



WHAT IS NEW IN PSCAD/EMTDC V5.1.0

Since PSCAD/EMTDC v5.0.2

Date: December 1, 2025

To display “What’s New” documents for other PSCAD versions and other MHI products, please refer [here](#).

New Features	Enhancements to Existing Features	EMTDC/LCP/Master Library Improvements
Automatic Page Module Creation from Selection	Project-Based Fortran Compiler Settings	Subsystems Solved in Parallel
New Subsystems Definition Script Segment	Compiler Advanced Options	Improved Ideal Branch and Transformer Processing
C-Coder	Project-Based, Command Line Event Options	GPU Sparse Algorithm Enabled for EMTDC
Previous Run Playback	Project-Based MATLAB Settings	Multi-Rate (or Multi-Timestep) Transmission Line Upgrade
Component Library Viewing Pane	SCons Replaces Legacy Make/NMake Build System	New Photo-Voltaic (PV) Source
Component Templates	Simulation Speed Enhancements (PSCAD Side)	New Synchronous Machine with Simplified Input
Allowing EMTDC to Run Ahead of PSCAD (Live)	Transmission Lines in Blackbox	New Universal Induction Machine with Simplified Input
Automatic Output Channel Creation	Other Blackbox Enhancements	All Classical Transformers, Including Auto Transformers, Updated
Auto Module Interface Creation	Combined PMR/PNI Simulations	Frequency/Phase/Magnitude Meter Component
Auto T-Line/Cable Interface Creation	Single Dialog when Launching a Simulation Set Containing Manually Started Projects	New PID Controller Component
Component Wire Insertion	The Parameter Grid Goes Workspace Wide	New Hysteresis Controller Component
New Group Box Display Component	Other Parameter Grid Enhancements	New Over and Under-Excitation Limiter Components
Customizable Default Settings	Wire Mode Refined	New Power Factor and Reactive Power (VAr/PF) Controller and Regulator Models
Tag Definition Setting	Python Interpreter Upgrade	New COMTRADE Recorder Component
New Evaluation Functions for Conditional Statements	Additional Python-Related Updates	Single-Phase PI Section
New Resource Macros for File Names and Paths	Mirror/Rotate Restriction Control for Component Definition Graphic Objects	FDNE Component Upgraded to Use Optimized Matrix Storage
Drag Panning	Short-able XNodes	Moving Average Filter
Selectable IP Address	Easy Kill of Launched Simulation Processes	New Duality-Based Transformers
	New Option to Automatically End Simulation Set on Task Crash	New Pipe-Type Cables
	Conditional Statements in Flyby Script Segments	Three New PLL-Based Examples Added



	Instrument Properties	New Data Entry Format for the Permanent Magnet Machine
	Animated Data Labels	AC Exciters (2016) Update
	Component Ports May Now Be Defined as Constant	DC Exciters (2016) Update
	Application Options Import/Export to *.csv File	Static Exciters (2016) Update
	Miscellaneous Addressed Deficiencies	Power System Stabilizers (2016) Update
		RMS Option for External Control of Voltage Source
		Greek and Math Symbols Displayed in EMTDC Constant Tag
		Twist Component Compact View
		Sequencer Start and Wait Components Now Edge Triggered
		#COSIM Directive in Co-Simulation Component
		Current Measurement Added to RLC Branch Components
		Default Capacitive Branch Behaviour in Variable Impedance
		Graphical Display Indicates Conversion of Output to Nearest Integer
		MoM-SO Method Added



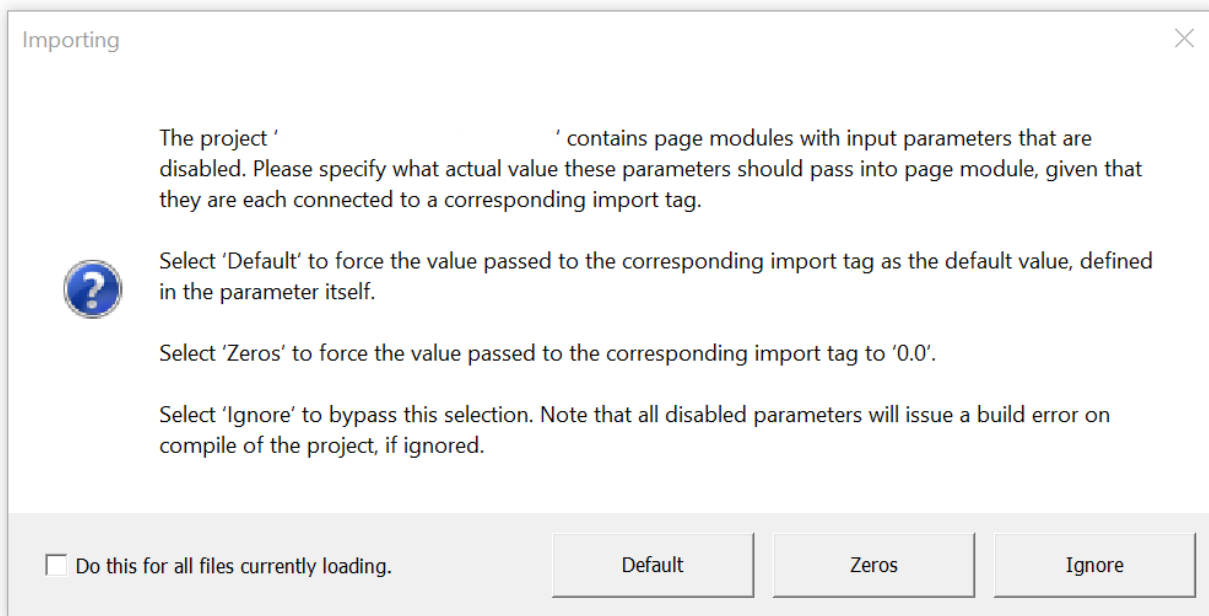
IMPORTANT NOTES ON IMPORTING FROM OLDER VERSIONS

Disabled Module Parameters (#9041)

The ability to add parameters to page modules was introduced way back in PSCAD v4.3.0 (circa 2010), along with the ability to disable these parameters.

Just prior to the release of PSCAD v5.0.0 in 2021, we identified the seriousness of disabling module parameters, as they are connected directly as inputs to actual circuits, drawn within the module schematic. If a disabled parameter is connected to an import tag on the schematic, which is likely an input to a controller or similar, what value should be passed into the import tag? The answer is important, as it can affect the accuracy simulation results.

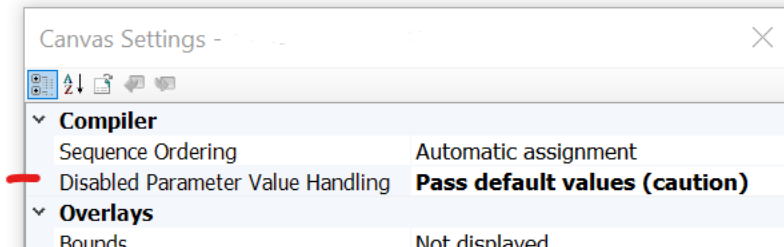
Our initial solution in v5.0.0 was to generate an error message when a disabled module parameter was encountered. We quickly discovered following the release that some users were using the disabled module parameter feature to organize their parameters visually, and as such were inundated with error messages on compile of their projects. We provided a quick solution to this by simply changing these messages to warnings in v5.0.1 and onwards. However, this was not a good solution and just masked the issue, so we came up with a better solution for v5.1.0.



When a project is imported into PSCAD v5.1.0, it is scanned for disabled module parameters. If encountered, the above dialog is presented on a per project basis. Please take care in selecting one of the three options provided.



If you need to go back and change the settings after import, it may be done on a per module basis, via the canvas settings:



Fortran Compiler and Operating System Support Changes: What's in and What's Out

As per usual, we will be adding support for the latest Fortran compilers and Operating Systems and removing support for the oldest versions.

