote server to start Symphony server application
- Dedicated hardware server
- More reliable up-time and centralize access

- Remote manager within the layout editor can be used to upload databases and to start Symphony server applications
- Windows services configuration is available so that manager can run automatically

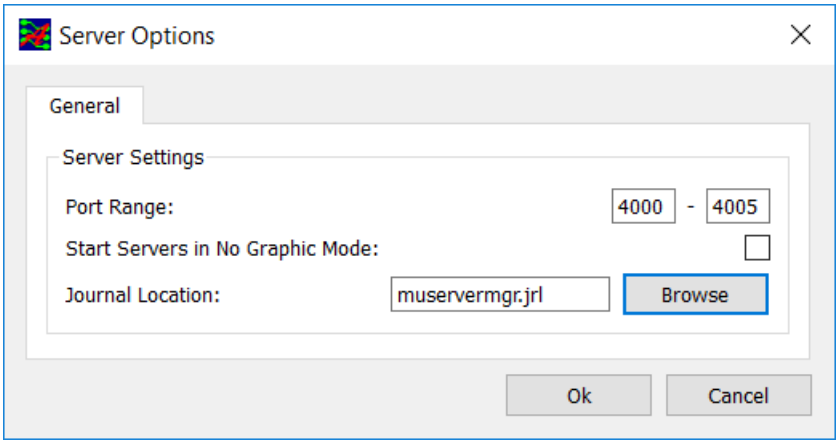


Starting Symphony Server Manager (`muservermgr.exe`) on the remote server requires the following configurations:

- Base Directory: Set the location of base directory where all databases are saved. By default, it is set to `PCBENV/SymphonyMgr` directory.

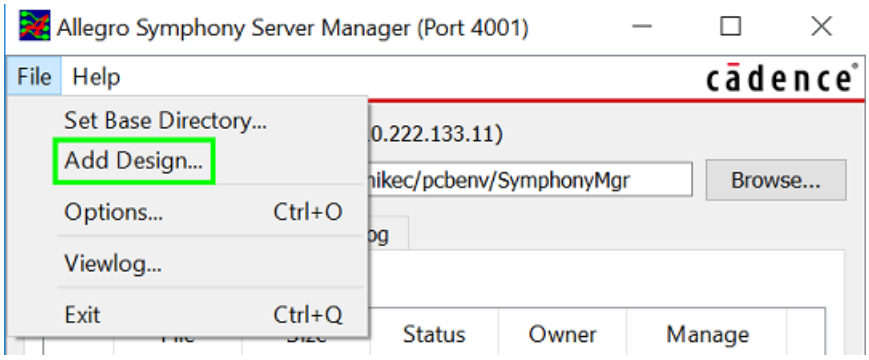


- Server Options: Specify the journal file location. By default, it set to the path of the database from which server manager was started.

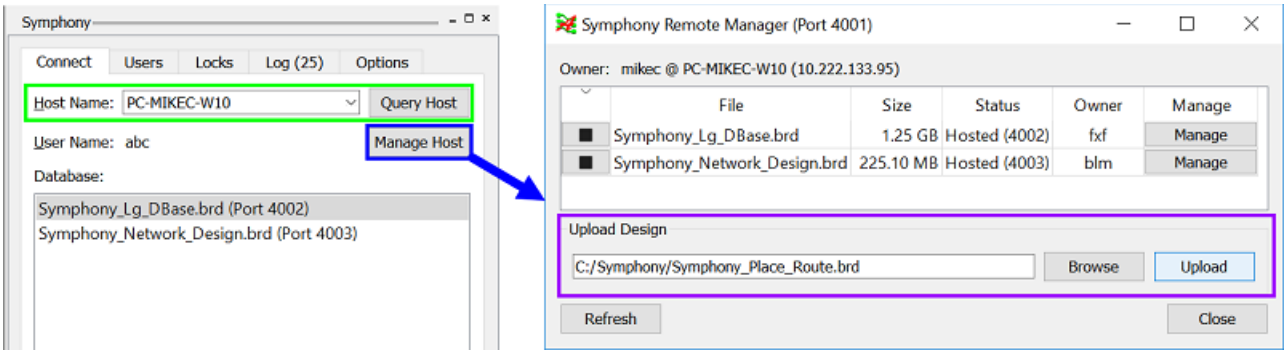


You can upload a database to the Symphony Server Manager either from the server manager or from the layout editor.

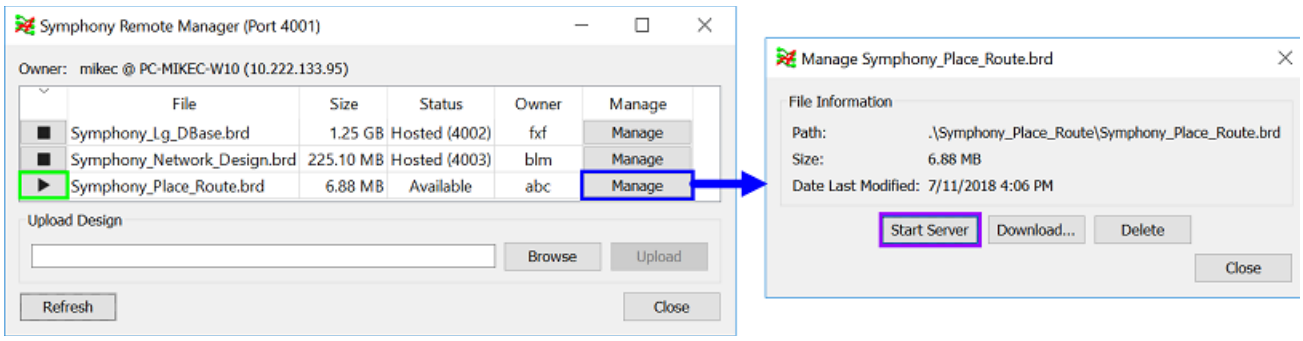
To upload a database directly in the Symphony Server Manager, choose *File – Add Design* in the Symphony Server Manager.



If uploading a database from the layout editor, choose *File – Symphony Connect* menu option. In the *Connect* tab of the server application, specify the host name of the Symphony Server Manager. Click *Query Host* and then click *Manage Host* button. The Symphony Remote Manager window opens and displays the database to upload in server manager.

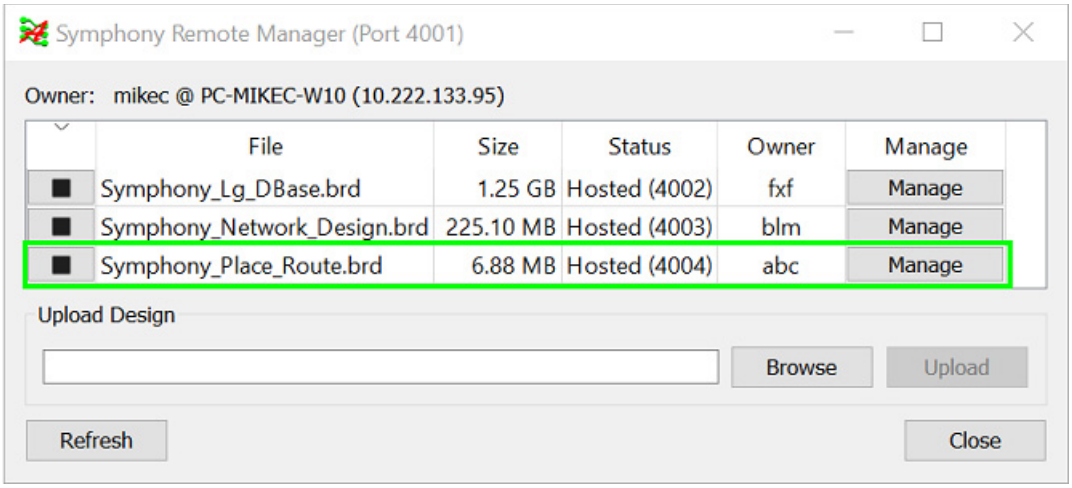


Once uploaded, the database becomes available on the Symphony Server Manager. To start the server application instantly, click the play button associated with the database in the Symphony Remote Manager window. If you want to check the file information before starting the server application click *Manage* button.

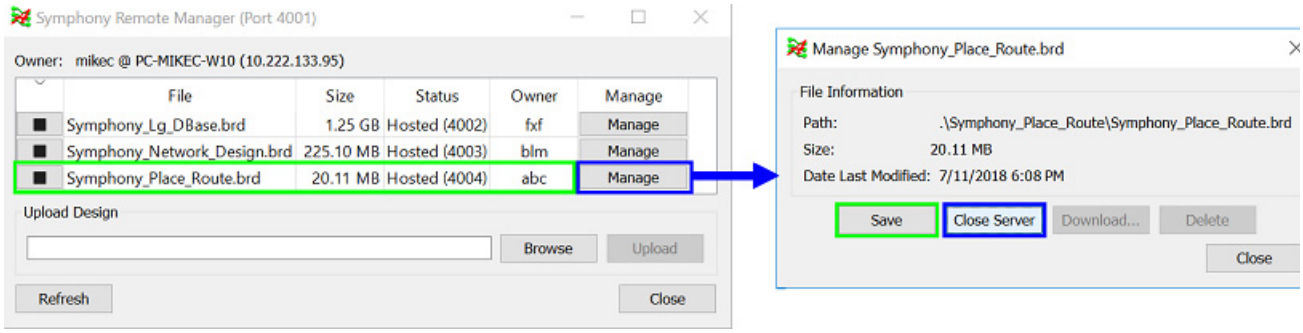


Note: All the clients can access *Manage Host* button and Symphony Remote Manager window to *Upload*, *Start*, *Save* and *Close* Symphony server applications.

The remote server runs the Symphony server application and shows the status of the databases as *Hosted* with the port number they are using. Clients can see the available databases and join the session.

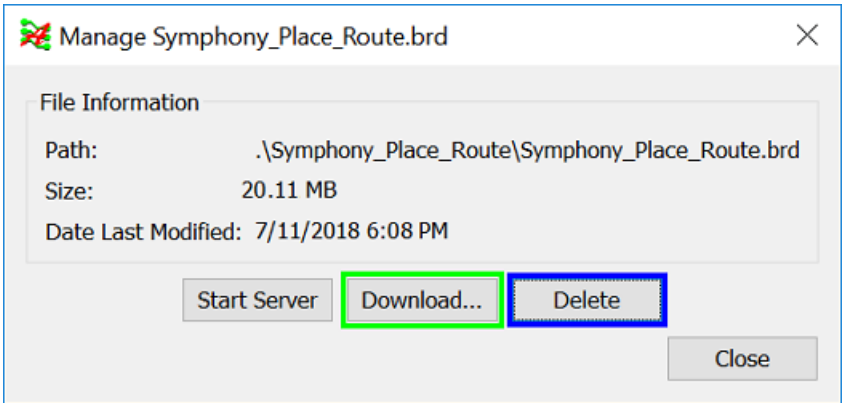


To get the final database, click *Manage Host* button in the Symphony server application to open the Symphony Remote Manager. Select the database and click *Manage* button. The *File Information* is displayed. You can save the final database and close the server application.



Once the server is closed the *Download* and *Delete* buttons becomes active. To get a local copy of the final database, click *Download* button. To remove all server related files

generated during the session in the server working area including the final database, click the *Delete* button.



Note: Only the client who uploaded the database to the Symphony Server Manager and reported as the owner in the available databases list can download and delete the final database.

Symphony Server Manager as a Windows Service

Symphony Server Manager functionality can be configured as a Windows service and has certain advantages over running a standalone server. Running server manager as a service is:

- More reliable up-time and automatically restarts on system reboot.
- Accessible all the time, even if no user is not currently logged into the machine.
- Eliminates the need to login to the remote machine to start Server manager application.

Note: To successfully start the service, the `muservermgr.exe` must be launched from its default location in the Cadence installation hierarchy (`<installation_directory>\tools\bin`).

Starting Windows Service

To access the Windows services configuration, use any one of the following methods:

- Start *Task Manager* and open *Services* tab. Select the service, right-click and choose *Open Services* option
- Open *Run* window and type `services.msc`
- Open command prompt and type `services.msc`

- From the command line using the `sc` command

Important

To determine the best way to configure the Windows services per organization policies check with the IT department.

Configuring the Symphony Server Manager

You can setup the Symphony Server Manager as a Windows service using the following ways:

Service Controller Utility

To manage the service using the `sc` command, use the following steps:

- Open *Run* window in administrator mode and type `cmd`.
- Enter the following commands in the system command window to install the service:

  ```
  sc create <Service Name> binPath="<muservermgr.exe>"
  ```

  - Choose a specific `Service Name`. For example, Cadence Symphony Server Manager.
  - Set the `binPath` to the Cadence installation hierarchy (`<installation_directory>\tools\bin\muservermgr.exe`) on the local machine.

  Note: The UNC or network drives paths are not supported.
- To delete the service, enter the following command in the system command window:

  ```
  sc delete <Service Name>
  ```
- Enter the following commands to start, query, stop, and restart the service:

  ```
  sc start <Service Name>
  sc query <Service Name>
  sc stop <Service Name>
  sc restart <Service Name>
  ```

Windows Service Dialog

The Windows *Services* graphical interface cannot be used to install or uninstall services, but can be used to view and manage already installed services. The same actions of starting, stopping, and restarting of services may be done through this dialog, and the underlying operations remain the same as though performed on the command line.

Command Line Options

To directly interact with the Window service manager, the Symphony Server Manager provides similar capabilities through command line options:

```
muservermgr -install

muservermgr -uninstall

muservermgr -start

muservermgr -stop

muservermgr -restart
```

To use these command line options administrator privileges are required. These options start the same underlying functionality as the `sc` command and the *Services* dialog. For each option, the Symphony Server Manager process attempts to complete the specified operation and exits.

Symphony Server Manager: Application vs Service

When running as a service, the Symphony Server Manager acts slightly different than when run as an application.

For the Symphony Server Manager application, the default working area (*Base Directory*) is set to the `PCBENV` directory in a sub-directory named `SymphonyMgr`. Whereas the Symphony Server Manager process uses the `AppData` or `ProgramData` directories to store the service log as well as the databases (`.\Cadence Design Systems\Symphony Server Manager\SymphonyMgr`).

The Symphony Server Manager when running as a service has no graphical interface. The clients can access the Symphony Server Manager from the layout editor using the Symphony Remote Manager dialog. The status of the Symphony Server Manager can be checked through Service log or through the `muservermgr.jrl`.

Starting Symphony Server Manager in No graphics mode

The Symphony Server Manager can run on a virtual machine with or without a graphical interface. The manager can be executed on the command line with switches to set basic configuration during startup as well as the ability to run in no graphics mode.

The following are the command arguments of `muservermgr.exe`:

```
muservermgr [args] [database repository path]
```

For more information, see syntax of `muservermgr` command in the *Allegro PCB and Package Physical Layout Command Reference* guide.

The Symphony Server Manager application running in non-graphical mode provides access to the available database through the Symphony Remote Manager dialog in the layout editor. Perform the following steps in the Symphony Remote Manager dialog to access the database:

- Enter *Host Name*
- Click *Query Host* button
- Click *Manage Host* button

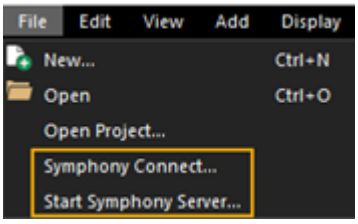
Informal First-Client Driven Team Design Mode

Informal first-client driven team design mode starts the standalone Symphony server application from a layout editor to share the open database from the local system of the database owner.

To start Symphony server in this mode, start a layout editor and select the *Symphony Team Design* product option.

Note: The *Symphony Team Design* product option is a part of the product bundle in higher tier products. and is not available to select separately. For example, Allegro Enterprise PCB Designer.

Two additional entries, *Symphony Connect* and *Start Symphony Server*, are visible under the *File* menu.

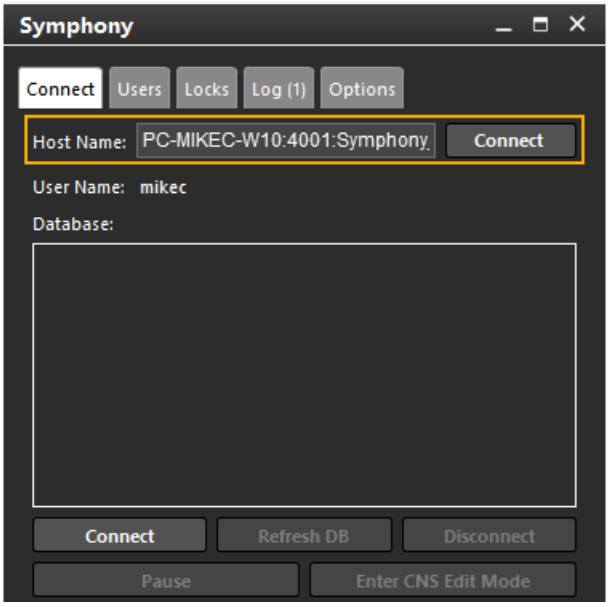


Select the *Start Symphony Server* menu option, to start the Symphony server application and share the currently saved database. Multiple clients can now connect to the selected database and perform design activities in a concurrent environment.

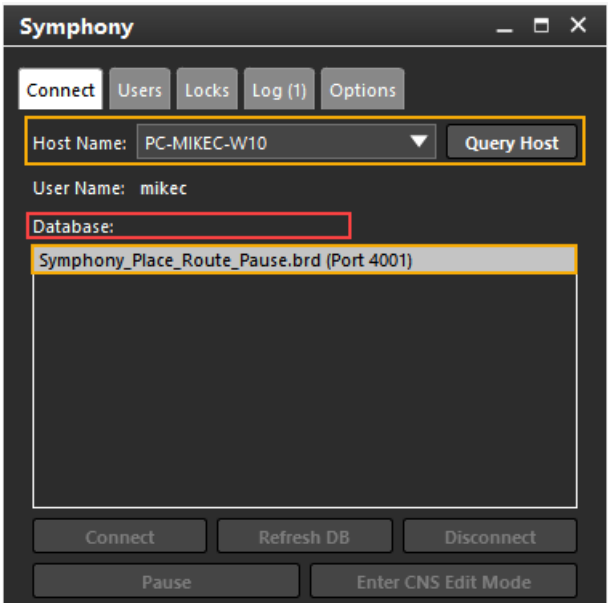
Note: The server application will be started using the name of the currently open database. Save the active database before starting the Symphony server application.

The user who starts the server is automatically connected to the server database and able to work in the concurrent environment.