Officially Supported Fortran Compilers and Operating Systems:

- Windows 10 and 11 (64-bit Only)
- Windows Server 2008 R2 SP1 or better, Windows Servers 2012 / 2016 / 2019 / 2022
- Intel® Fortran Composer XE for Windows 12.1+, 13.x, 14.x, 15.x
- Intel® Parallel Studio XE for Windows 16.x
- Intel® Fortran Compiler for Windows 17.x, 18.x, 19.x
- Intel® Fortran Compiler OneAPI for Windows 2024, 2025
- GFortran (8.1, 13.2)

No Longer Officially Supported:

- Windows 7
- Intel® Fortran Composer XE for Windows 12.0
- GFortran 4.6.2

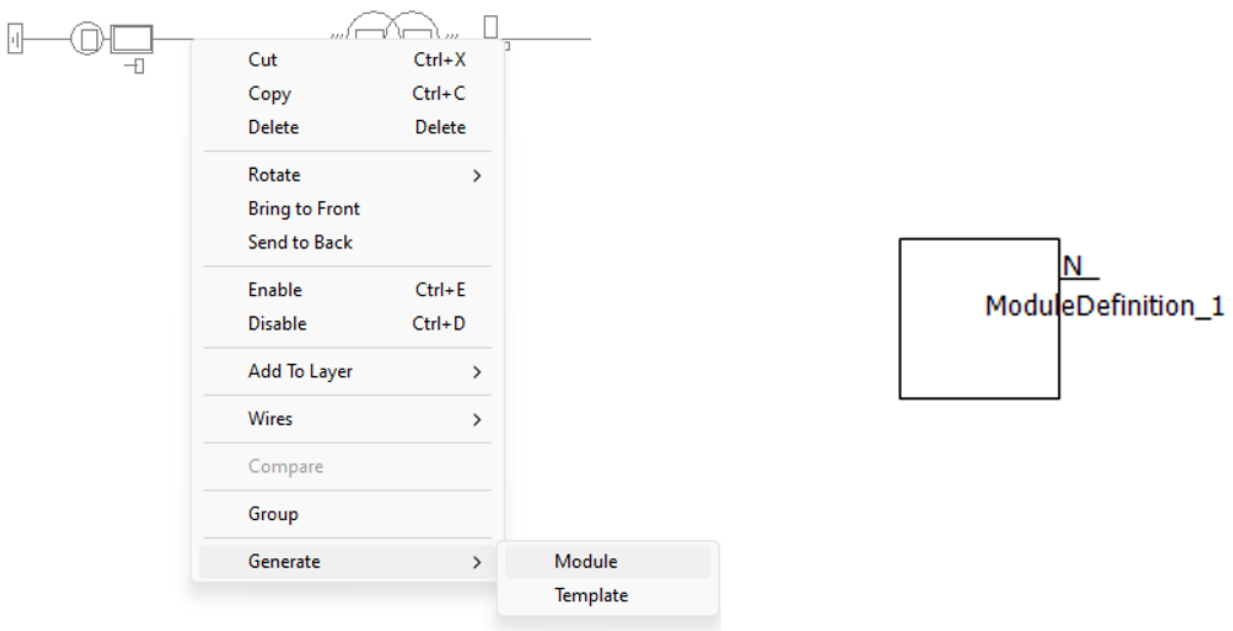


PSCAD

New Features

Automatic Page Module Creation from a Selection (#236)

Users may now collapse any selection of schematic components directly into a page module. Simply select one or more components, using any of the available selection methods, right-click and select *Generate / Module*. The algorithm will consider both hard-wired and wireless signals that cross the selection boundary. Wireless signals include both those from data and node labels from/to output/input parameters, as well as #OUTPUT directive signals, generated within components, that are passed out via text parameters. If necessary, new parameters will be added to the generated module component to accommodate any wireless signals. All in all, the module component should be run ready once generated.



New Subsystems Definition Script Segment (#7719)

A new script segment entitled *Subsystems*, is now available for use in all, non-module component definitions. New script directives (#SUBSYSTEM, #ENDSUBSYSTEM, #PORT, #TLINE, and #ENDTLINE) allow users to define two or more subsystems and declare which components ports belong to which subsystem. A section encompassed by the #TLINE/ENDTLINE directives, houses the contents of the solved *.tlo/*.clo file (generated by the PSCAD Line Constants Program), along with specifications for port connections.



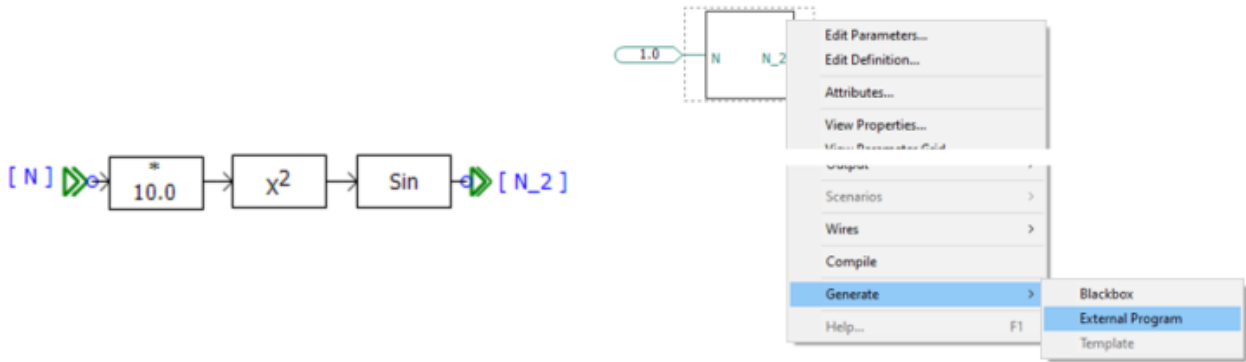
```

1 #SUBSYSTEM SS1
2 #PORT N, NBBF1
3 #ENDSUBSYSTEM
4
5 #SUBSYSTEM SS2
6 #PORT N_2, NBBF2
7 #ENDSUBSYSTEM
8
9 #TLINE OUTPUT SEGMENT_0 SS1(NBBF1(4), NBBF1(5), NBBF1(6)) SS2(NBBF2(1), NBBF2(2), NBBF2(3))
10 3 0 1 1 60.00000 60.00000 100000.0 0.1000000E-08 0.1000000E-08 /
11 1 1 0.498991943323560 651.480263770966 0.371477753870379E-03 0.371477753870379E-03 0.577525920918064 0.707106781186552 -0.418034447512749
12 2 2 0.342410766460306 318.322113016821 0.187492671522948E-04 0.187492671522948E-04 0.576998805315474 -0.876933448581134E-14 0.806532331270984
13 3 3 0.338634082834414 267.045965961836 0.191764641425243E-04 0.191764641425243E-04 0.577525920918064 -0.707106781186543 -0.418034447512765
14 #ENDTLINE
    
```

C-Coder

PSCAD may now be used to generate a fully formed, ANSI-standard, C language program, which may be run independently of EMTDC. The resulting code can then be used to program micro-controllers, etc.

C-language program generation is accomplished by constructing a control circuit using a collection of our more basic, master library components, specifically from the *Control System Modeling Functions* (CSMF) section. Once constructed inside of an encapsulating page module, the C-language program is generated with a single click.