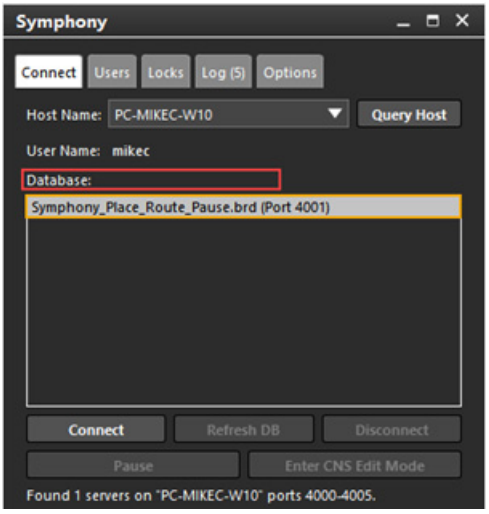
Users who want to participate in the concurrent team designing, need to select the *Symphony Connect* menu option to join the Symphony server application. Clients paste server link provided by the server owner in the *Host Name* field and click the *Connect* button.



The Symphony server database is visible under a TCP Port on the host machine (server) available for clients to connect to the database. Alternatively, clients can enter the host name or IP address to connect and query the host available Symphony databases.



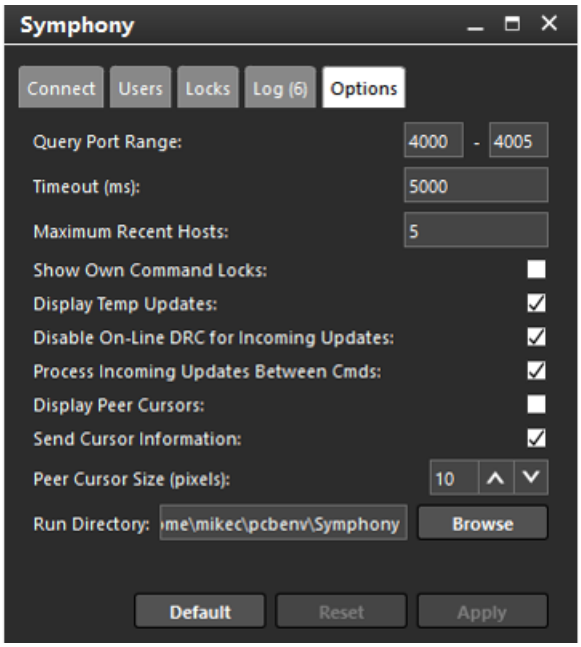
The above figure shows the state, where clients are not yet connected to the database, as a query has been sent to the host to check for available databases. The query results are populated in the *Database* section of the form.



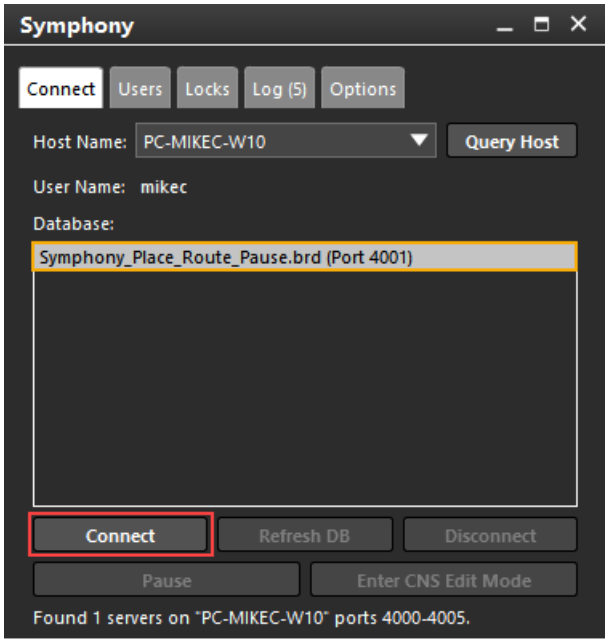
Note: The server displays the *No Open Database* indicator when running without a database loaded. It occurs when the design hasn't fully loaded, when it has failed to load, or when it has been closed from the server UI. If the design still needs to finish loading when it

is queried, re-querying the server again displays the design name when it is fully opened. For more details, view the `muserver.jrl` file.

The parameters for team design environment are set in the *Options* tab of the *Symphony* window.

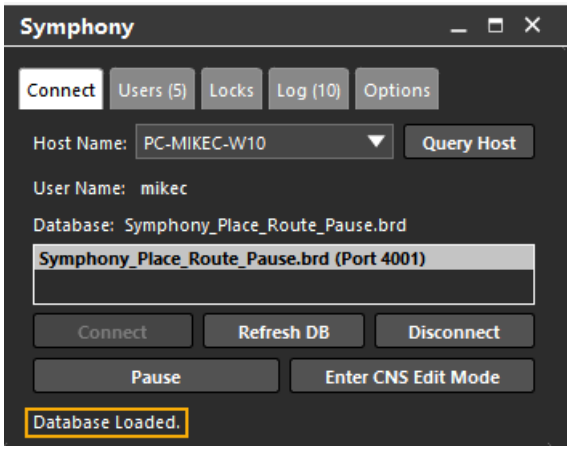


When the list of shared databases is available, select the database and click *Connect*.



Note: This connection step is only required for other clients as the user who started the server from within a layout editor is automatically connected to the local server database.

You can view the amount of data loaded into layout editor in percentage. When the database is loaded from the server, status is displayed as *Database Loaded*.

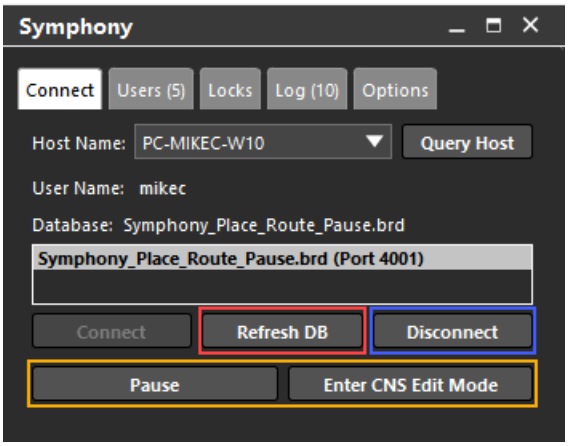


The action buttons for client-side are *Refresh DB, Pause,* and *Disconnect* are enabled.

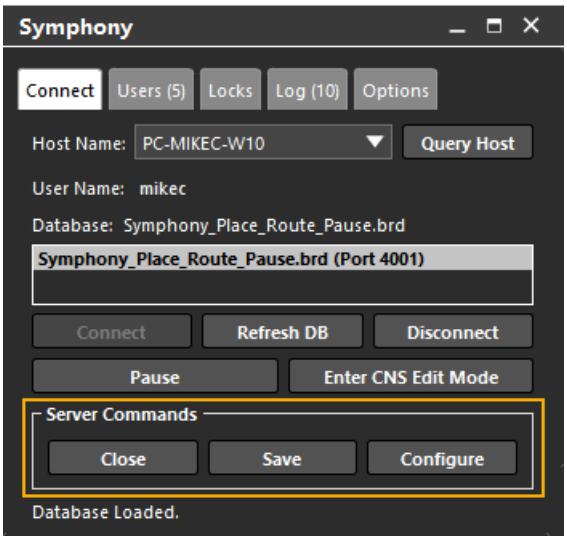
Database is refreshed using the active copy on the server. The *Refresh DB* button does not use often as database updates are sent to each client automatically.

To make design changes outside the concurrent design environment, clients can pause the session. Using *Pause* button, client can take the full control of the database to do the tasks such as netlist import, netlist back annotation and so on.

Disconnecting from the Symphony server database returns to standalone layout editor.



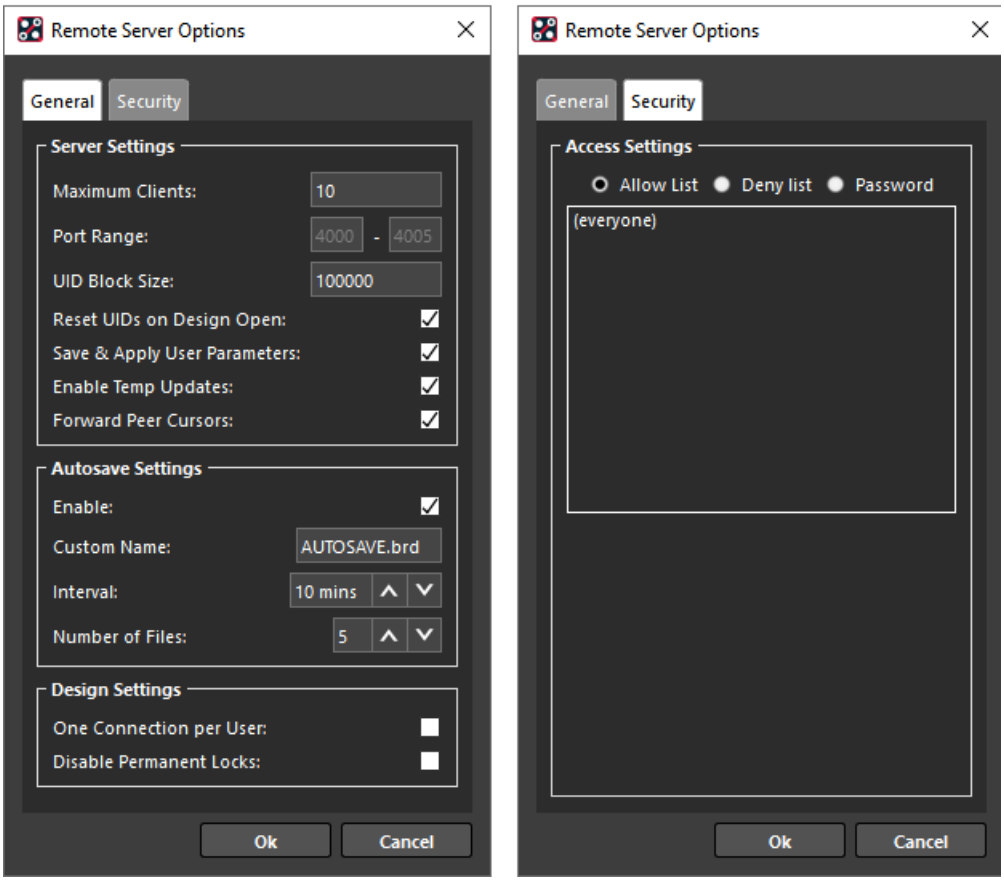
The *Server Commands* section under the *Connect* tab is only available for the client – who started the Symphony server application.



In this case, the Symphony server application (`muserver.exe`) has options to remotely communicate to Symphony server application.

- **Close**: Closes the Symphony server application and prompt to save changes. All the clients are also gets disconnected from the server application.
- **Save**: Saves the active database running in the Symphony server application and leaves the server active.

- **Configure**: Remotely controls all the Symphony server application options except *Port Range*.



For more information, see `muconnect` command in the *Allegro PCB and Package Physical Layout Command Reference* guide.

Interaction Between Layout Editor and the Symphony Server Application

Design updates are committed to the server database as each command is completed with “in commands” updates being seen by all the clients.

Users can join and leave the session at any time – server maintains the updates. Client-level parameters, zoom levels, and layer display are maintained when the user rejoins the session.

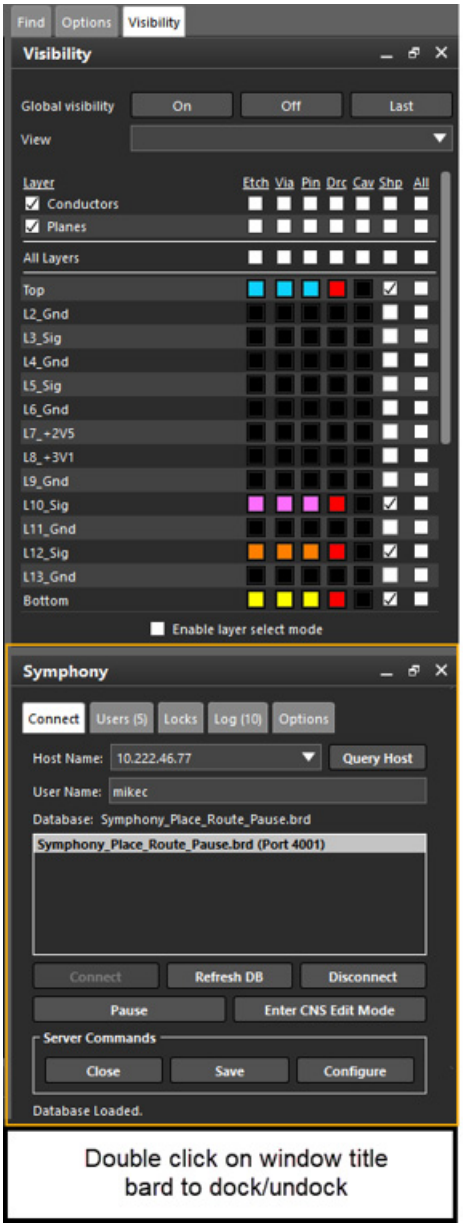
Save commands are not available but *Write As* capability allows a local copy to be saved, only if absolutely necessary. All design changes completed while connected to Symphony database are incorporated to the server database.

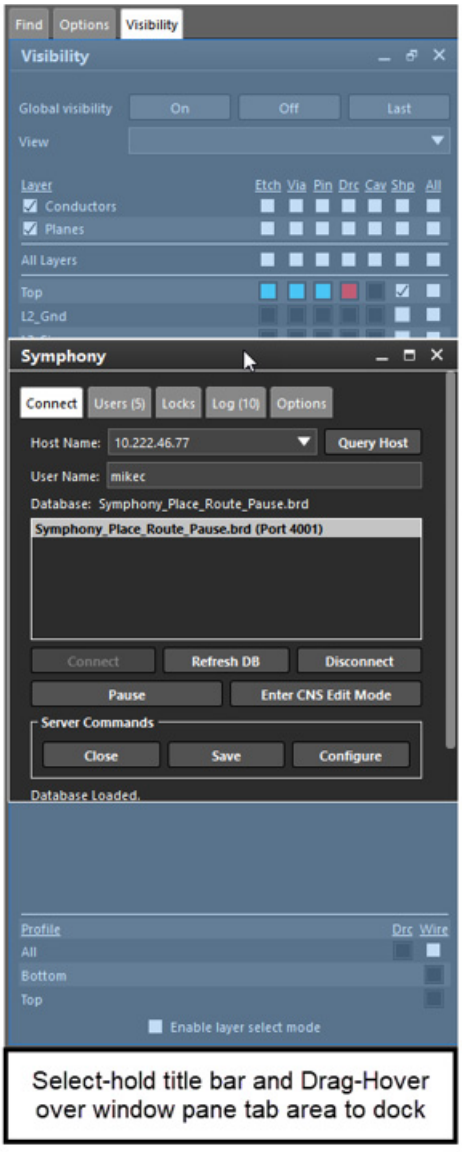
No off-line updates/imports to Symphony Server database are supported inside concurrent design environment.

Symphony Window

The *Symphony* window can be docked or undocked to the lower-right corner of the *Windows* pane tab area by double-clicking on the title bar. You can also select the title bar of the window and drag and drop it on the *Windows* pane tab area to add *Symphony* as a *Windows* pane tab.

The position of mouse cursor determines the location where the *Symphony* window will be docked; upper, lower, or middle to add as an additional tab.



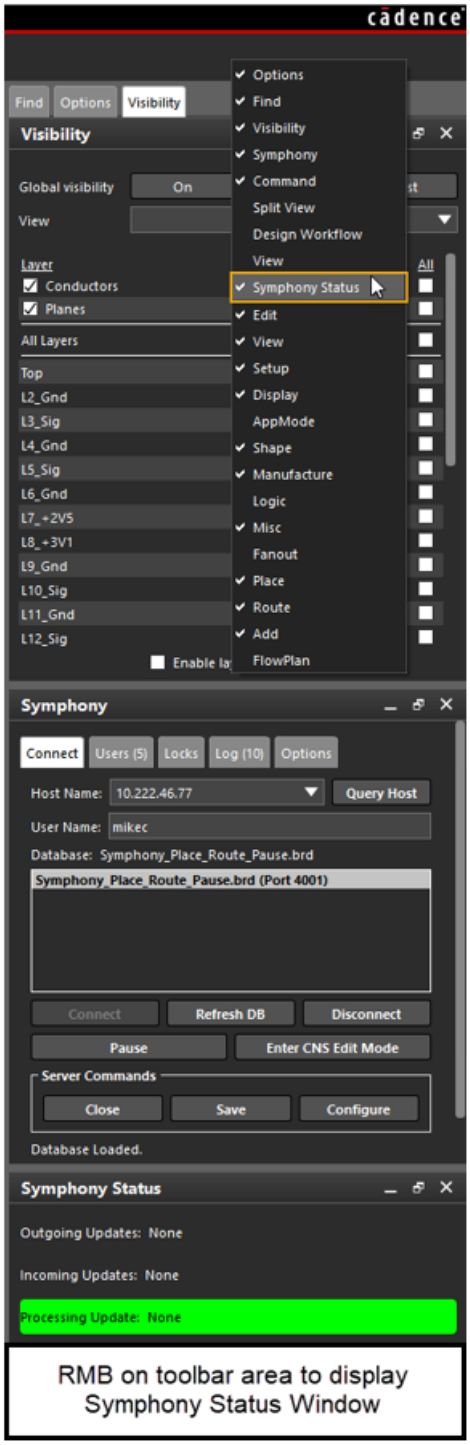


Symphony Status Window

Right-click the toolbar area and choose *Symphony Status* to display. The *Symphony Status* window pane is docked on the lower portion of the *Windows* pane tab area.

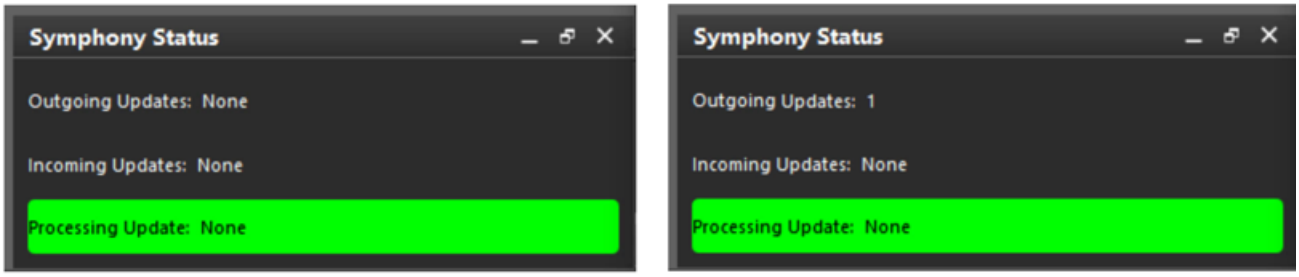
The *Symphony Status* window provides a summary of *Outgoing Updates, Incoming Updates* and *Processing Update*.

Note: *Symphony Status* should only be used for informational purposes and should not interpreted as an “out-of-sync” condition or a server issue.

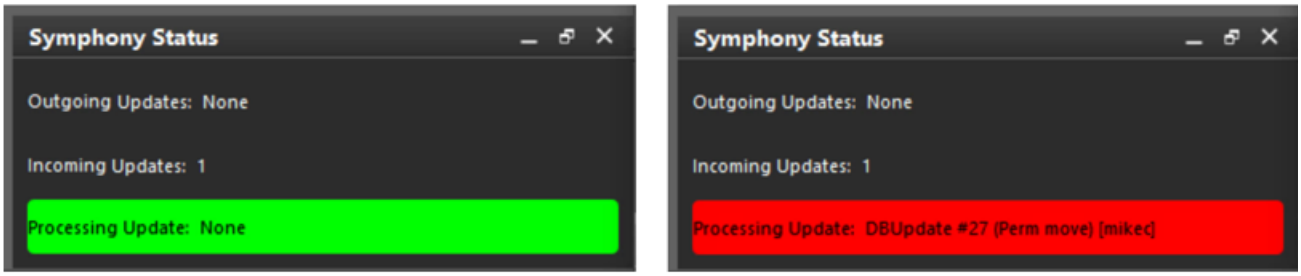


Outgoing updates shows the number of updates sent to server for integration. When accepted by the server, the number decreases. Whereas, incoming updates shows the number of pending client updates integrated on the server, but not yet integrated locally. Processing update shows the updates being processed with the DBUpdate tracking number, command, and the client that initiated the change.

Design change made on Client #1



Client #2 notified of incoming update and automatically processed



Being in an active command blocks server updates from being integrated. Checking the *Status* window for incoming updates shows the number of pending server updates. When the active command is completed, all pending incoming updates are processed.

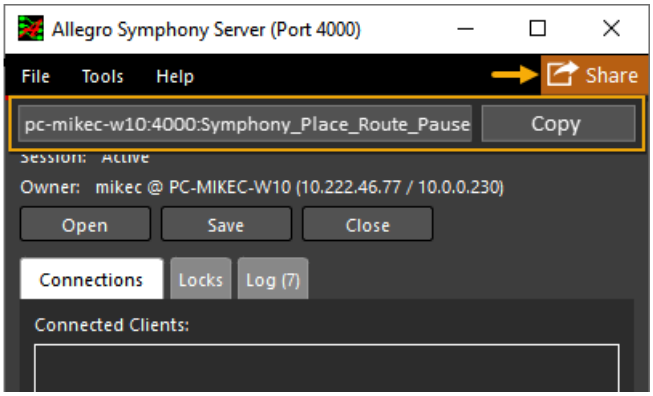
Depending on the number and type of updates, the layout editor may go into a busy state to complete updates. Application mode commands may allow updates to happen more frequently.

Updates normally process quickly and having pending updates should not be a concern.

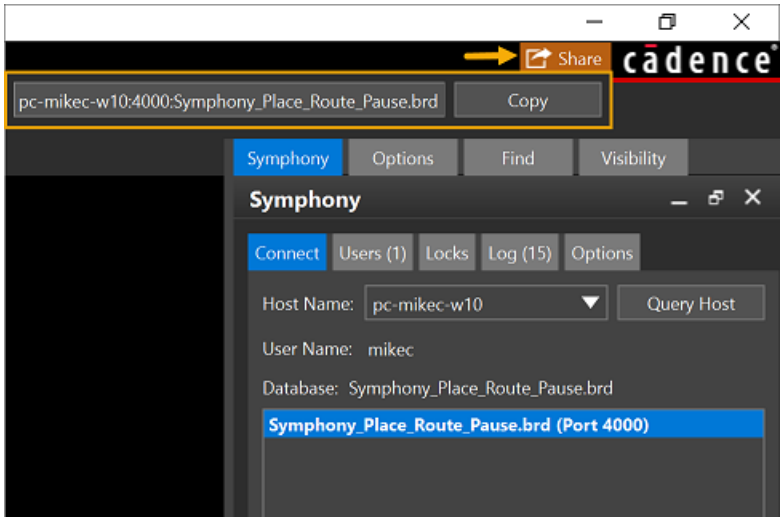
Share Link

To make it easier to connect to the Symphony Session, use the *Share* button that exist at the top-right corner of the Symphony Server and the Layout Editor user interface.

Design owners that start the Symphony Server have access to the *Share* button to generate a share link that can be provided to users to connect to the session.



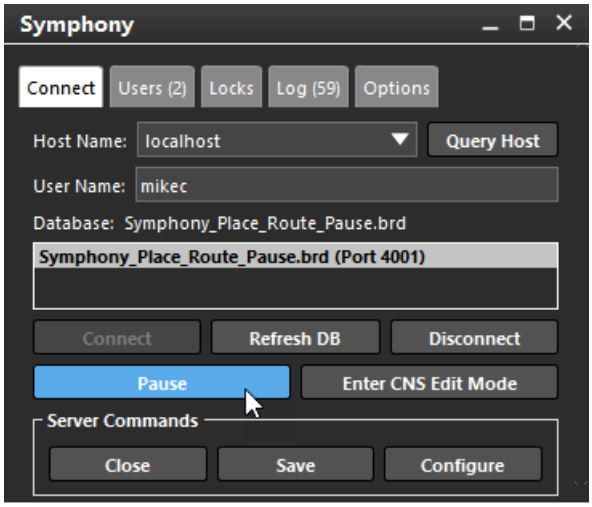
Clients connected to the session also have access to the *Share* button to generate a share link for other users to quickly connect to the session.



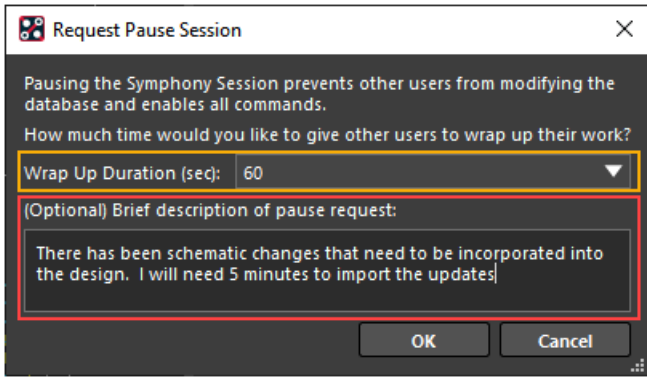
Pause Session

The *Symphony* window provides an ability to clients to temporarily pause the session and take full control of the server database to modify the database outside the concurrent design environment.

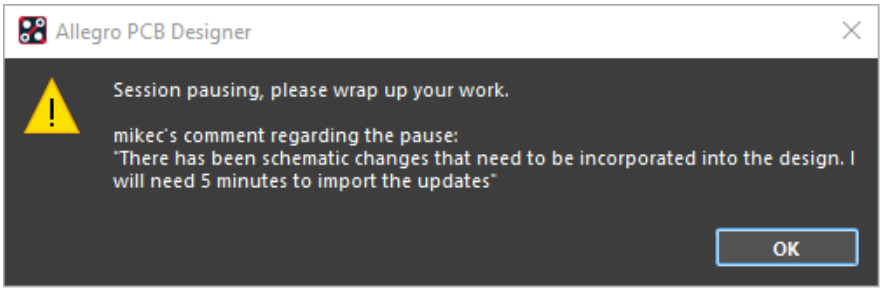
For pausing a session, select the *Pause* button in the *Connect* tab.

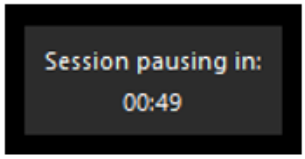


Before sending a request to pause the session, enter the *Wrap Up Duration* in seconds. Specify a reason to issue a pause request and click *OK* button in the *Request Pause Session* UI.

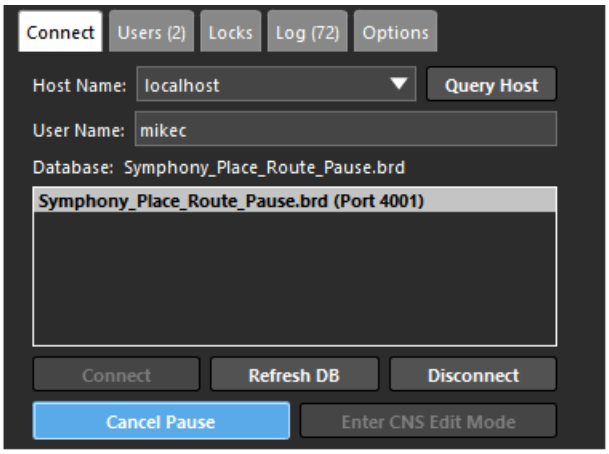


All the other clients receive a notification to conclude the work. A countdown clock starts displaying at the top right corner of their design canvas.

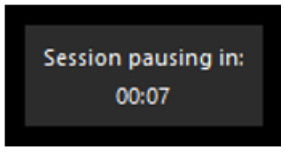


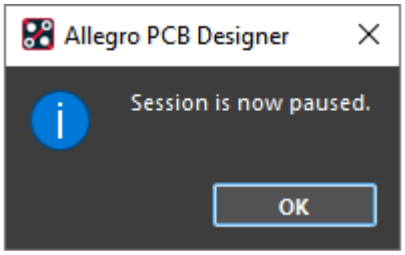


If the other clients require more time to wrap up their work, the client that requested the pause can cancel it during the wrap-up duration countdown by clicking on the *Cancel Pause* button.

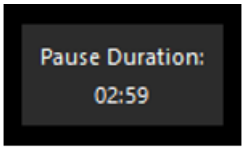


The countdown clock turns red when less than 10 seconds are left and a notification dialog pops up once the session is paused. The countdown clock remained in the *Paused* state during the requested time span.

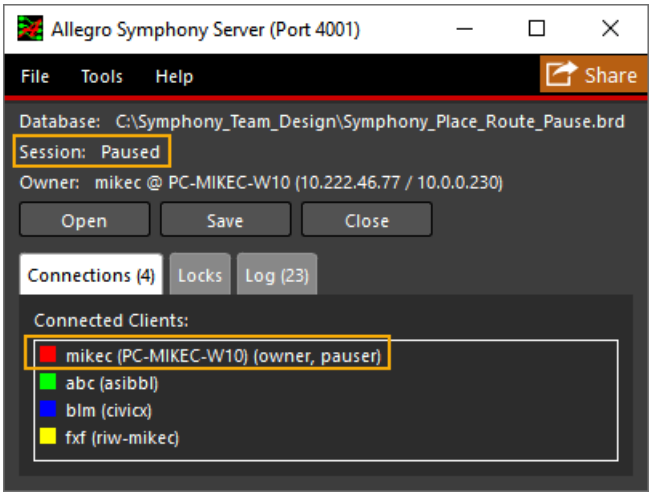




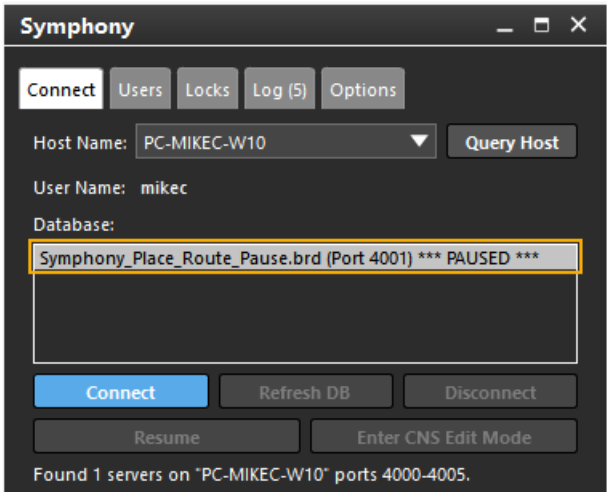
For the client who initiated the pause session, a duration counter starts and the server database becomes available for standalone operations.



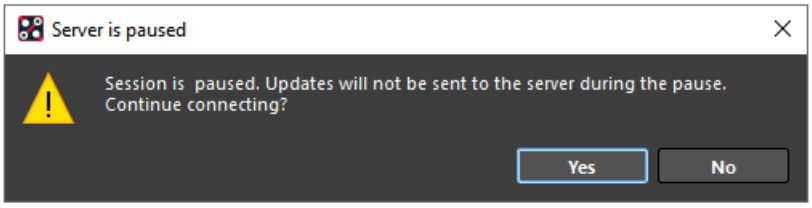
In the Symphony Server application, the status of the session changes to *Paused* and reports the client who paused the session.



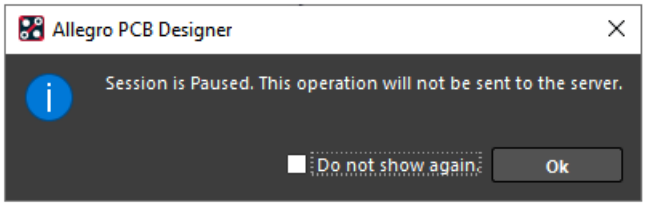
An identifier PAUSED is added to the databases that are paused in the *Connect* tab of the *Symphony* window. The paused databases still available for new clients to connect.



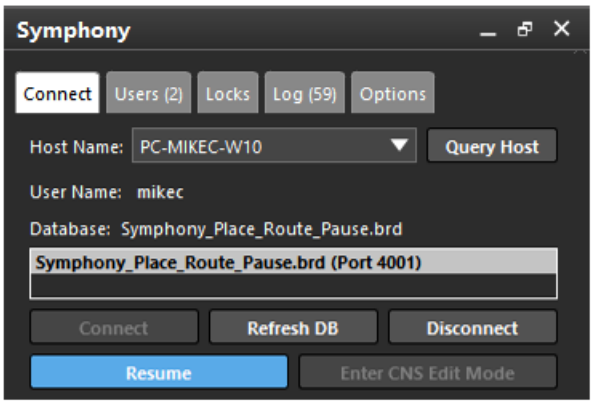
New clients, however, only view the database in its pre-paused state. On trying to connect to a paused database, a warning message displays.



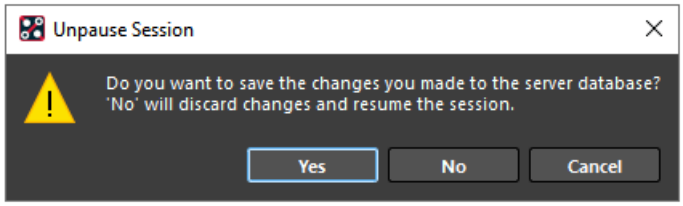
When connected, clients can modify the database, but the changes will not be integrated to the server. A reminder message is displayed after completion of each command that can be suppressed by enabling *Do not show again* check box.



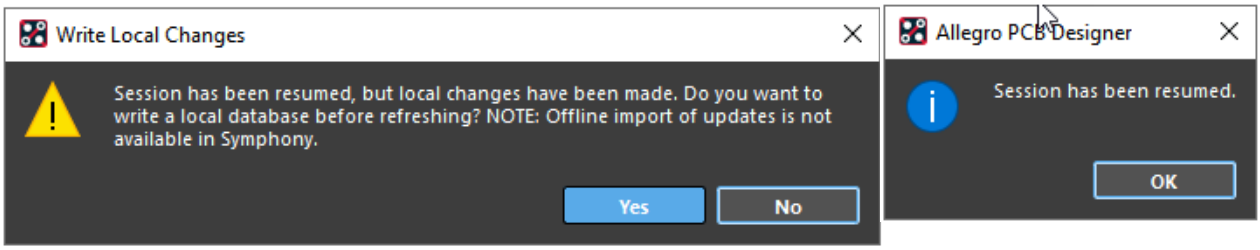
Once the database changes are completed, the client who paused the session clicks the *Resume* button in the *Connect* tab to update the server database.



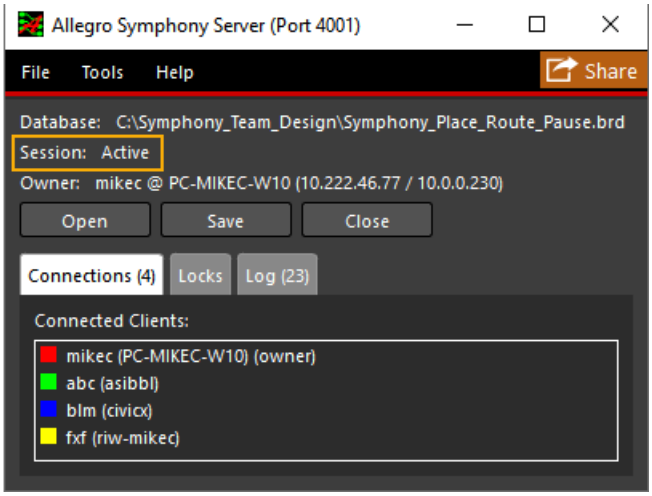
The client also has an option to discard the changes and release the pause any time.



The other clients receive a notification immediately and get a latest copy of the server database. If other clients made any changes during the paused session they can save a local copy for reference only.



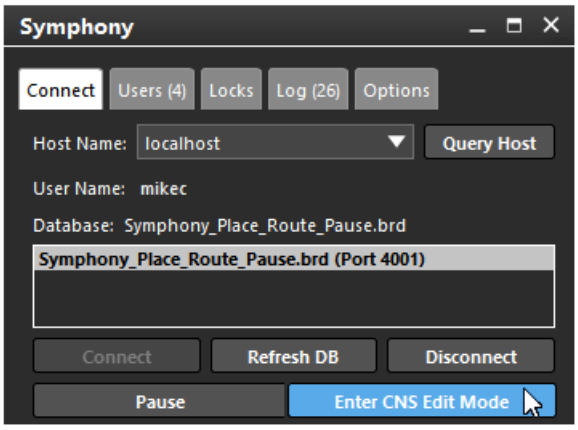
The status of the Symphony Server application session returns to *Active*.