To facilitate this functionality, we have added a brand-new component definition script segment, called 'C', where we have placed C-code script that is functionally equivalent to the component's original Fortran-based script. The 'C' component script segment is only ever considered when generating a C-language program.

```

monostable (Definition)
1 #STORAGE REAL:4
2 #SUBROUTINE EMTDC_XMSTAB 'Monostable Multivibrator /w Interpolation
3 /
4 #IF INTR==0
5 #LOCAL REAL RVD2_1 2
6 #LOCAL REAL RVD2_2 2
7 RVD2_1(1) = FLOAT($in)
8 RVD2_1(2) = 0.0
9 #ENDIF
10 CALL EMTDC_XMSTAB($INTR,$T,
11 #IF INTR==0
12 +RVD2_1,RVD2_2)
13 $out = NINT(RVD2_2(1))
14 #ELSE
15 +$in,$out)
16 #END
    
```

Fortran Segment

```

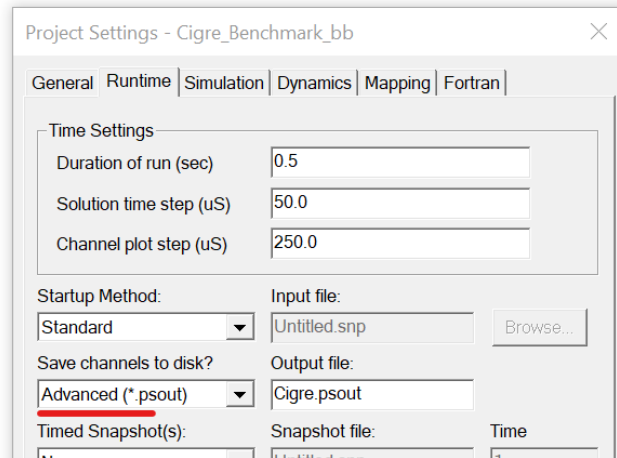
monostable (Definition)
1 #INCLUDE $(HomeDir)\CCoder\stdinc.pds
2 #STASH STRUCT $xmstabData data
3 #STASH STRUCT $xzcdData0 xzcdData
4 $data.pulse = $T;
5 #IF $INTR==1
6 #WARNING Interpolation on input is turned on for this modulator.
7 #ELSE
8 $data.in = $in;
9 $pscadlib_xmstab(&($data), &($xzcdData));
10 $out = $round($data.out);
11 #ENDIF
    
```

C Segment

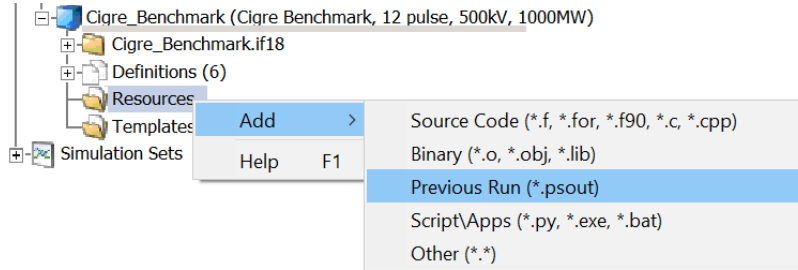


Previous Run Playback (#5948, #6781)

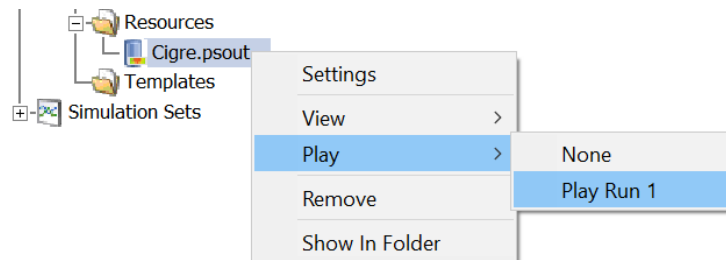
Previous Run Playback enables a replay of a previously run simulation, where the output results have been written in *.psout format (i.e. *Save Channels to Disk?* Is set to *Advanced (*.psout)*).



Once the simulation is complete, simply add the resulting *.psout file as a *Previous Run* project resource:

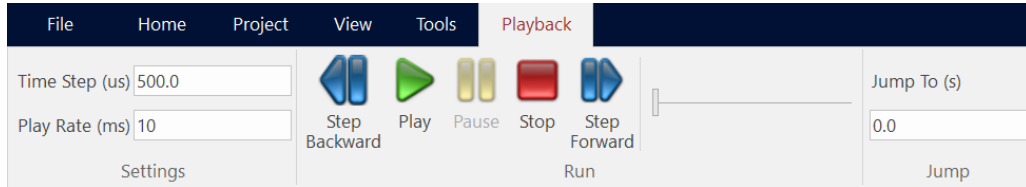


Once added, right-click again and select *Play* and then the run stored within the *.psout file you would like to play back:





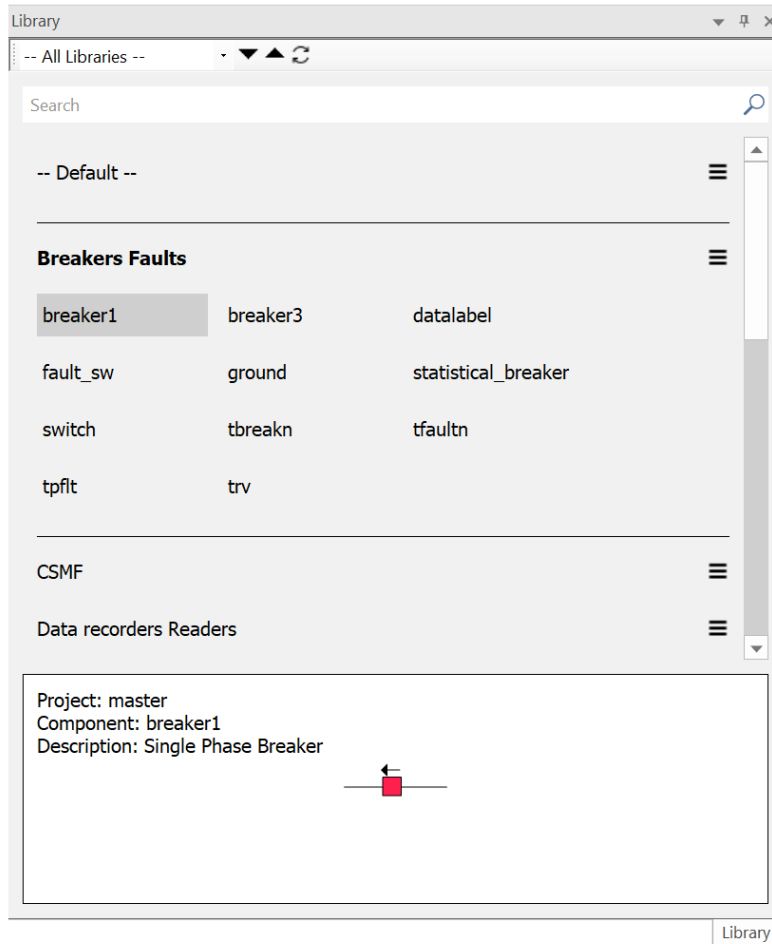
The playback speed is adjustable. The graphs, controls, instruments, and animations will be re-displayed during the playback. The ribbon control bar will automatically display a Playback tab, which provides an assortment of controls:



Previous Run Playback is useful for doing presentations where performing the simulation may take a long time or perhaps the simulation is too fast for in-simulation analysis.

Component Library Viewing Pane (#6409, #8592, #8595)

A new viewing pane has been added that provides a searchable environment for finding and viewing available component models, across the entire workspace.