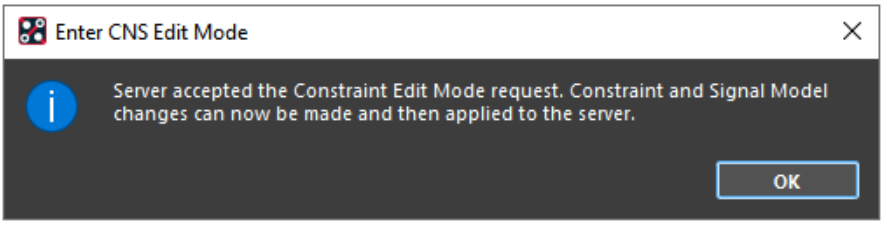
Constraint Edit Mode Flow

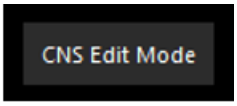
Each client can enter Constraint Edit Mode (CNS Edit Mode) to update design constraints while everyone in the Symphony session continues their concurrent design work. Only one client at a time can enter CNS Edit Mode. After the constraint updates are completed the server database and all other clients are refreshed.

A new button *Enter CNS Edit Mode* is added to the *Connect* tab of the Symphony server window.

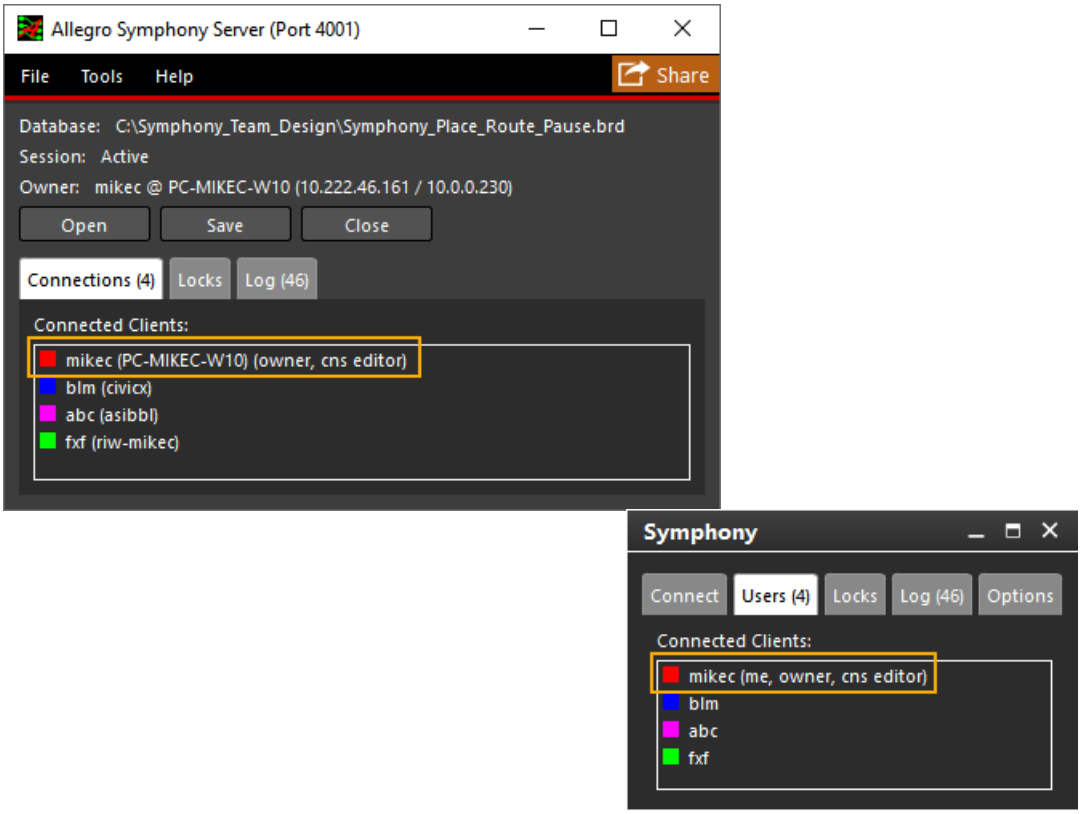


The client who requests to *Enter CNS Editor Mode* receives a confirmation message from the server. Clicking *OK* in the message starts displaying a label *CNS Edit Mode* at the top right corner of the client's design canvas.





The *Connections* and the *Users* tab of the Symphony server application also indicates the client name who has entered CNS Edit Mode.

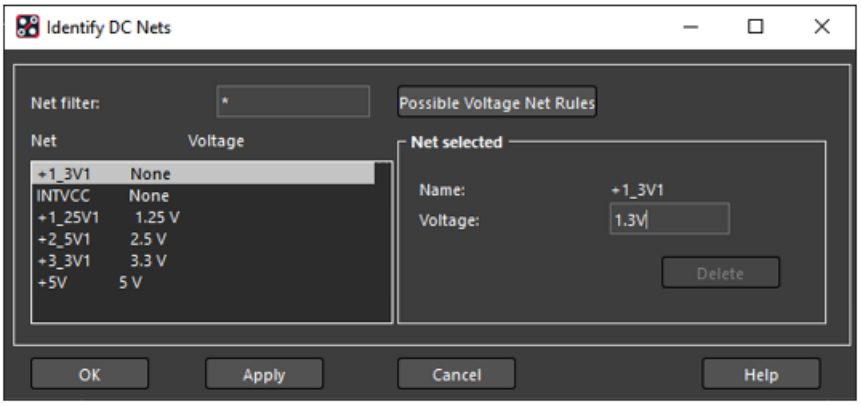


Note: Only one client at a time can enter CNS Edit mode.

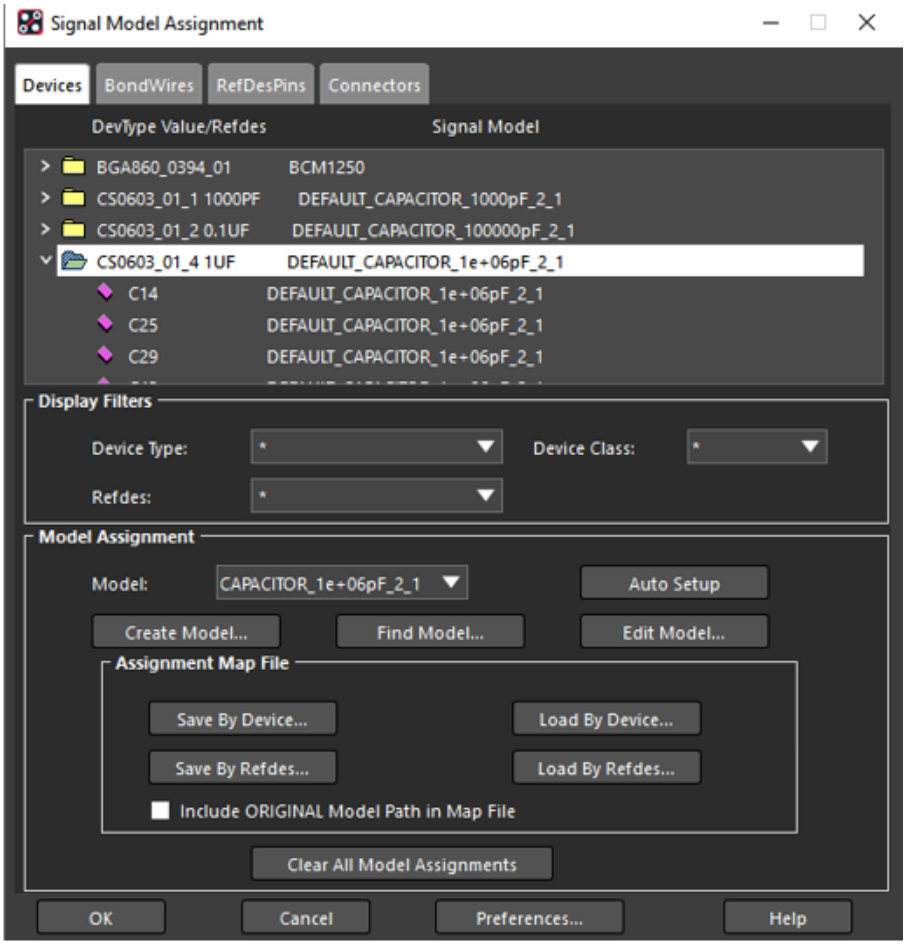
Tasks Performed in CNS Editor

The CNS Editor can perform the following activities:

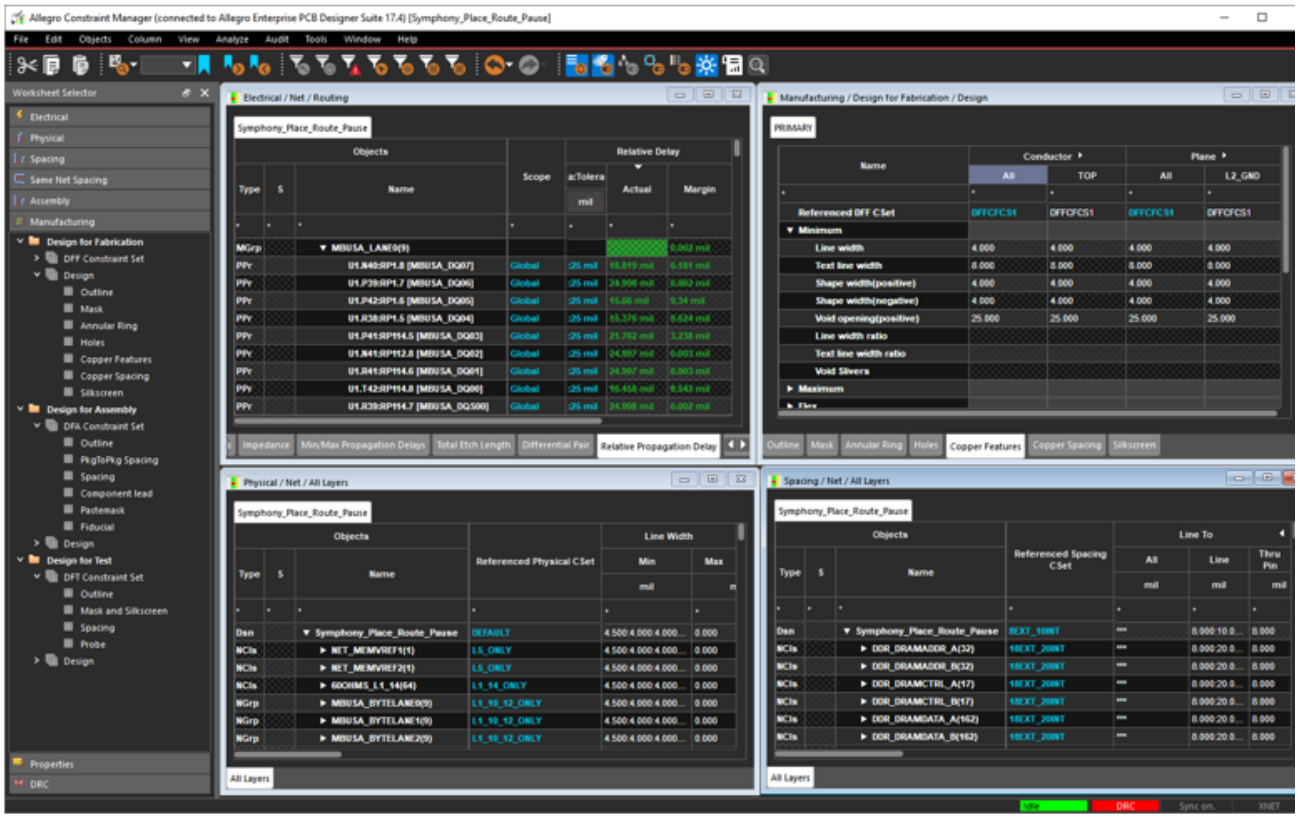
- Identifies DC Nets to assist in creating XNets and assigning signal model.



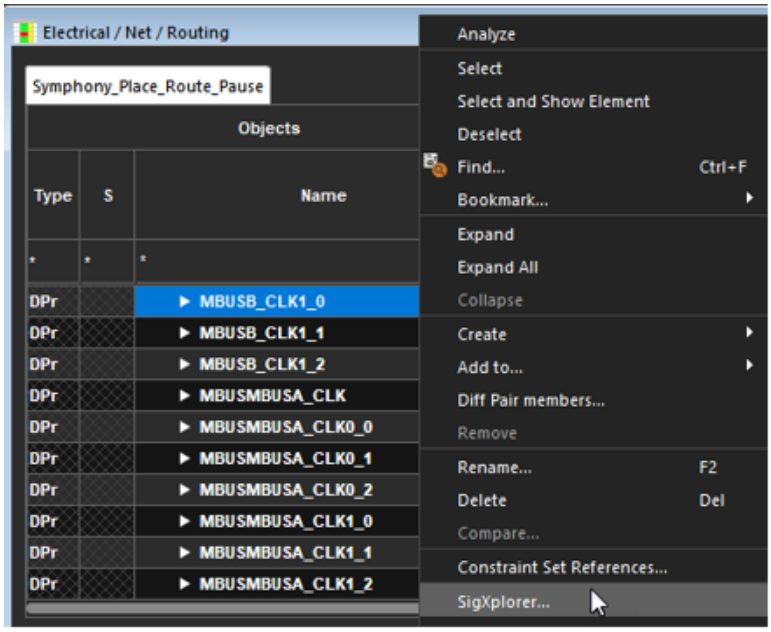
- Creates and updates signal model assignment.

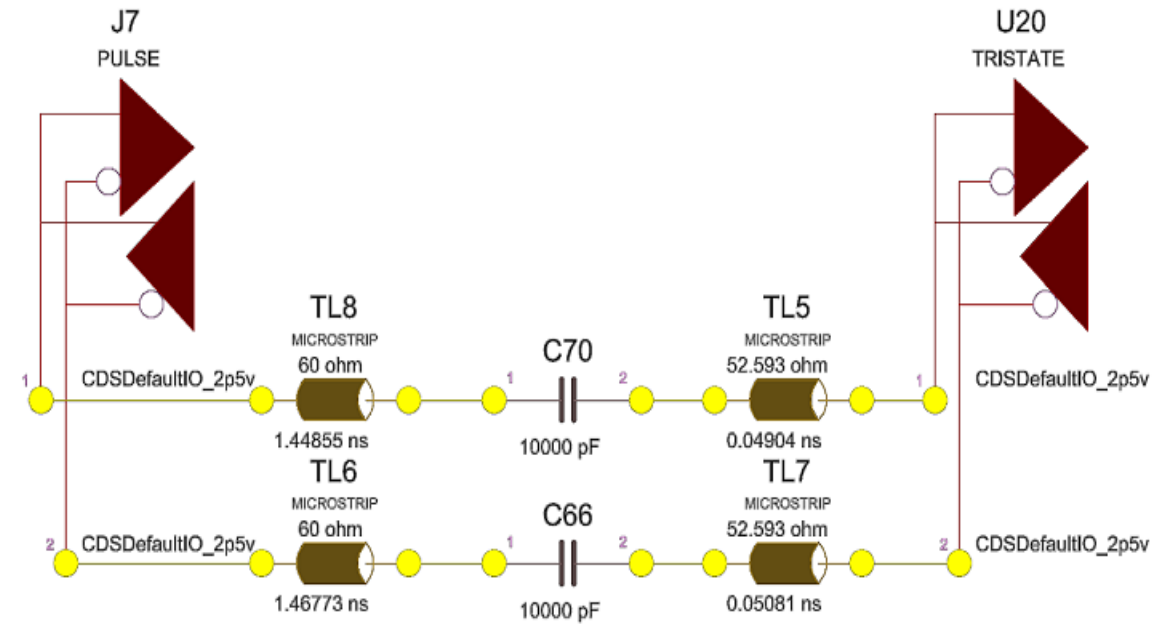


- Provides access to Constraint Manager. All the domains with full hierarchal support are accessible. Users can create and manage hierarchical objects, such as Net Class, Net Group, differential pairs and so on.



- Provides topology extraction in SigXplorer with Electrical Constraint Set definition.

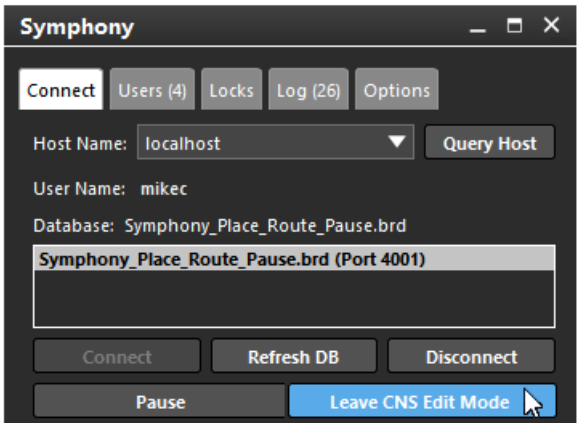




- Supports non-constraint design updates to assist in the validation of design rules. But only constraint-related updates are accepted by the server.

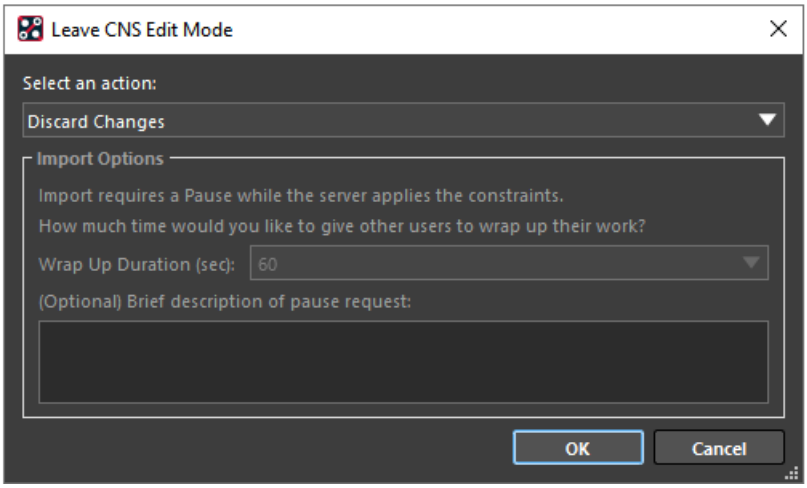
Note: Other design-level modifications for example, cross-section updates, netlist updates, import IDX and so on must be completed outside of CNS Edit Mode.

After making changes in constraints values, client selects the *Leave CNS Edit Mode* button under the *Connect* tab and requests server to update the database.

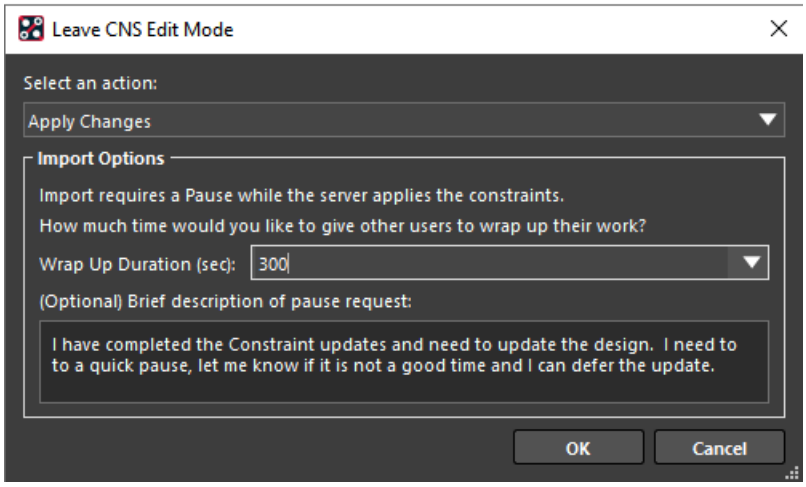


The *Leave CNS Edit Mode* dialog opens, which has two action options.

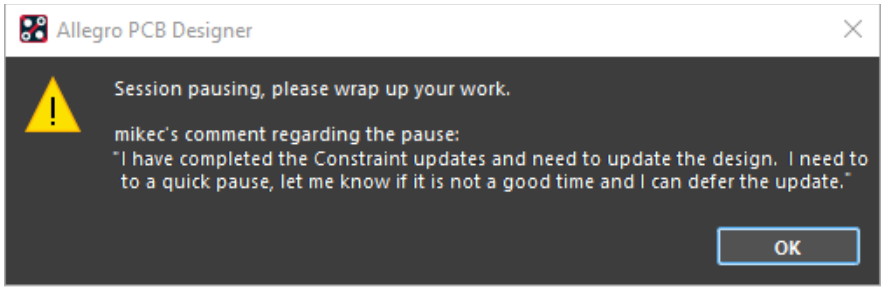
- *Discard Changes*: Discards all the changes. Clicking *OK* button cancels the Constraint Edit Mode request and allows the client to rejoin the Symphony session.



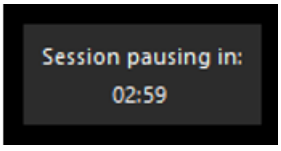
- *Apply Changes*: Applies all the changes. Specify the *Wrap Up Duration* and the reason for requesting a pause session. Clicking *OK* button initiate a pause session.

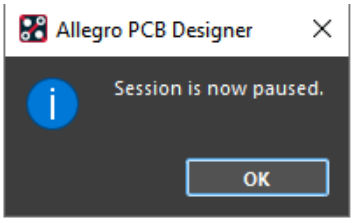


All the other clients receive a notification to conclude their work. A countdown clock starts displaying at the top right corner of their design canvas.



The countdown clock turns red when less than 10 seconds are left and a notification dialog pops up once the session is paused. The countdown clock remained in the *Paused* state during the requested time span.





Client sessions remains paused during constraint updates.

SKILL Support

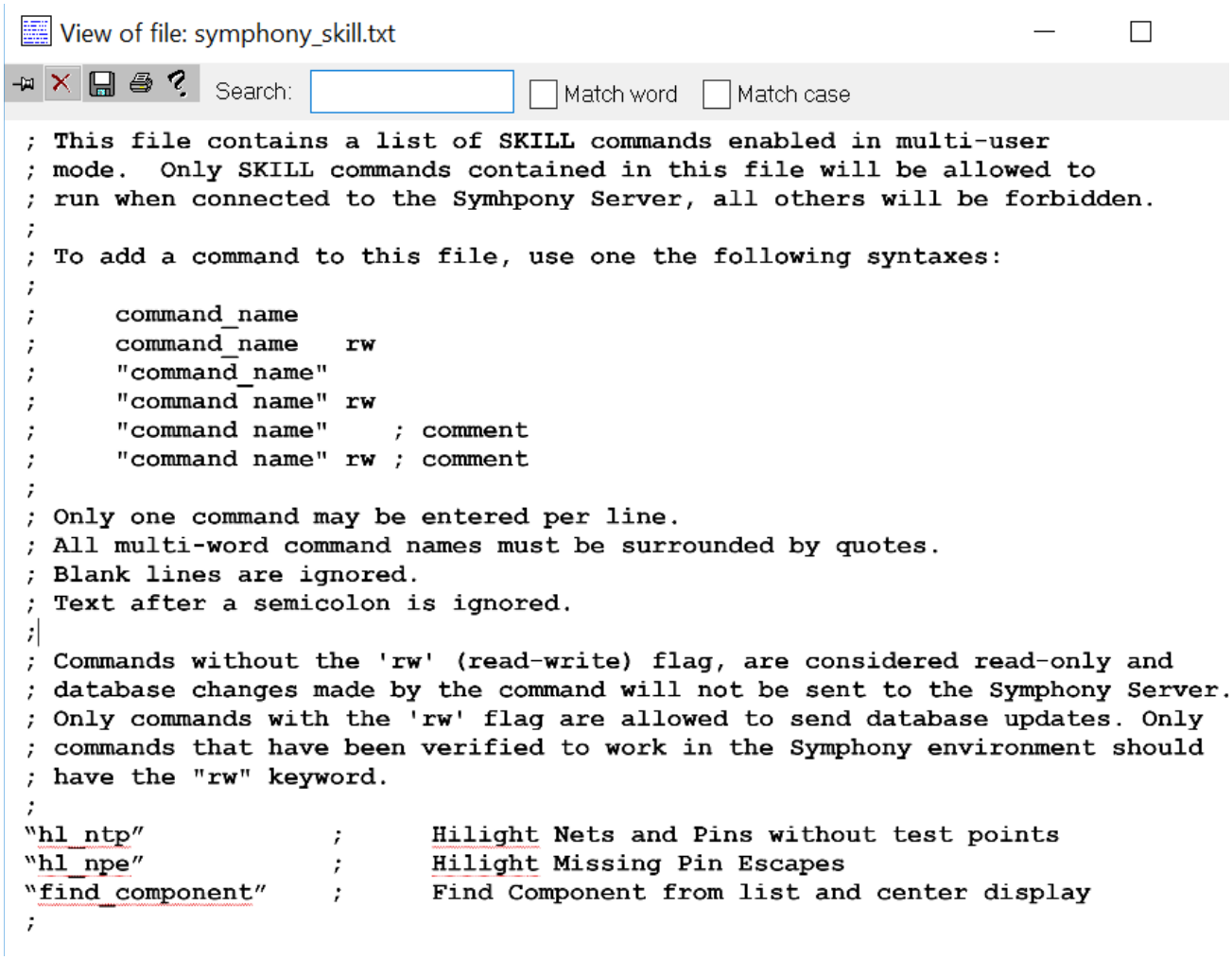
Commands using SKILL routines that perform database updates are not supported inside the concurrent design environment. To avoid accidental database changes by registered commands that execute SKILL routines are disabled at the clients end.

Read-Only SKILL Support

The SKILL routines that perform read-only operations and do not require database updates are safe to execute in the multi-user design environment.

A new configuration file `symphony_skill.txt` can be used to enable commands that perform read-only database operations at clients end. The default location of this configuration file is `PCBENV` directory. You also can manage this file at site level by saving it in the `CDS_SITE/PCB` directory.

Note: Connecting to a Symphony database automatically creates the `symphony_skill.txt` in the `PCBENV` directory. The header of the file describes the usage of the file.



Read-Write SKILL Support

The SKILL routines that perform actions are currently supported in the concurrent design environment can be updated to communicate with the Symphony server. Refer to Allegro User Guide: Symphony SKILL Reference for updating critical SKILL routines for the Symphony team design environment.

Routines modified to work in the concurrent design environment must add the registered command to the configuration file `symphony_skill.txt` with the `rw` flag after the command name. This flag allows the database updates made by the command to be sent to the Symphony server.

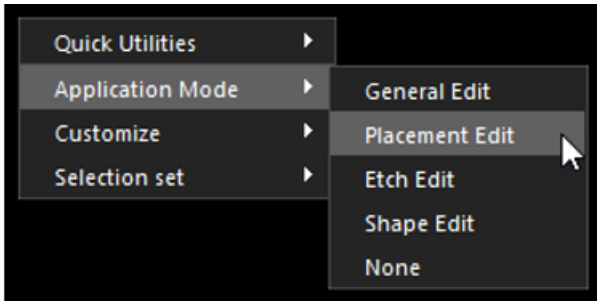
Note: Only commands with the `rw` flag are allowed to send database updates to the server.

Place Replicate Apply

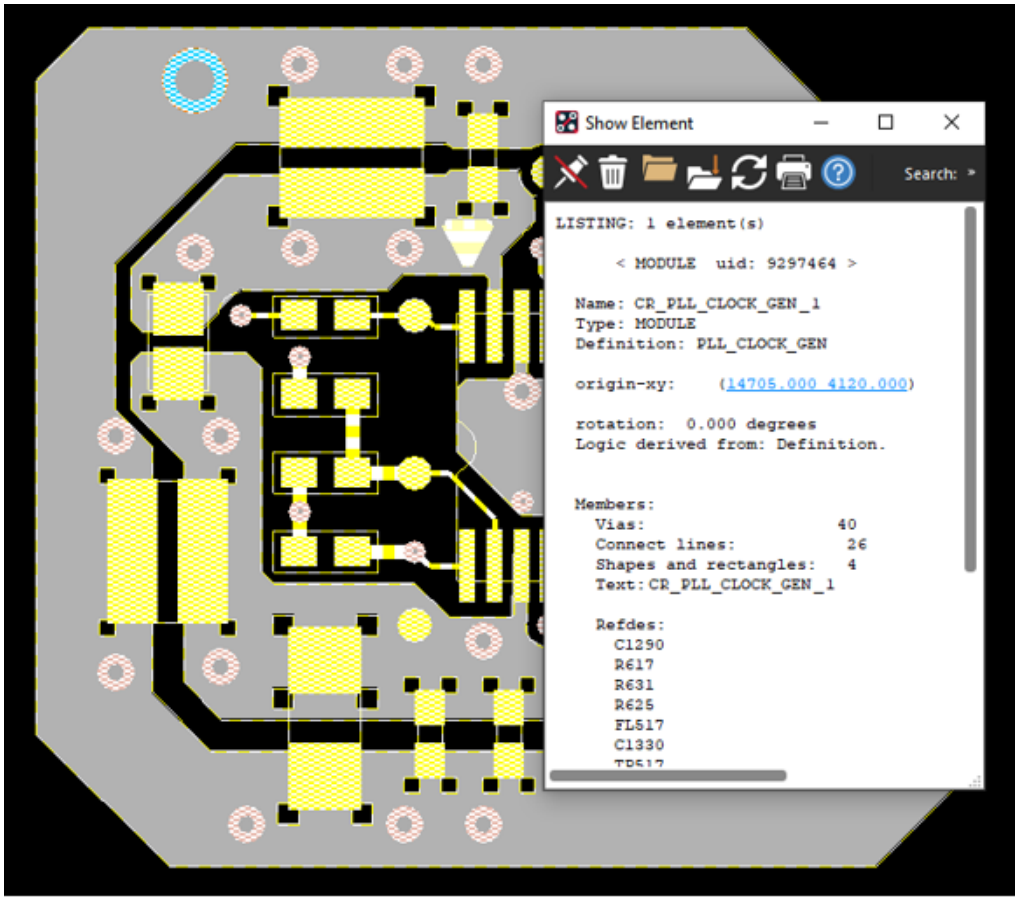
In the concurrent design environment, clients can create and share place replicate modules with other clients. The place replicate command no longer requires a local place replicate module definition database (`.mdd`) file to use the apply functionality. All the place replicate instances are available in the database to reuse.

Note: Modules containing padstacks that are not available in the database are not loaded in the concurrent design environment.

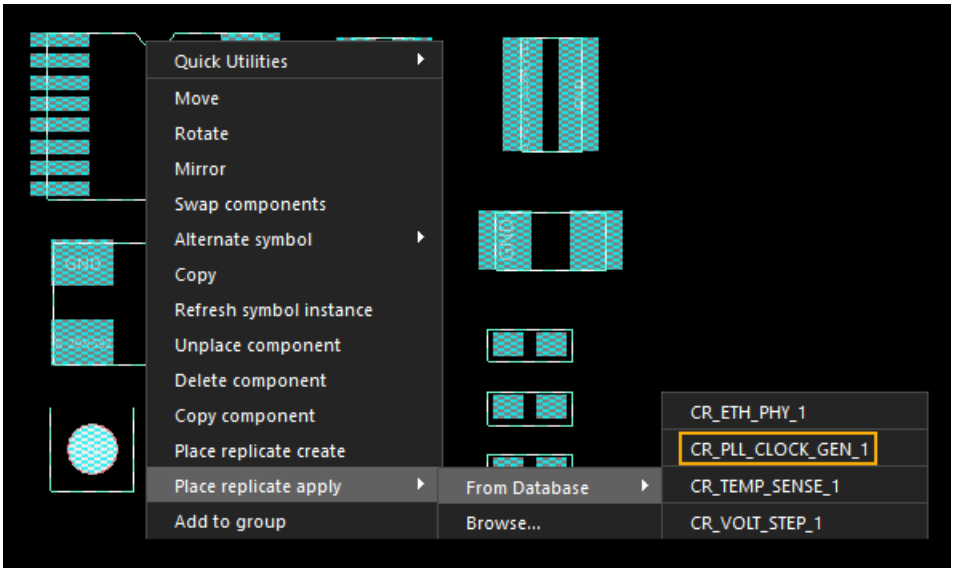
To apply place replicate modules, set application mode to *Placement Edit*.



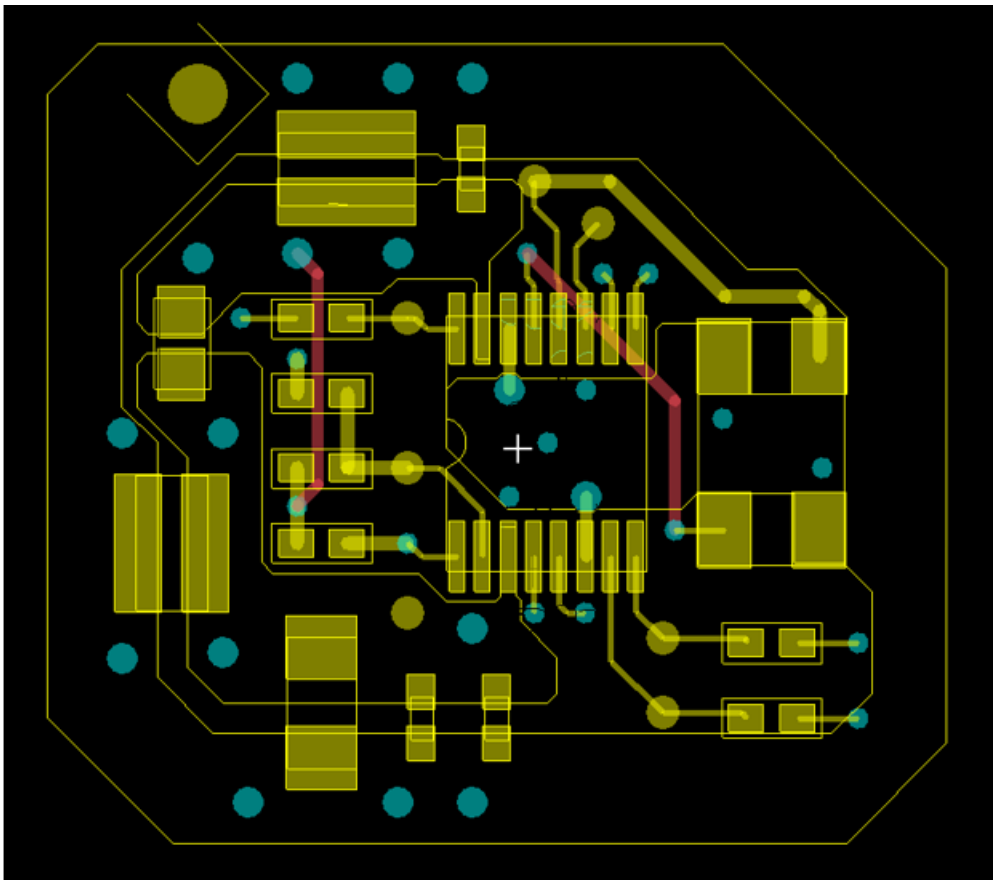
In the *Find* filter, enable *Groups* and run `show element` command for the selected place replicate group.



Window select the components and hover over one of the selected components. Select the place replicate module using right-click pop-up menu *Place replicate apply — From Database*.



The place replicate module is attached to the cursor for placement.