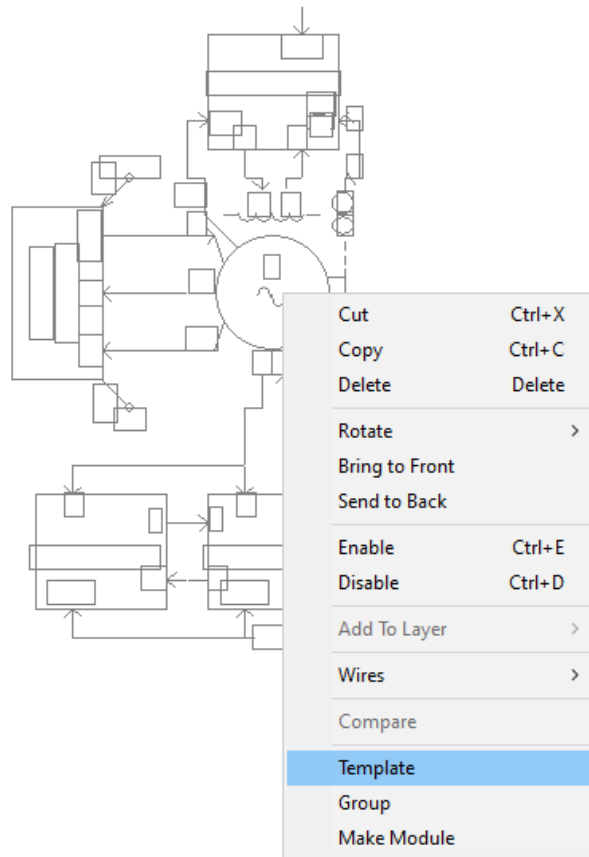




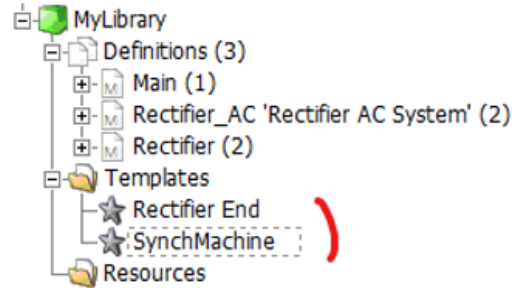
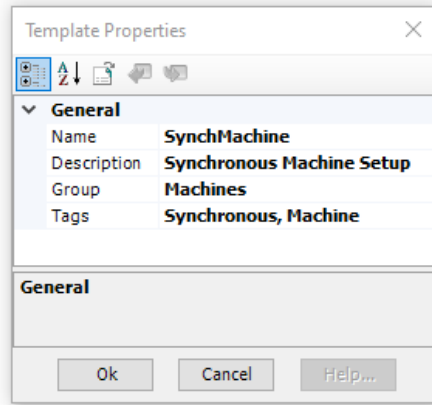
Filters provide the ability to focus the view on a specific project, or across all loaded projects in the workspace. Fully searchable, users can quickly find what they are looking for and add an instance of the component to the schematic with a single click. A viewing window at the bottom of the pane provides information about the component, including its graphical appearance.

Component Templates (#10043)

Pre-configured collections of components and modules may now be consolidated and stored as a *Template* object. Functional in both library (*.pslx) and case (*.pscx) projects, simply select a group of components on the schematic canvas, right-click and select *Generate | Template*.



Once templated, the pre-configured group of components are stored in the project under the *Templates* branch in the workspace tree:

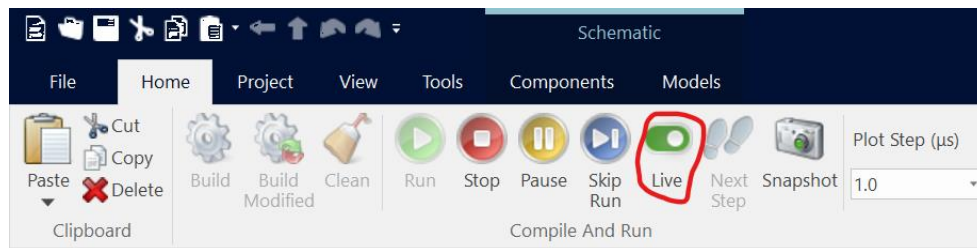


Once created, the template becomes viewable and searchable (according to its name and given *Tags*) via the *Component Library Viewing* pane. An instance of the template can be created either from the *Component Library Viewing* pane, or directly from the workspace tree.

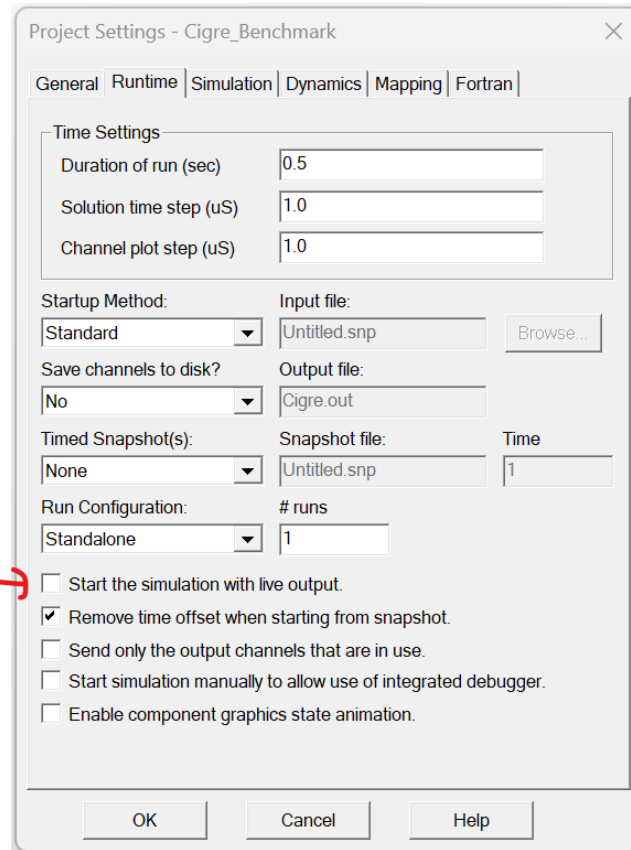
Allowing EMTDC to Run Ahead of PSCAD (Live) (#9654, #10044)

Starting in PSCAD v5.1, EMTDC by default will run unfettered to PSCAD. That is, EMTDC is not obligated to wait for PSCAD to process data it sends back for plotting, etc., before proceeding to the next time step. This increases the simulation speed, and along with other speed optimizations, helps EMTDC to run faster than it ever has before, especially for very parallel simulations.

The only instance where perhaps EMTDC running ahead would not be desired is when there is a necessity to manually modify online controls, such as sliders and switches, where it is important for PSCAD to be in synch with EMTDC, so that the manual control signal variations occur at the proper time. As such, a means to disable EMTDC running ahead is provided in PSCAD via a ribbon button called *Live*.



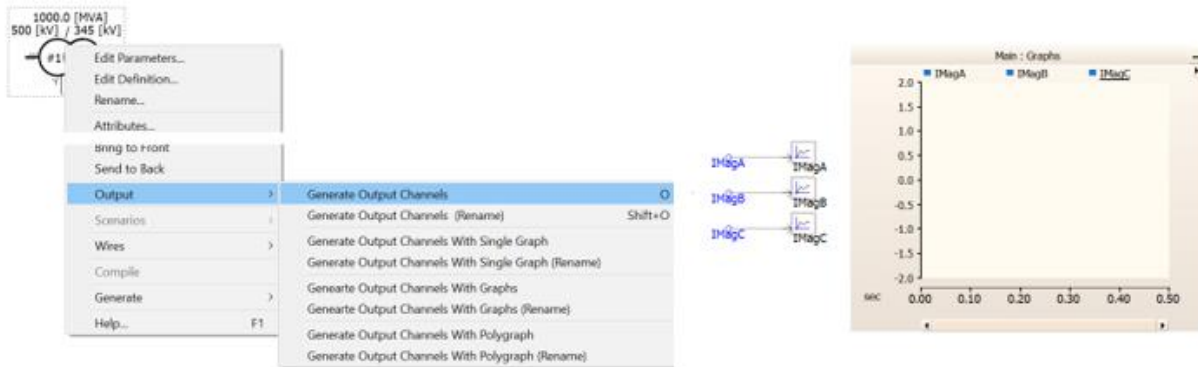
Note that the Live setting is also a project setting, that can be saved as always on:



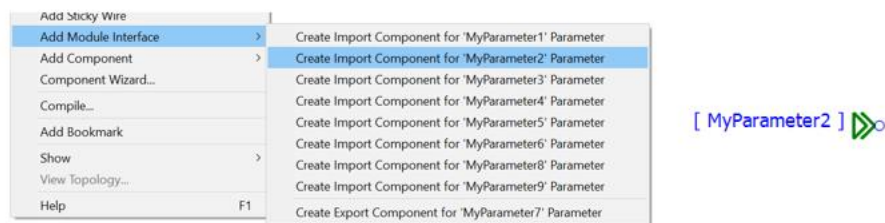
Added Niceties to Enhance the Circuit Building Experience (#1797, #8963, #8986, #9742, #9253, #10211, #10290)

We have added several new features that will help to make user's circuit-building experience more enjoyable:

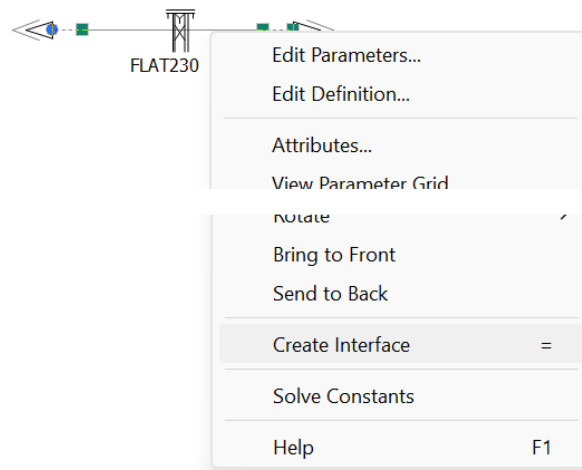
- Automatic Output Channel Creation:** A component context menu option has been added that will scan the component for output parameter signals, and the automatically construct an output channel with a data label attached, with names pre-set to the output parameter name. In addition, users may take this one step further by automatically creating and adding these output channels to a graph, or a meter in a control panel.



- Add Module Interface:** A new schematic canvas menu function allows you to quickly add XNodes or import/export tag components to a module canvas, based on any module port or parameter that has not already been paired with one of these components. The component is added pre-named to match the chosen parameter or port name.

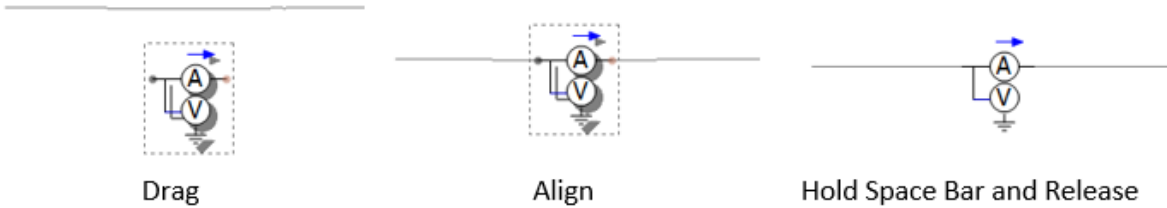


- Auto T-Line/Cable Interface:** A new transmission line/cable context menu function allows you to quickly generate a preconfigured, interface component to correspond with any transmission segment properties component. Simply right-click on the transmission segment properties component and select Create Interface.

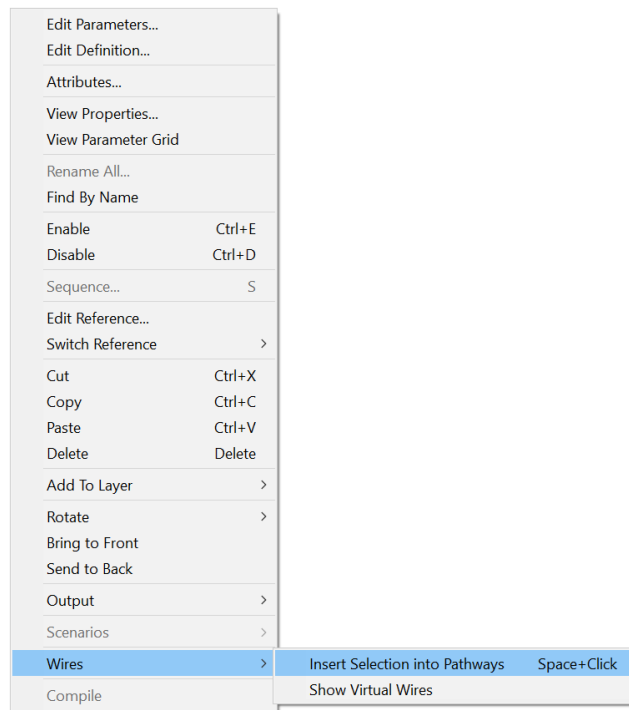




- Component Wire Insertion:** A new mechanism has been added to allow users to automatically insert a component inline within a wire, without going through the cumbersome steps of breaking the wire apart. Simply move the component atop a wire, lining up the connection ports, hold down the space bar and release the mouse button. Voilà!



Or simply align the component atop the wire, right-click and select *Wires | Insert Selection into Pathways*.



This functionality not only helps when adding series components like ammeters or multimeters but also helps simplify automated construction of circuits using Python script.

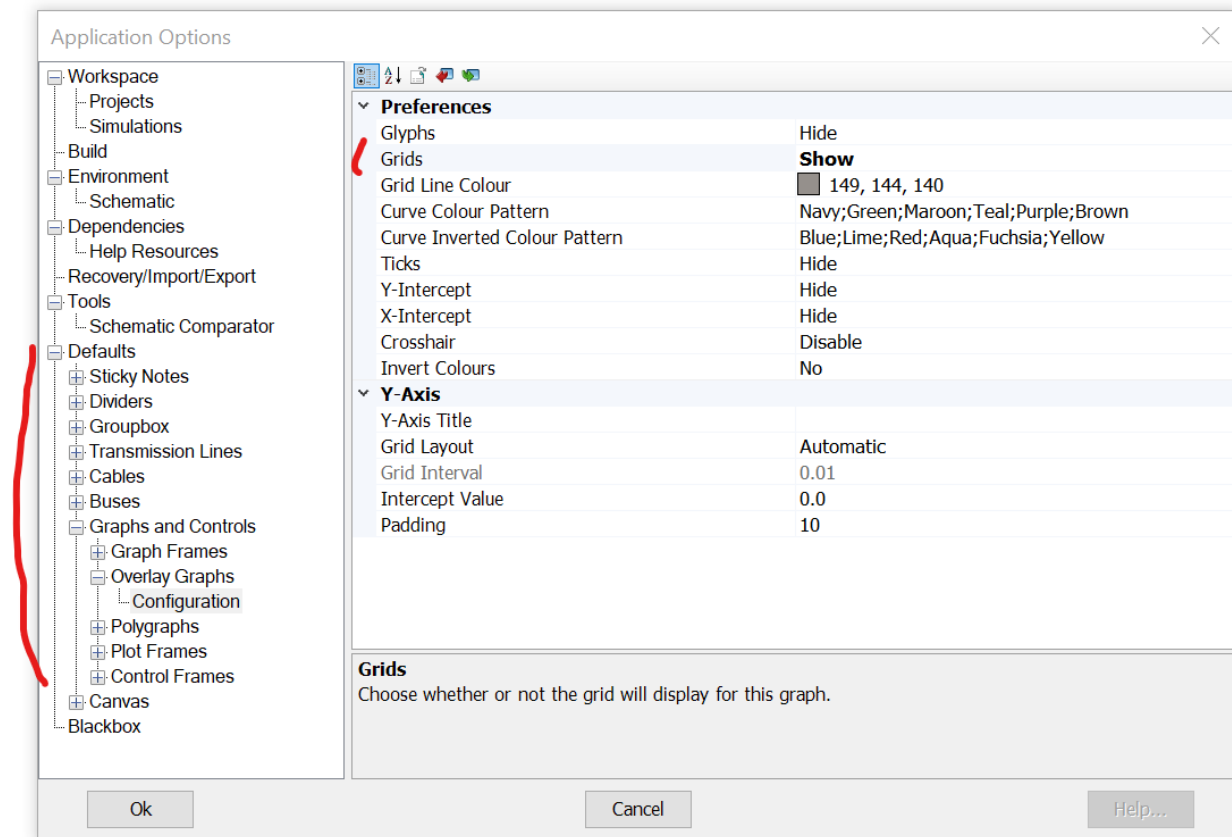
New Group Box Display Tool Added (#9374, #9378, #9421, #9429)

A new group box display object has been added. Like a sticky note, this object is optimized for use as a visual grouping tool. The group box can be adjusted for a variety of border styles and colours, as well as a coloured or transparent background.