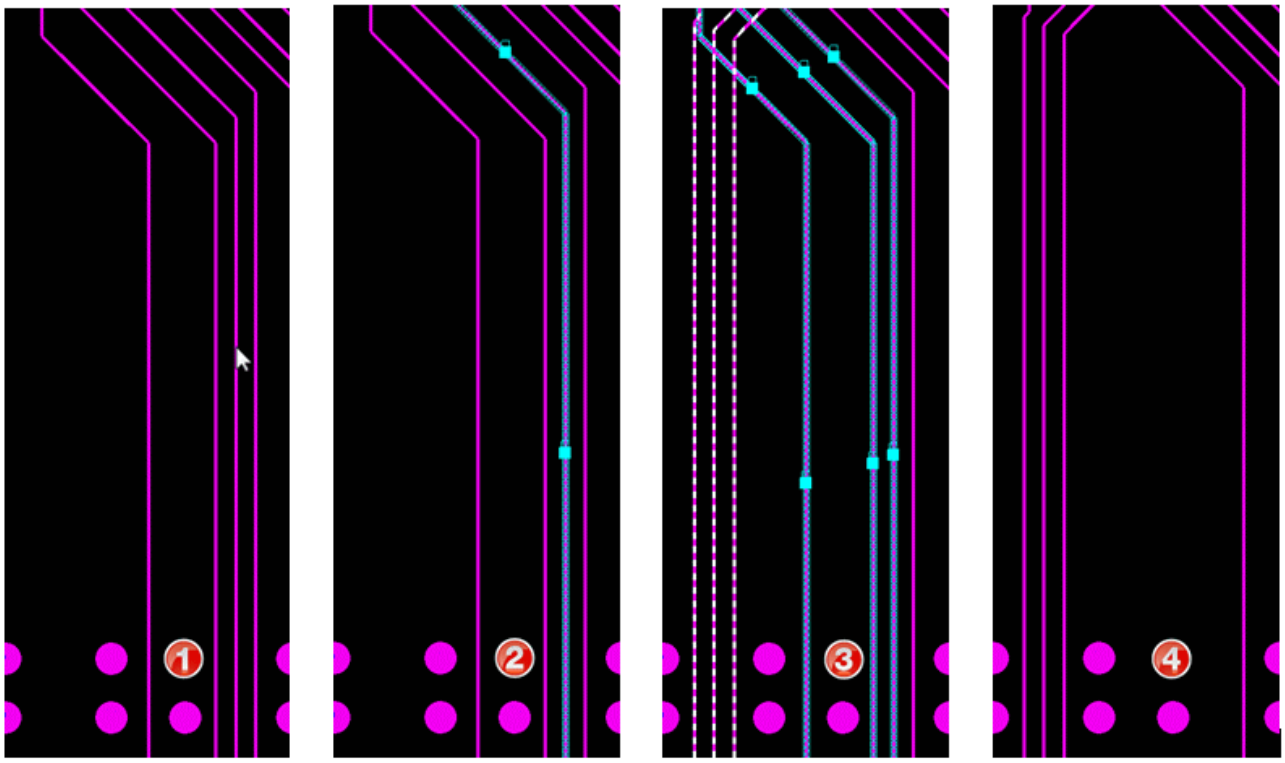
Multi-User Lock Display

The overall Allegro® Symphony team design environment is similar to running a layout editor, but there are some slight differences in behavior.

- Design updates are integrated to the server database upon selection of a new object or when the active command is completed.
- If a client is in an active command, updates that have already been integrated on the server, may be delayed.
- Activities performed by other clients generate a real-time temporary display in preparation for their updates to be integrated in the design.
- Multi-user locks are generated on the canvas as other client interact with design objects. This lock display is color-coded to a particular client for easy identification.

Example 1: Client performing a Slide operation and display as seen by other clients:

1. Client #1 selects a cline using `slide` command.
2. Other clients see the selected cline with temporary lock display color-coded to the client performing the slide. This prevents other clients from acting on the cline.
3. Client #1 selects a new location for the cline bubbling other clines along the way and generating temporary locks on other effected traces. A preview of the *Slide* results will be seen using the current etch layer color stripped with the temporary highlight color while in the `slide` command.
4. Client #1 completes `slide` command, changes are integrated by server, and final updates are dispatched to all clients.



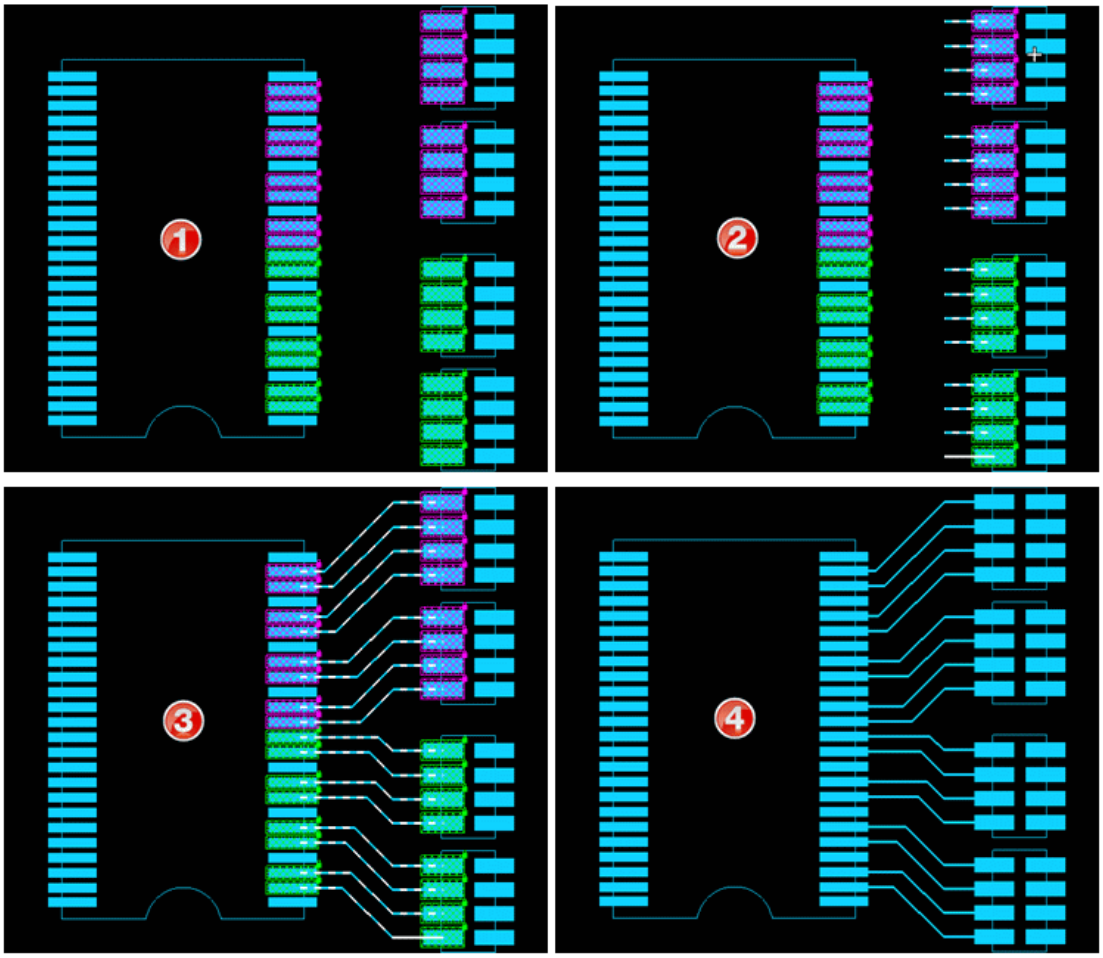
Example 2: Two clients perform an `add connect` ***and display as seen by other clients:***

1. Group routing from resistors to TSSOP IC using `add connect`.
   - ❑ Client #1 (Purple) selects the upper eight resistors connections to the IC.
   - ❑ Client #2 (Green) selects the lower eight resistors connections to the IC.

- Client #3 (Blue – shown in the following figure) will see the pin-pair connection with a temporary lock display color coded to the client performing the route.

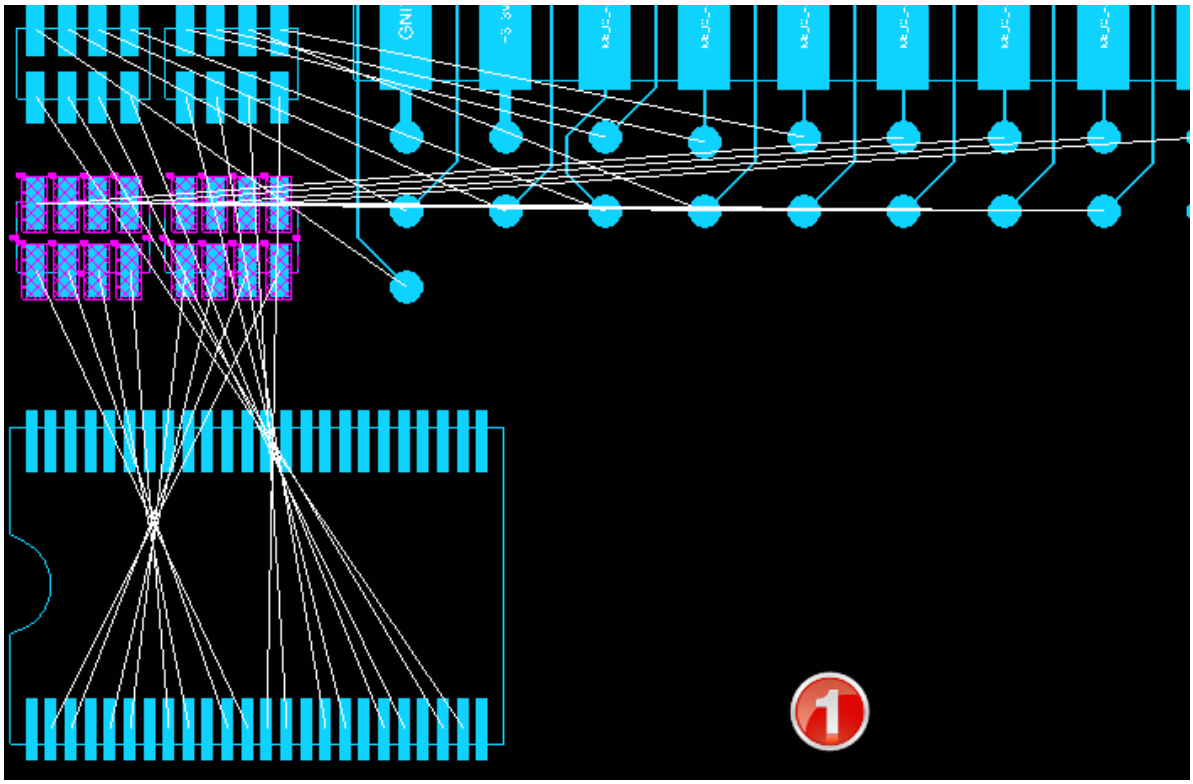
Note: Other clients see the selected pins with temporary lock display color coded to the user performing the routing. This prevents other clients from selecting the pins for routing the same connections.

2. All clients see a preview of the route using the current etch layer color stripped with the temporary highlight color.
3. After Client #1 and #2 complete the connections, right-clicking and choosing *Done* sends the update to the server for integration.
4. Server dispatches the final updates to all clients.



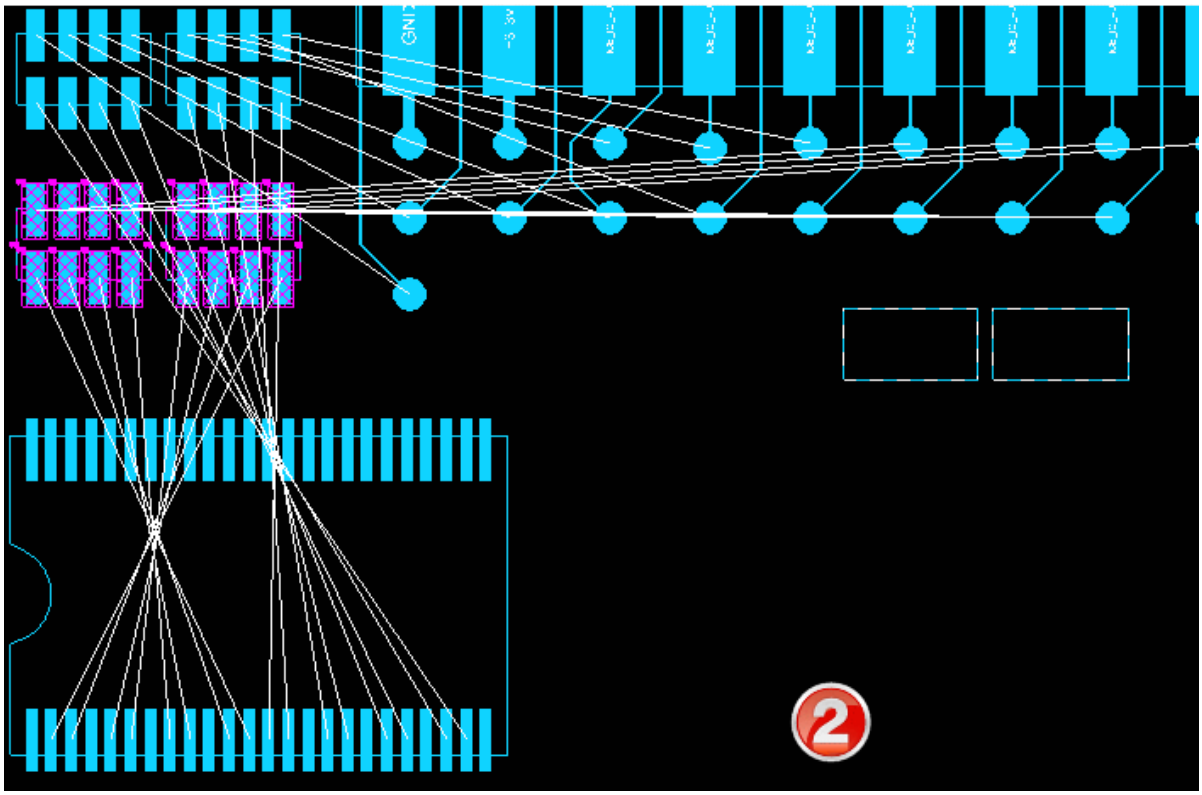
Example 3: Client performing a component `move` ***and what will be seen by other clients:***

1. As client selects a component for movement, other clients see the selected symbol(s) with temporary lock display color-coded to the user performing the `move` command. This prevents other clients from acting on the symbol.

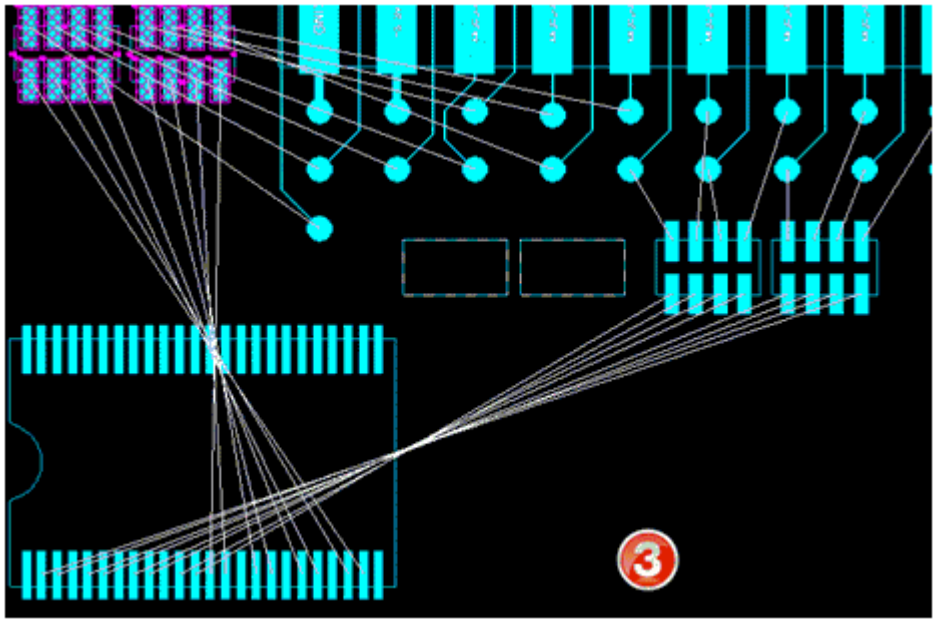


2. As the client selects a new location for the component, other clients see a preview of the placement using the visible *Package Geometry* subclass color stripped with the temporary highlight color.

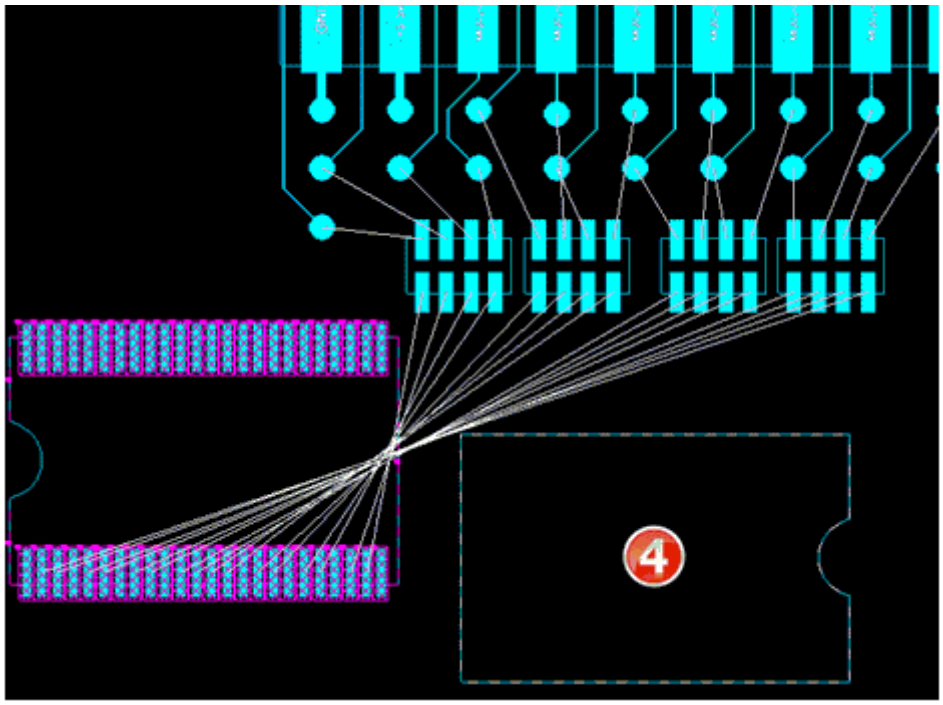
Any visible *Package Geometry* subclass is used as the temporary display of the new location. The example shows the *Assembly_Top* subclass visible on the other clients.



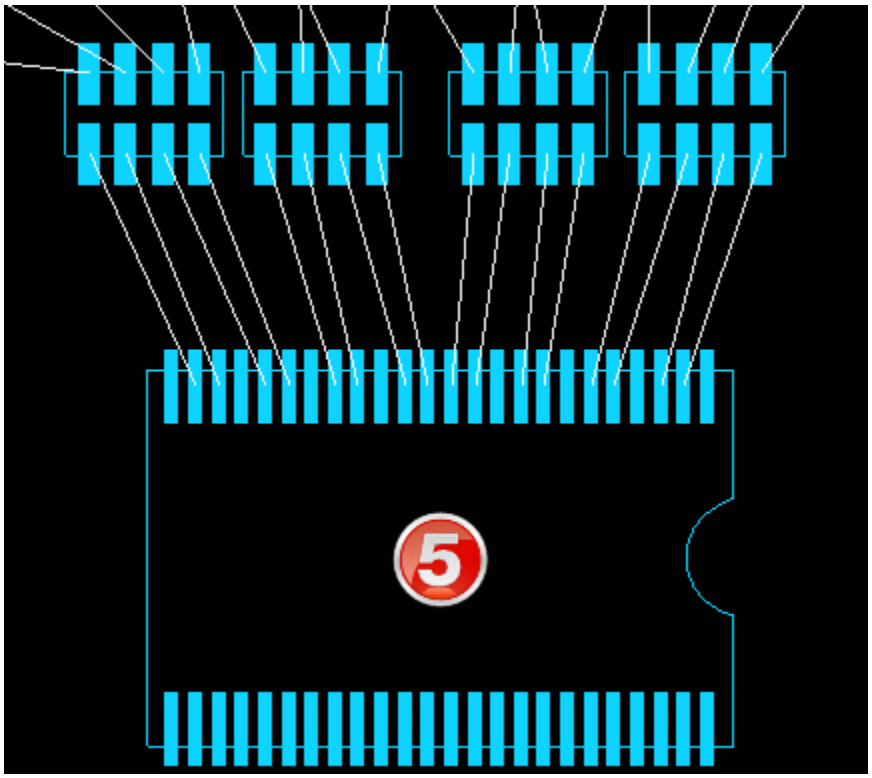
3. As client select a different component, updates are sent to the server for integration and changes are dispatched to other clients.