Customizable Default Settings for Various Objects (#748, 4225, 4841, 5619)

Users can now adjust the default properties for most newly created schematic objects. For example, if you prefer the drawing grid to be enabled when adding a new graph, you can set the property default as enabled. Object default properties are part of the *Application Options* dialog, and so these settings are stored in your user profile and will persist across multiple workspaces.

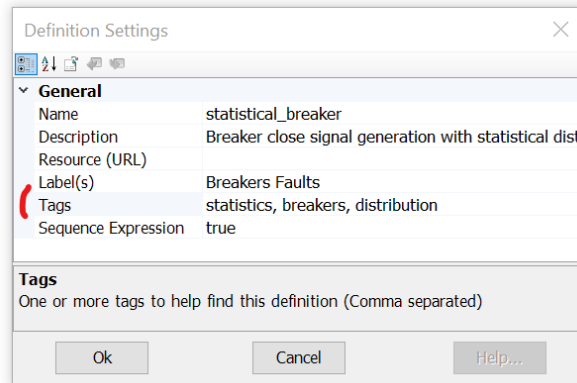


Component Definition Enhancements (#8594, #8721)

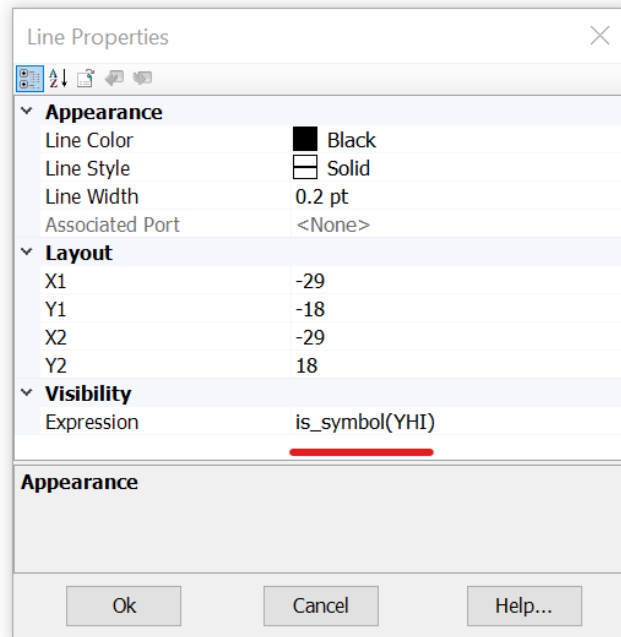
Some new enhancements have been made to component definitions:



- **Tag Setting:** Users can add tag names to their custom component definitions to help with searching; for both the main search tool, and when using the *Component Library Viewing Pane*.



- **New Evaluation Functions for Conditional Statements:** Three new evaluation functions have been added for use in component definition graphic and parameter conditional statements, which may be used to control the enable/disable/visible state of parameters or graphics.
 - `is_symbol(<symbol_name>)`: This function will return true if a valid signal name (such as ABC123) is entered as a value in the parameter corresponding to the entered symbol name. Otherwise, a false is returned.
 - `is_literal(<symbol_name>)`: This function will return true if a valid literal (such as 1.23) is entered as a value in the parameter corresponding to the entered symbol name. Otherwise, a false is returned.
 - `is_empty(<symbol_name>)`: This function will return true if nothing is entered as a value in the parameter corresponding to the entered symbol name. Otherwise, a false is returned.



New Resource Macros for Automating File Names and Paths (#2525, 3307, 9622)

A huge number of new macros have been added to help with the automation of resource file names and paths (this includes file and paths used in project settings and application options).

For example:

- **\$(FileVersion):** Substitutes the last version of PSCAD used to save the current state of the project file (ex. 4.6.3).
- **\$(Creator):** Substitutes the name of the project creator (the user who initially created the project).
- **\$(CreateDate):** Substitutes the date the project was created.
- **\$(CreateTime):** Substitutes the time the project was created.
- Plus, many more.

For a complete list of new macros, see the [Appendix](#) at the end of this document.

Miscellaneous Niceties (#9405, #9516)

In addition to the above, the following new features have been added:

- **Drag Panning:** Users may now drag objects, such as a graph panel or component, to the edges of the viewable canvas and PSCAD will pan (i.e., auto-scroll) in the direction the mouse is pulled.

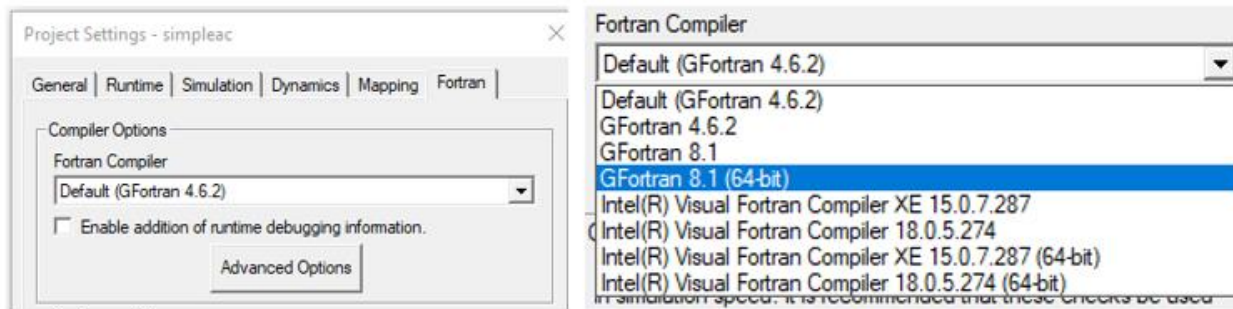


- **Selectable IP Address:** PSCAD now provides all IP addresses available on the host machine in the Application Options (under Workspace | Simulations), so user can manually select which one to use. This can help resolve any IP address contention issues that arise.

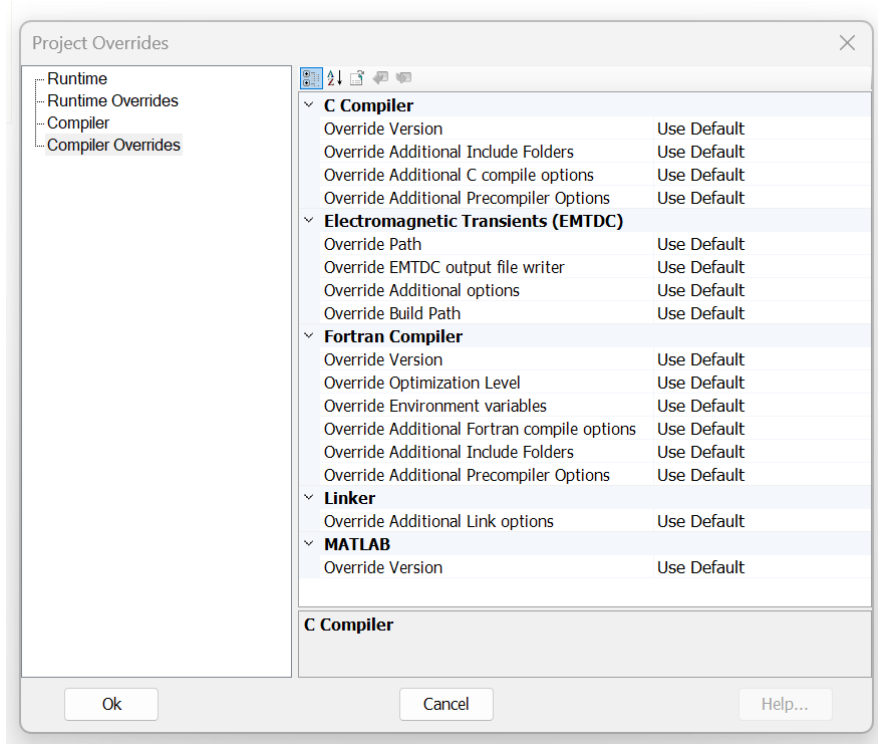
Functionality Deficiency Fixes

Project-Based Fortran Compiler Settings (#8632, #8813)

Fortran compiler settings may now be specified on a per-project basis as needed (as opposed to purely application-based in previous versions). By default, PSCAD will use the compiler settings specified in the *Application Options | Dependencies* category. However, if different compiler settings are necessary for different projects within the same workspace (for example, a PNI simulation launching multiple projects in parallel), users may now override the compiler settings set in the *Application Options*, on a per-project basis via a *Project Settings* selection.

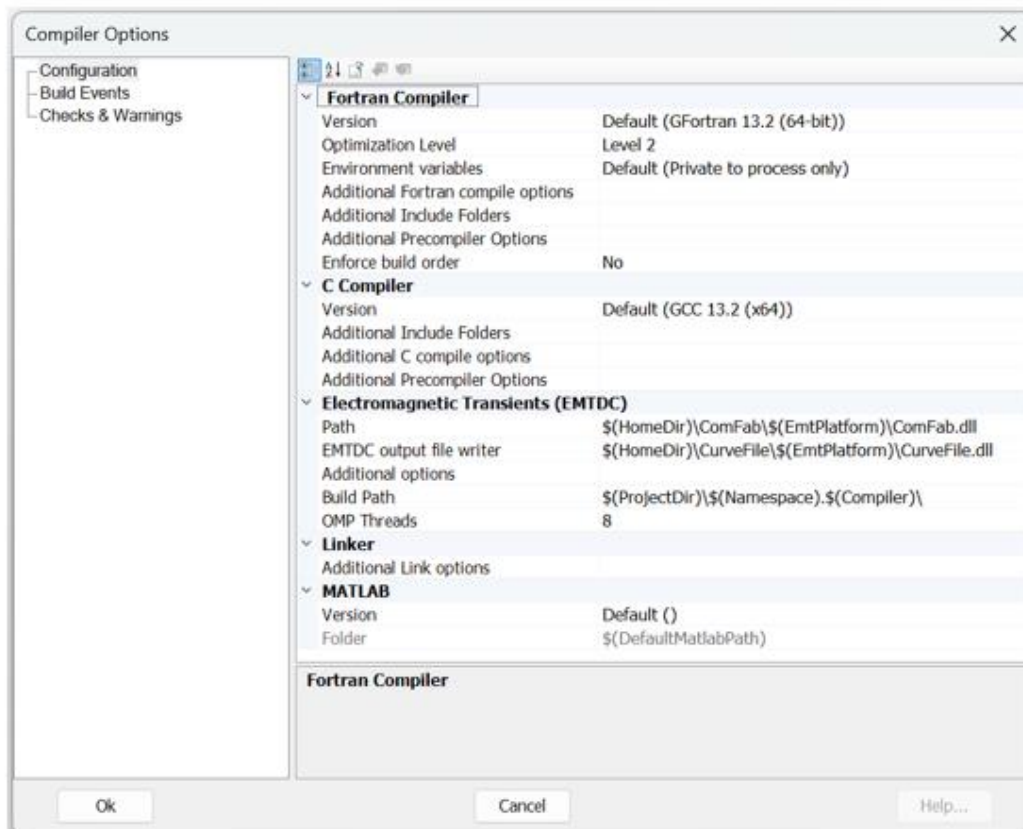
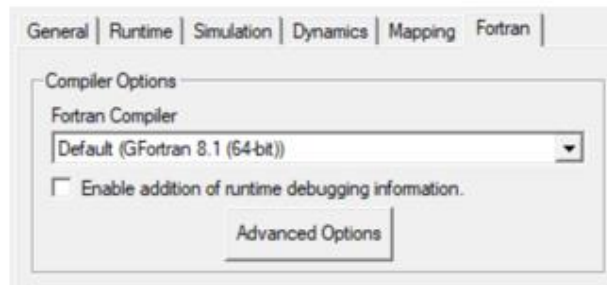


The ability to set Fortran compiler and Visual Studio configurations on a per-project basis is also now available to be overridden on a per simulation task basis. This enables you to run simulation sets in sequence that include simulation tasks using different compiler/Visual Studio combinations in different simulation sets.



Compiler Advanced Options (#4732, #8037, #9206)

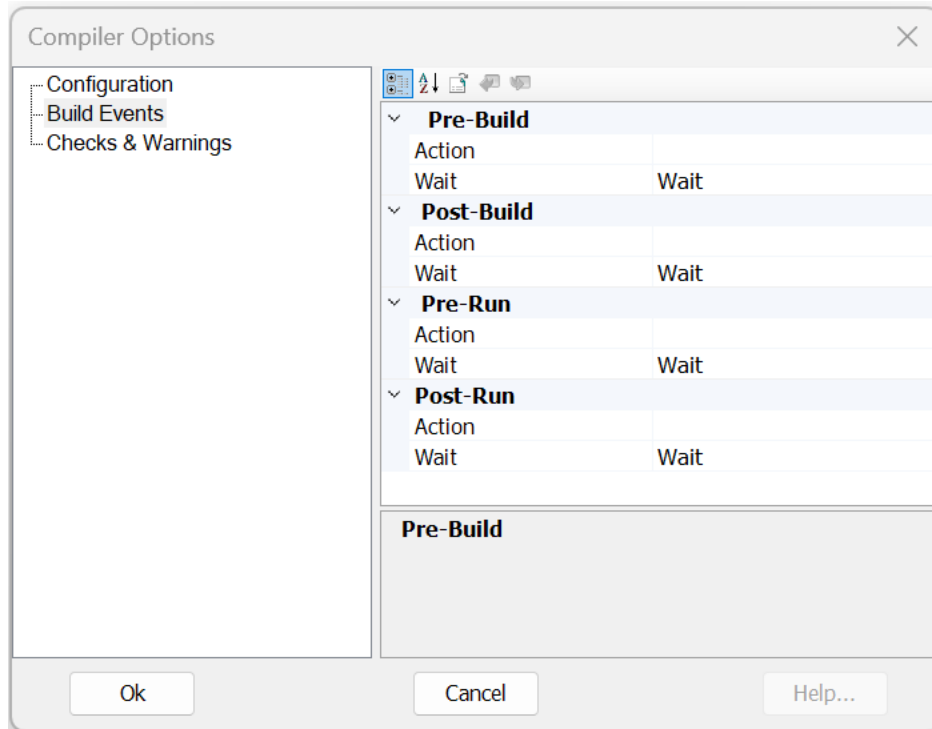
The project settings now include far greater access to internal compiler settings, via an *Advanced Options* button, within the Compiler Options area, under the Fortran tab.



Project-Based, Command Line Event Options (#9928)

Users may now launch command line-based programs or events at specific times between building and running their simulation. Launch events can be specified to occur during pre-build, post-build, pre-run, and post-run, via the *Build Events* category in the *Compiler Options* dialog (see *Compiler Advanced Options* above).

An example usage for this feature is say a user needs to run a certificate program after generating the EMTDC executable file, before it runs, to allow their local security program to run first. In this instance, the user would provide command line commands in the post-build or pre-run input fields.



Project-Based, Multiple MATLAB Version Support (#8915)

Users may now associate different versions of MATLAB with different case projects. This functionality is controlled from within the Compiler Advanced Options as shown above. By default, the MATLAB version specified in the Application Options is used, but this may now be overridden at the project level.