4. Temporary display lock is removed from the placed symbols and is applied it to the newly selected components.



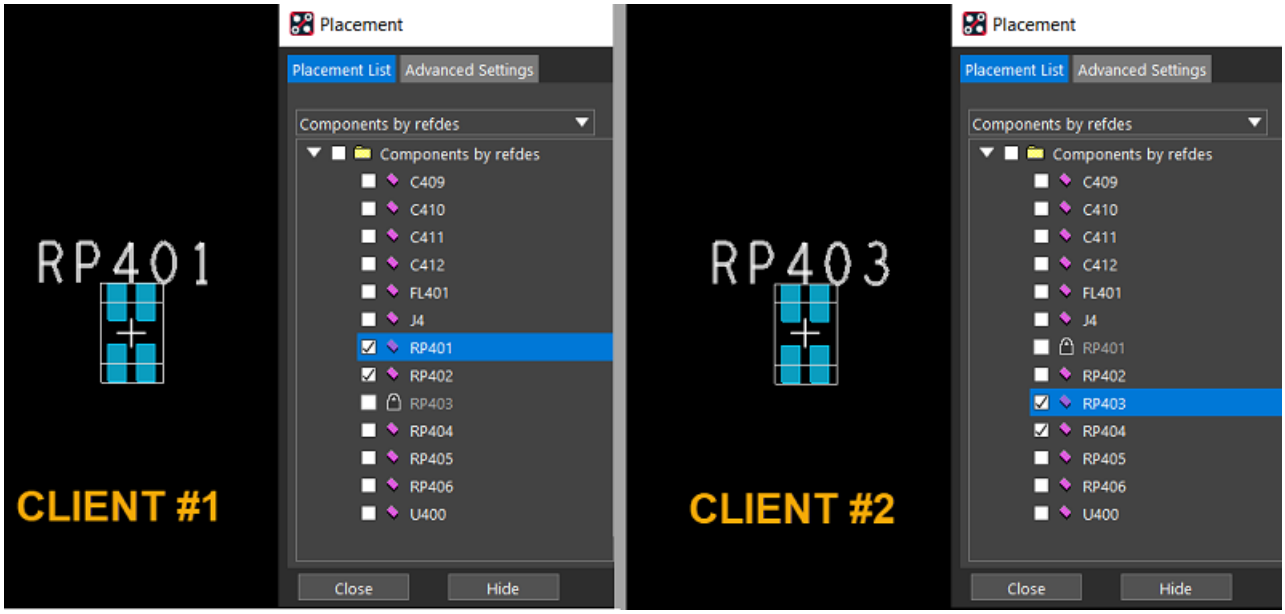
5. Placement rearrangement complete, ready for routing.



Example 4: Clients simultaneously performing component placement from unplaced component list:

1. Each connected client has access to the unplaced component list and can easily place component on canvas.
2. When a client is actively placing a component a lock icon appears on other clients preventing it from being selected for placement. This prevents two clients from placing the same component at the same time.
3. Selecting multiple components at the same time in the Placement dialog only generate locks for the component attached to the cursor or previously placed while in the

command. This prevents one client from selecting all components and locking placement for other clients.



Symphony Window – Locks Tab

This tab displays objects that have been locked by connected clients.

Permanent object locks explicitly set by the client to prevent other clients from interacting with the objects. Any client locks are cleared once they disconnect from the session.

```
Lock tab entry label = [Perm]
```

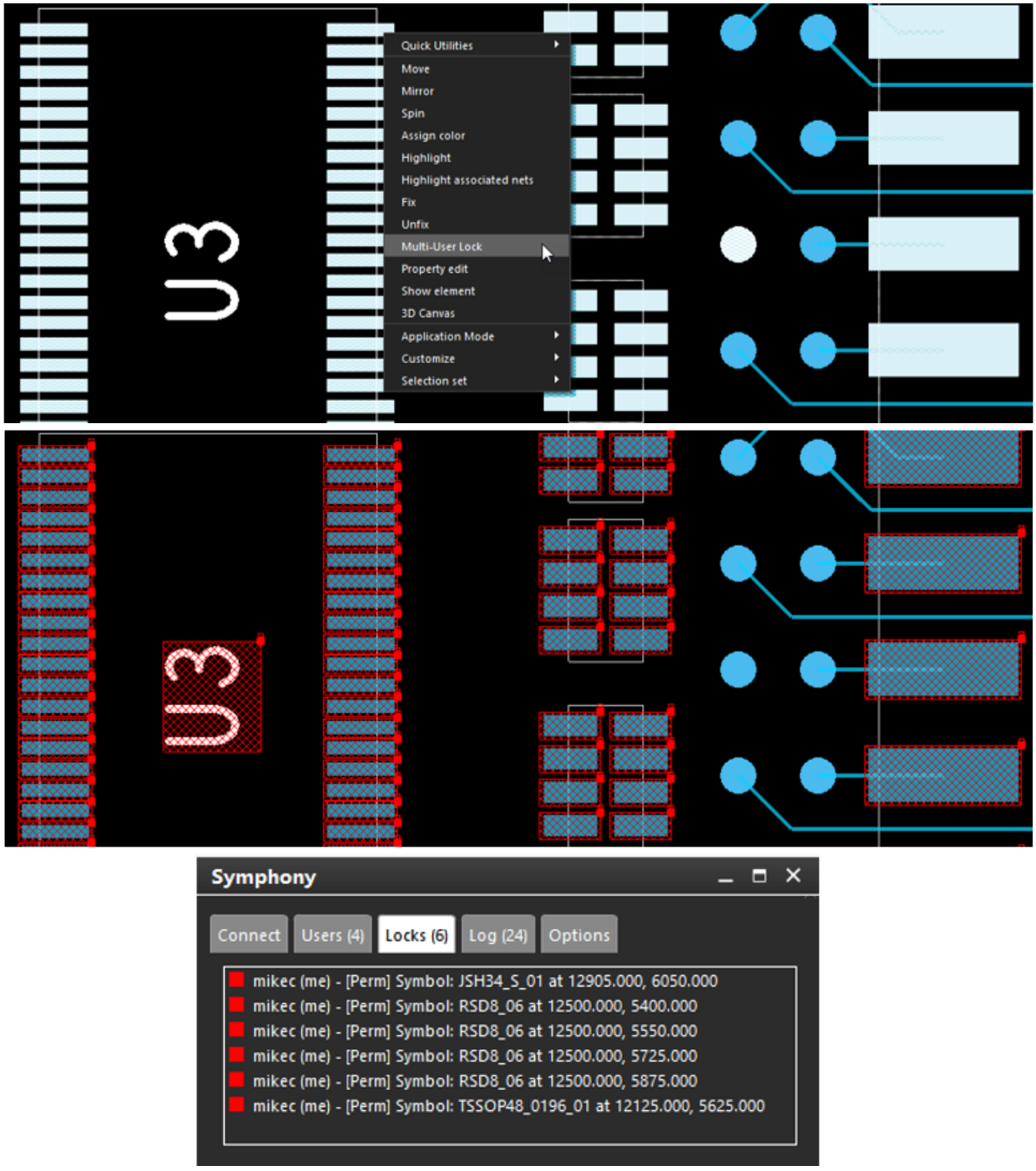
You can add permanent object locks as follows:

- Select object, right-click and choose *Multi-User Lock*.
- Choose *File – Edit – Multi-User Lock* and select objects.

To release permanent locks right-click and choose *Multi-User UnLock* or select *File – Edit – Multi-User UnLock* menu option. You can also release object locks from the *Lock* tab of Symphony server window.

Note: Locks are automatically released when the client disconnects from the Symphony server application.

In figure 3-1, select the object and choose *Multi-user Lock* from the context-sensitive menu. The object has been locked for multi-user editing.



The client who locked the objects still interact with the locked objects while restricted to all other clients.

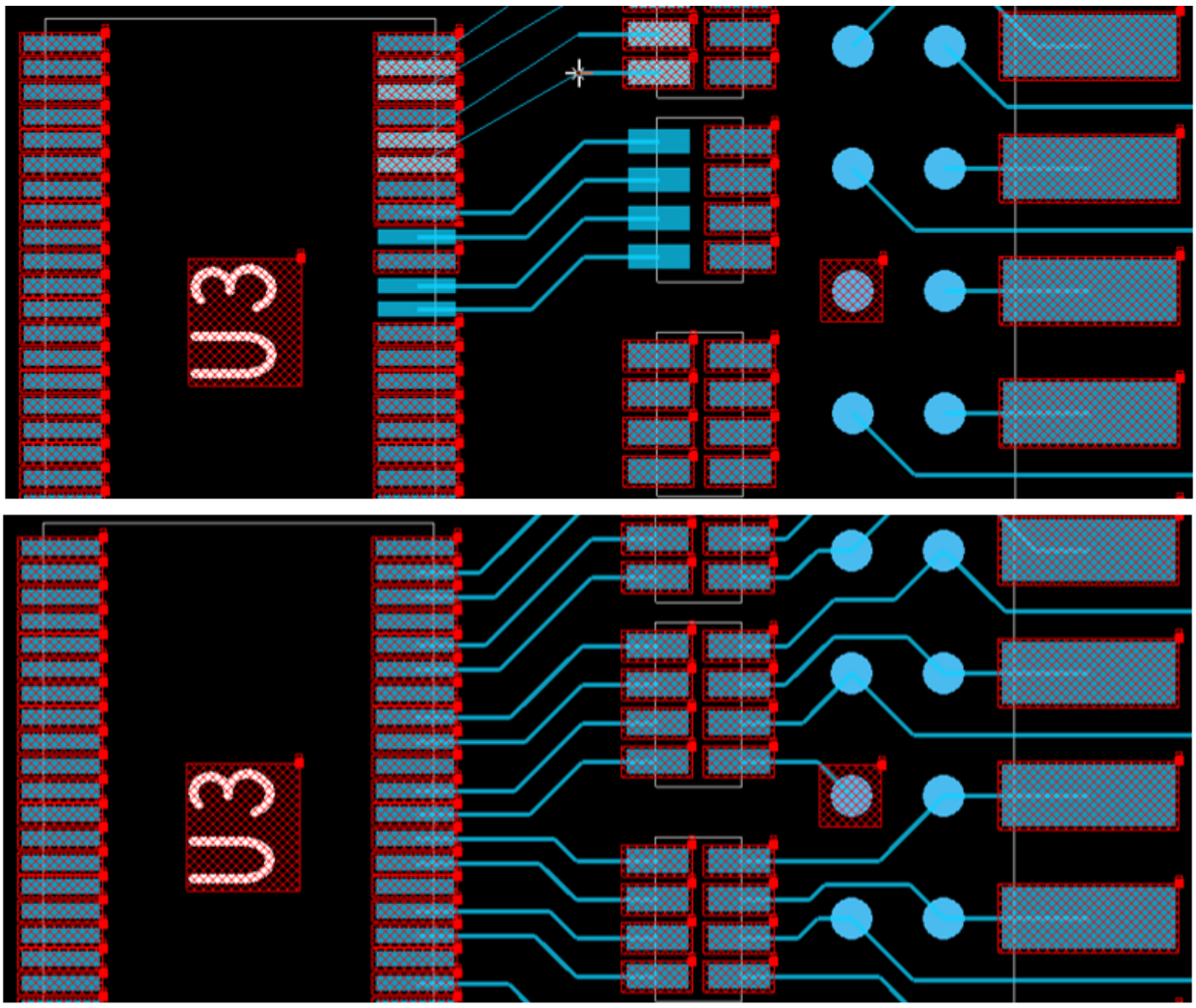


Figure 3-1 Permanent Locking

Temporary object locks are generated when a client is working on an object, and once the command is completed, the lock is released. This prevents multiple clients from acting on the same object at the same time.

```
Lock tab entry label = [Cmd]
```

In the figure 3-2, the pin is temporary locked while in `add connect` command.

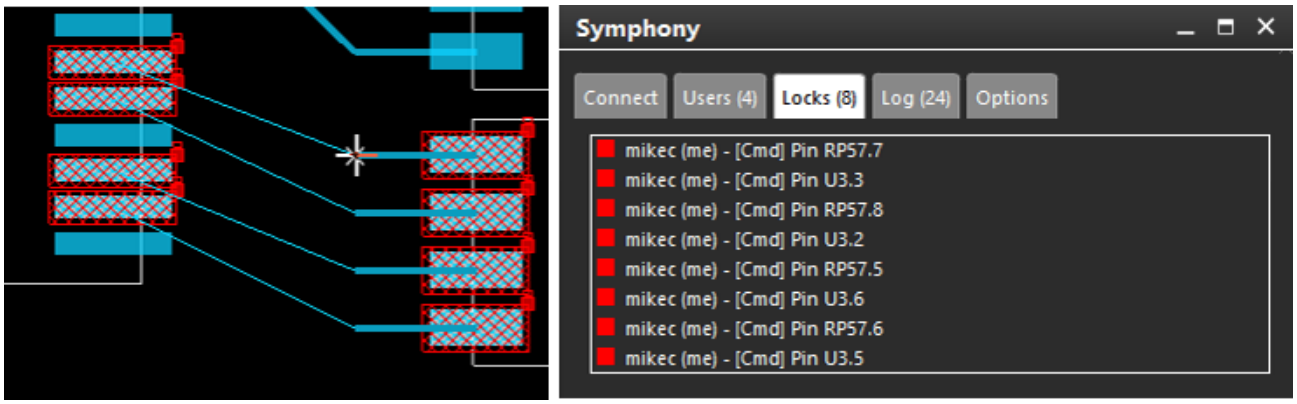
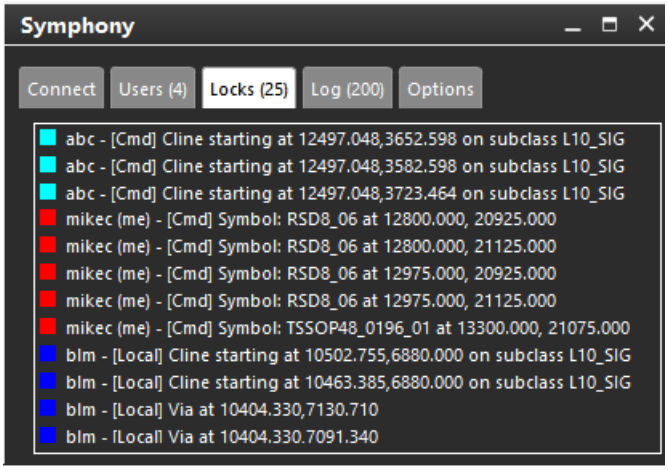


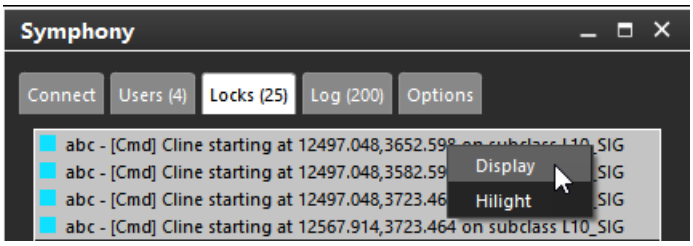
Figure 3-2 Temporary Locking

Local object locks are temporary object locks that have not been cleared on the local client while they are in an active command as it blocks server updates.

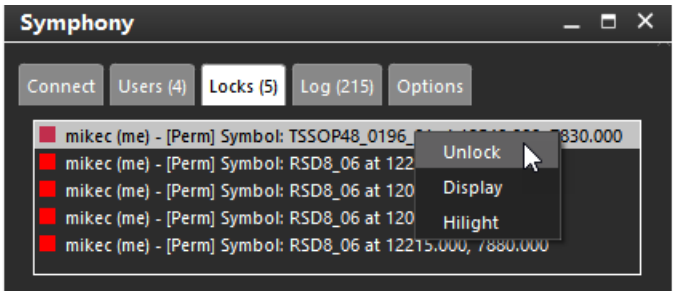
`Lock tab entry label = [Local]`



Double-click the client locked object or right-click and choose *Display* to re-center the display.



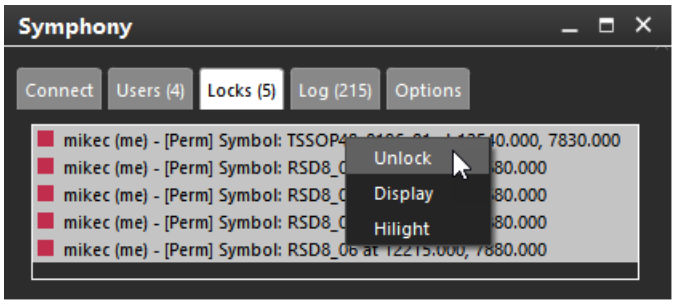
Right-click locked object and choose *Unlock* to quickly remove the lock.



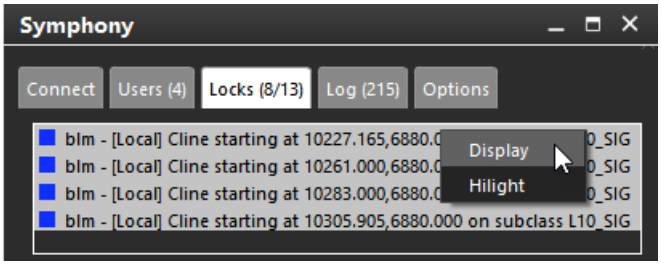
Note: You can only remove your own objects locks.

Tip

Select multiple rows by holding the control key or dragging across the rows. Right-click on selected objects to execute options.



Apply a temporary highlight or dehighlight locked objects using the right-click pop-up menu.

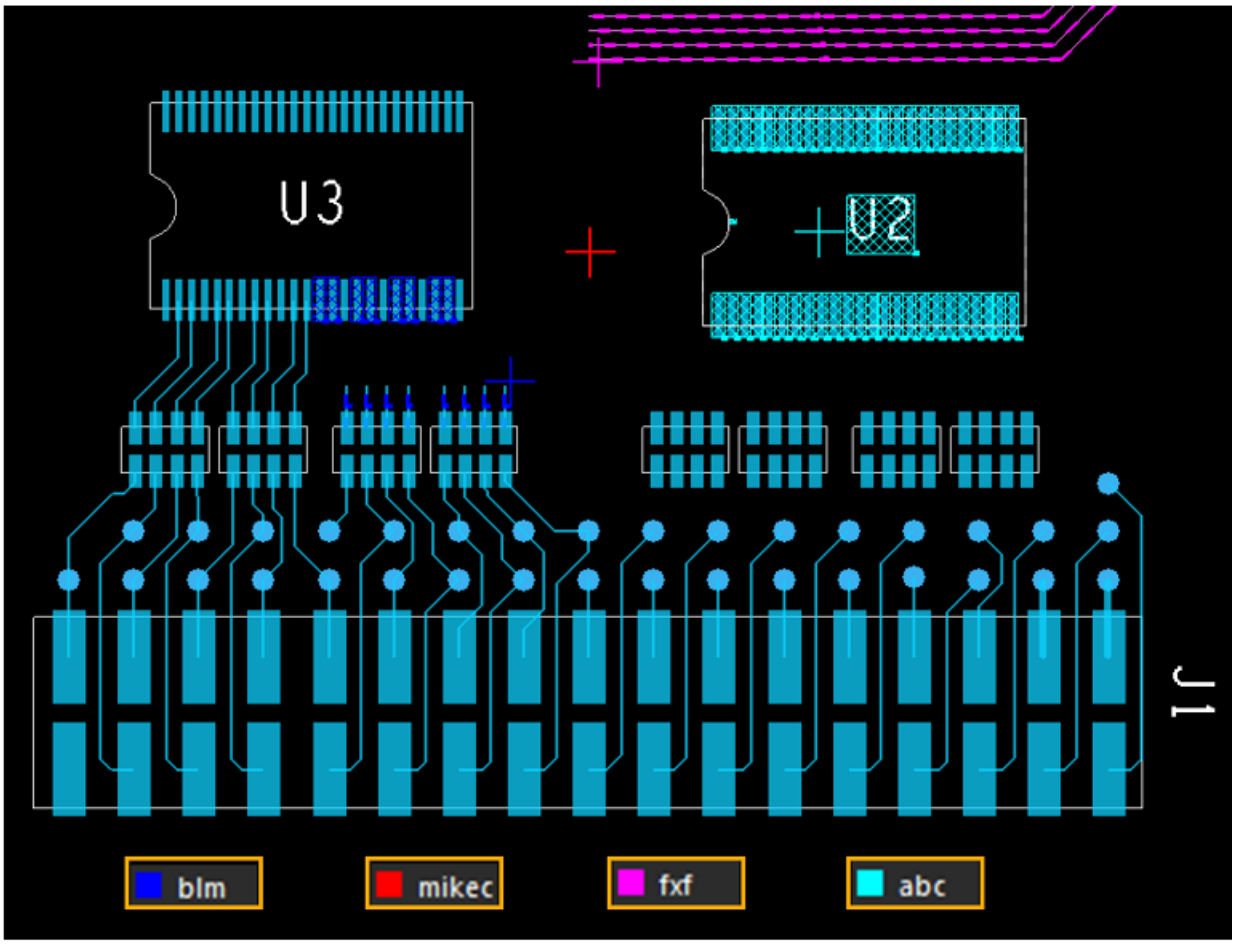


Multi-User Cursor Display

To provide visibility of the locations of the other clients in a Symphony session, the cursor locations can be displayed that shows their canvas movements in real time. Clients can control whether their cursors should display on the canvas of other clients or not. Clients can also selectively control the display of cursor of other clients on their canvas.

Clients cursors are color-coded. You can view the user-specific colors in the *Users* tab of the *Symphony* window for identification.

Cursors of other clients dynamically move around the canvas when they move their cursors and are color-coded based on color specified in the *Users* tab.



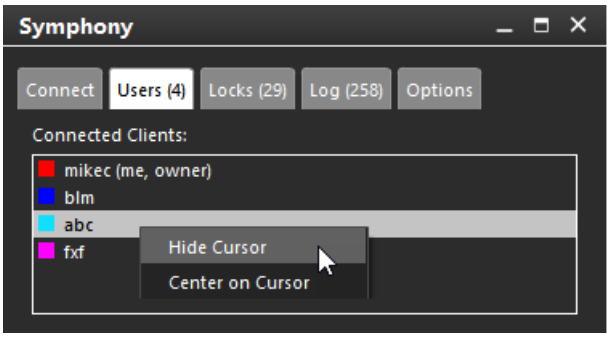
Symphony Window – Users Tab

This tab displays different users connected to the server and provide options for cursor settings on the pop-up menu.

- Show/Hide Cursor: Enable or disable the display of cursor for the selected user.
- Center on Cursor: Centers the canvas according to the cursor location of the selected user.

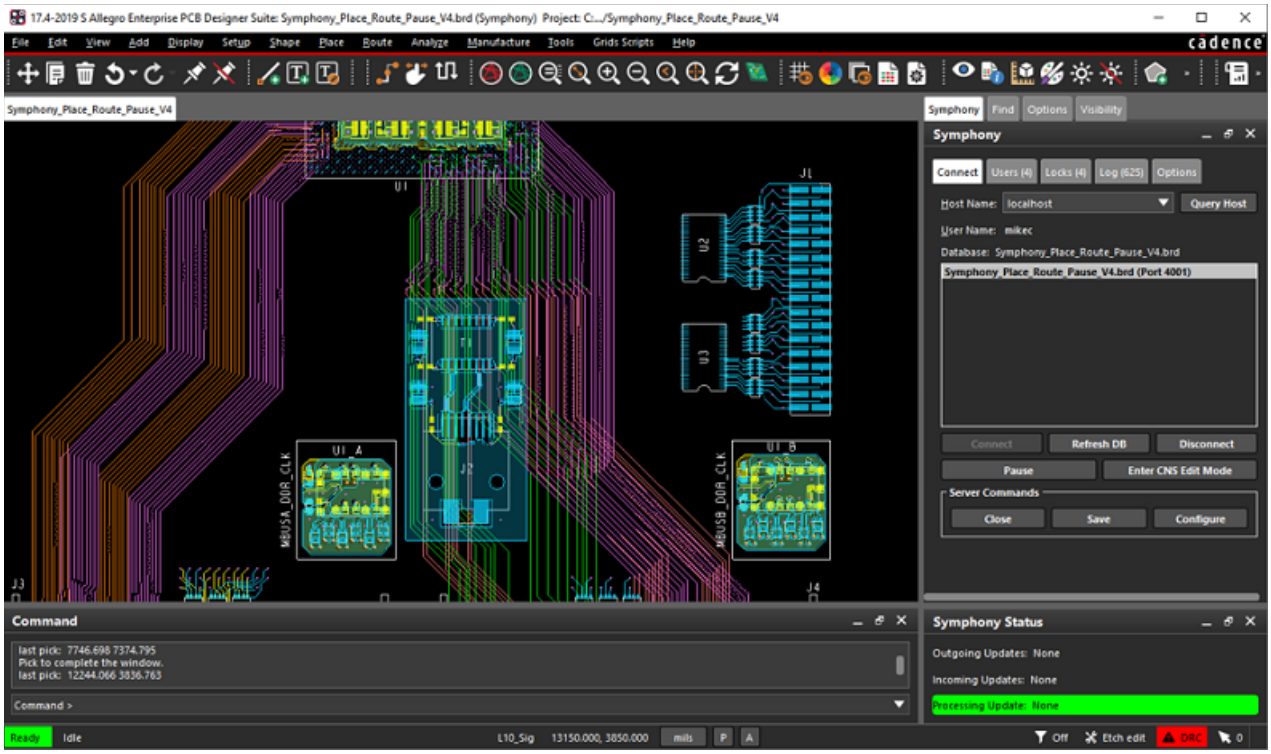
  Note: While centering the cursor, visible layers and zoom level will not automatically

adjust to match the other client.



Symphony Team Design Commands

Most of the commands and functionality have migrated from standalone layout editor into the concurrent environment. Review of remaining features will continue and updated to communicate with the Symphony server application, as required.

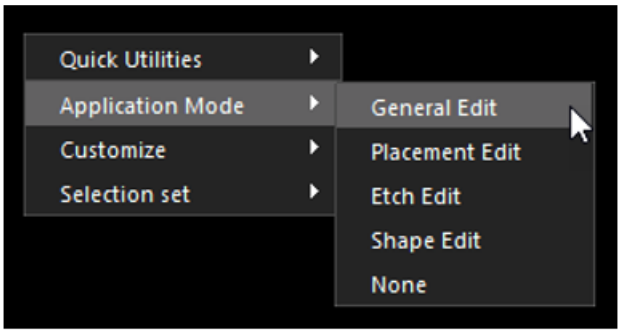


Supported Application Modes

Following application modes are available when working in Symphony team design environment:

- General Edit: Assists in general and placement activities.
- Placement Edit: Assists in components placement.
- Etch Edit: Assists in routing activities.

- Shape Edit: Assists in shape editing activities.



Supported Commands

Placement Tasks

- Placing components, mechanical and format symbols, and schematic-driven module instances: `place manual` (*Place – Manually*) and Placement Edit application mode

  Note: Placement of the first symbol instance in the design will be sent to other clients and the Symphony Server. The Symphony Server does not require library access, but it is recommended that each client has library access.

- For optimizing placement, including working with reuse modules and placement replica groups: `move`, `spin`, and `mirror`
- Swapping and aligning components during placement: `swap components` and `align components`
- Creating, updating and applying in-design reuse circuits created by other clients: `place replicate create`, `place replicate apply`
- Swapping pins: `swap pins`

  Note: Back annotation of pins swaps to the schematic is not supported in the concurrent design environment.

Interactive Routing Tasks

Routing supports following commands for:

- Manual routing: `add connect, slide, delete,` and `delay tune`
- Routing cleanup: `change, custom smooth, vertex`, and `delete vertex`

- Cutting clines, lines, and shapes: `delete by line` and `delete by rectangle`
- Moving and copying clines and vias: `move, copy,` and `change`
- Adding phase bumps to either member of the differential pair: `phase_tune`
- Creating bounding shape: `create bounding shape`
- Creating and copying fanout clines and vias: `create fanout` and `copy fanout`
- Spreading clines between adjacent plane layer voids: `spread between voids` (PCB Designer) and `spread clines` (APD+ and SiP)
- Create and Place Structures on the canvas: `create structure` and `place structure`
- Create structures using Parameterized based structure generator: `new structure`
- Structure placement during the Add Connect command with Return Path Net assignment

Integrated Analysis and Checking

- Analysis Flow: Performs analysis for electrical issues on the selected nets or the directed groups and reviews results tables and plots with cross-probing to canvas. *Visions* provide graphical overlay on routing layers for direct feedback. (This feature is available with High-Speed option).
  - Impedance Analysis: Runs an impedance simulation or load analysis results from a previous simulation, navigate to areas of interest with cline segment color coding on the canvas including enhanced datatips.
  - Coupling Analysis: Runs a coupling simulation or load analysis results from a previous simulation, navigate to areas of interest with cline segment color coding on the canvas including enhanced datatips.
  - Crosstalk Analysis: Runs a crosstalk simulation or load analysis results from a previous simulation, navigate to areas of interest with cline segment color coding on the canvas including enhanced datatips.
    - Requires Allegro Sigrity SI license
  - Return Path Analysis: Runs a return path simulation or load analysis results from a previous simulation, navigate to areas of interest with return current flow visualized on plane layers.
    - Requires Allegro Sigrity SI license

- Checking Flow: Reviews electrical issues based on design rules using *Visions* graphical overlay on routing layers for direct feedback.
  - Impedance DRC Vision: Cline segment color coding driven by impedance rules applied to the nets including enhanced datatips.
  - Return Path DRC Vision: Cline segment color coding driven by return path rules applied to the nets including enhanced datatips.

Auto-Interactive Routing Tasks

Auto-interactive routing supports following commands for quickly and efficiently completing bus and high speed routing. Auto–I interactive Breakout technology commands are available in the context menu in the Etch Edit application mode.

Note: Enable *Design Planning* product option for using auto-interactive breakout technology.

- Sequence, Breakout, Trunk Route, Trim to Breakout, and Delete Breakout

  Note: Breakout sequence is on a per client basis and not saved in the master database. Client sequence is cleared upon leaving the session.

- Auto Connect and Adjust Spacing (Trunk)

  Note: The `auto connect` command is not supported in APD+ and SiP.

- Finalizing routing with the guidance of Allegro® Timing Vision: Auto –Interactive Delay Tune `(aidt)`, Auto–Interactive Phase Tune `(aipt)`, and Remove Tuning `(detune)`

Shape Editing Tasks

Symphony team design supports following shape editing tasks:

- Editing shape boundary by sliding edges, adding a notch, joining edges, moving multiple segments, and chamfering corners
- Plowing and healing through dynamic shapes during routing and placement activities
- Updating shape instance parameters
- Dynamic and static shapes: `move, mirror, copy, change` and `spin`
- Improved server acceptance of shape updates
- *Dynamic Fill* mode *Fast* is available to improve dynamic shape performance

- Update all shapes to *Smooth* state before enabling *Dynamic Fill* mode *Fast* to maintain *Smooth* results during active etch editing activities.
- Overall performance improvements of processing updates are seen on the individual clients as well as the Symphony Server.

Note: The *Dynamic fill* mode changes and updates all shapes to *Smooth* state must be performed before sharing the design with Symphony Server.

Shape editing activities, such as deleting islands, changing shape type, compose/decompose shape, updating global parameters, shape degassing, and adding shape to generic group are not supported.

General Tasks

Common tasks performed by users in the Symphony team design mode for:

- Lines and text: `move, mirror, copy, change,` and `spin`
- Add line, arc, circle, rectangle, and text: `add line, add rarc, add arc, add circle, add rect`, `add frect`, and `add text`
- Edit text: `text edit`
- Generic groups such as Drill Legend Charts, Cross Section Charts: select and `move`
- Reports generation and Full/Window DRC Updates: `reports` and `drc update`

  Note: Waive DRCs are displayed, but DRC errors cannot waived.
- Copy and paste objects to multiple destinations that are automatically snap to the center of pins, vias and to dangling cline end points: `copy` and `paste.`
- Add or remove FIXED property to all the selected objects in single execution: `fix` and `unfix.`
- DRC Browser
  - Spreadsheet interface assists in locating DRCs, which dynamically updates when DRCs are removed or created.
- Read-only SKILL support
  - Allows clients to execute read-only SKILL commands through a configuration file that can be configured at user and site-level.
- Symphony SKILL reference guide

- Assist migration of SKILL routines into Symphony
- Multi-user API functions are available to ease code transition
- AXL-SKILL support for 44 functions to perform database changes
- Read-only support 571 AXL-SKILL functions

- Z-Copy shape support
- 3D Canvas support for
  - 3D visualization while connected to a Symphony session
  - Dynamic rigid-flex bending, cutting planes, and collision checking
  - Soldermask and silkscreens
- Edit property support for
  - Adding and modifying user-defined and standard object properties

  Note: Modifying properties that requires additional actions are not supported in Symphony team designing. For example, propagation delay.

  Note: You cannot edit properties on dynamic void in Symphony team design environment. (CCR#1892250)
- Create text files for net names, reference designators, or function designators: `define list`
- Import and export sub-drawing: `clpcopy` and `clppaste`
  - Import is limited to clines, vias, and shapes
- Capture canvas image: `capture image`

Note: Options that require separate licenses such as, RF-Structure Editing, Package Analyze, 3D Viewer, Advanced WLP and Advanced Package Router are not supported.

- Support for creating and changing color visibility view: `colorview create`, `colorview load`, and `colorview restore`
- Assign net to via: `via assign net`
- Symbol spreadsheet export
  - Enable the ability to export a pin number matrix of a symbol in an `XML`, `TXT` and `CSV` format for import into a standard spreadsheet tool.

User Interaction

- User settings for visibility, zoom level, preferences, and color assignment.

  Note: User-specific settings are preserved and applied when session is rejoined.

- User-specific color-coded object locking/unlocking on objects to reserve sections and to avoid team member conflicts: `mulock` and `muunlock`. Hover over datatip identification of lock owner.
- Client cursor location and tracking
  - Color-coded cross-hair tracking of clients movements
  - Show or hide cursor of all or selected clients
  - Center the display of the canvas according to specific client cursor

Manufacturing

For generating test points in the Symphony team design mode following commands are available:

- Manually add, delete, and move test points including test pad replacement and testing directly on a cline: `testprep manual`
- Scan and highlight untested nets and zoom the display to extents of a net so that test point additions can be made: `testprep manual`
- Add, delete or modify test point related properties on nets and symbols: `testprep properties`
- Set test point label display, selective padstack replacement and test point spacing requirements: `testprep prmed`

Symphony Server

Server application now supports:

- Pause session
  - Allows an exclusive single-client access to the server database to perform activities outside of the concurrent design environment such as import netlist, IDX import, pin swapping, and back annotation

  - Pauser gains full functionality of the layout editor. Other clients can work in view mode only until the session is resumed

- Multi-threaded communication architecture
  - Divide server-related tasks into multiple threads to improve server performance
- Filter server locks and logs.
- Removal of multi-user locks from the *Locks* tab of server application window.
- Constraint Edit Mode: Allows a single client exclusive access to the server database to perform Constraint Manager activates while everyone continues their concurrent design work.
  - Modifies constraints across all domains with full hierarchal support
  - Supports signal model creation and assignment for XNet generation and Integrated Design Analysis.
  - Topology extraction in SigXplorer along with Electrical Constraint Set definition.
- Technology Dependent Footprints
  - TDF-enabled databases can be opened and shared by the Symphony server application.
  - TDF mapping or symbol refresh can only be updated while session is in pause mode.
  - Dynamic switching of footprints by moving them between zones or by changing placement layer (Top, Embedded, and Bottom) is not currently supported.
- Symphony Server Manager
  - Remote management of dedicated server to start and manage Symphony server applications.