SCons Replaces Legacy Make/NMake Build System (#9859)

The legacy make/nmake command-line interface tool, which has been used by PSCAD since its inception to provide build instructions for the associated Fortran compiler (via the generation of a configuration file called a *makefile*), has been replaced with the more modern, python-based tool called *SCons* (<https://scons.org/>).

SCons is an open-source, software construction tool. Think of SCons as an improved, cross-platform substitute for the classic make utility with integrated functionality, like *autoconf/automake* and compiler caches such as *ccache*. In short, SCons is an easier, more reliable way to build software.

SCons makes the dependency management and building of large simulations much faster and more reliable.



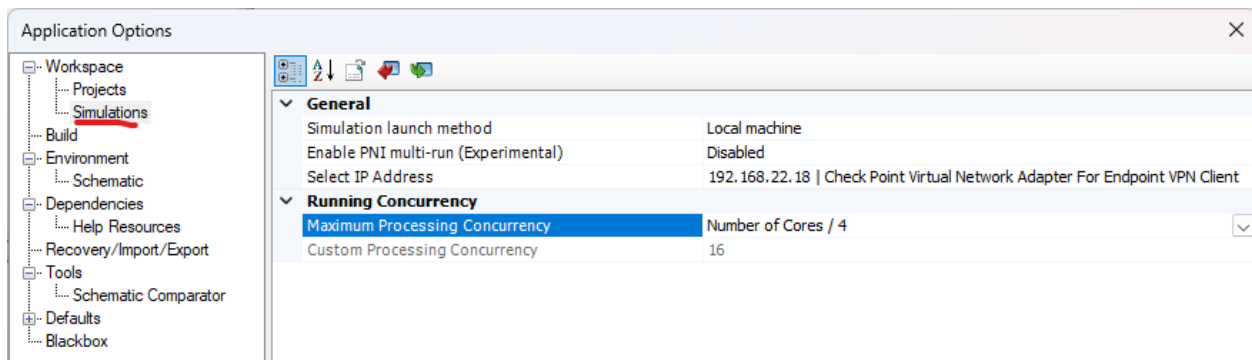


Simulation Speed Enhancements (PSCAD Side) (#10011)

As user’s projects become larger and require more computing resources, we have been working hard to identify choke points and other processing inefficiencies.

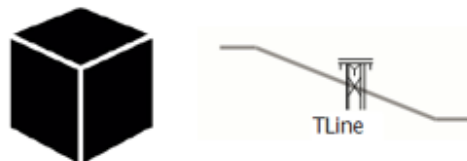
- PSCAD requires different resource levels allocated, depending on how it is being used. If PSCAD is running on the same computer, on which the EMTDC simulations are running, then it is a good idea to limit the number of threads that PSCAD is allowed to run. If the simulation (EMTDC) is being launched on a different computer, then PSCAD should be free to use all the resources on its computer to process the data coming back from the simulations.

A new application option is added to allow the user to limit (or not limit) the number of threads PSCAD can run on its computer. By default, this is set to ¼ of the machine’s cores, but it is adjustable by the user.



Transmission Lines in Blackbox (#9056)

Overhead transmission lines and underground cables are now supported when blackboxing modules. Utilizing a brand-new definition script segment called *Subsystems*, users may include transmission systems within a module to be blackboxed, provided both ends of each line exist within the same module. The presence of transmission lines also means that multiple, electrical subsystems are now supported for the first time, within non-module components.



Other Blackbox Enhancements (#9065, #9533, #9615, #8598)

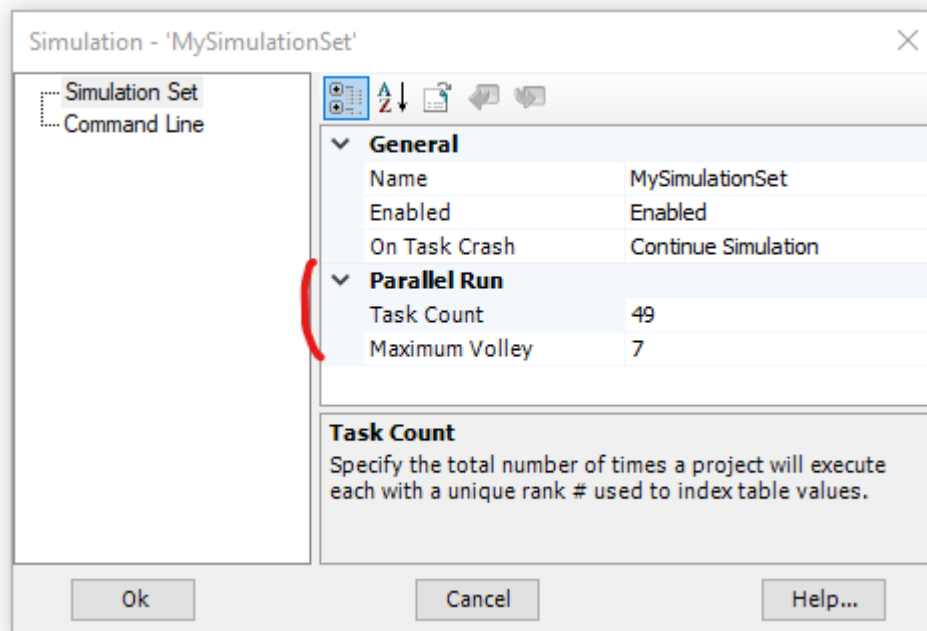
The following describes various other enhancements that have been added to Blackbox:



- **Definition Description:** Blackboxed component definition descriptions are now copied from their respective source module. A new blackbox application option is added to accommodate user entry of definition labels. These labels are appended to any existing labels.
- **Options:** Blackbox-related application options are now presented as a pop-up dialog at each blackbox operation. The options presented each time are read from those set within *Application Options | Blackbox* category.
- **Co-Simulation Component:** Any Co-Simulation component that may exist in the module being blackboxed, is now fully supported.
- **Subroutine Naming Convention:** To avoid possible conflicting subroutine names, Blackbox now ensures that the subroutines it creates are unique from those created by PSCAD, for EMTDC. Blackbox now adds 'BB' to the subroutine names: *BBDyn*, *BBDyn_Begin*, *BBOut* and *BBOut_Begin*.

Combined PMR/PNI Simulations Now Possible (#9055)

Those who utilize the High-Performance Computing (HPC) functionalities in PSCAD, rejoice! It is now possible to simultaneously run a *Parallel Multiple Run (PMR)* of a *Parallel Network Interface (PNI)* simulation. Setup of such a simulation is straightforward: Simply provide *Task Count* and *Maximum Volley* parameters to the Simulation Set encapsulating the PNI simulation:

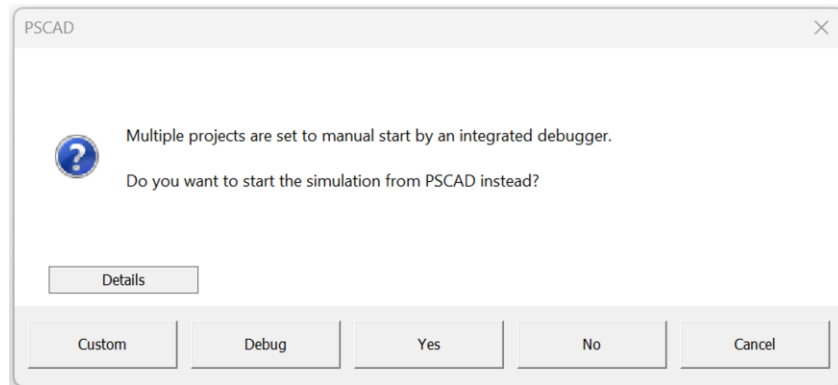


Note that usage of the *Rank Number* tag in your PNI simulation projects is still necessary as per a standard PMR simulation.



Single Dialog when Launching a Simulation Set Containing Manually Started Projects (#9775)

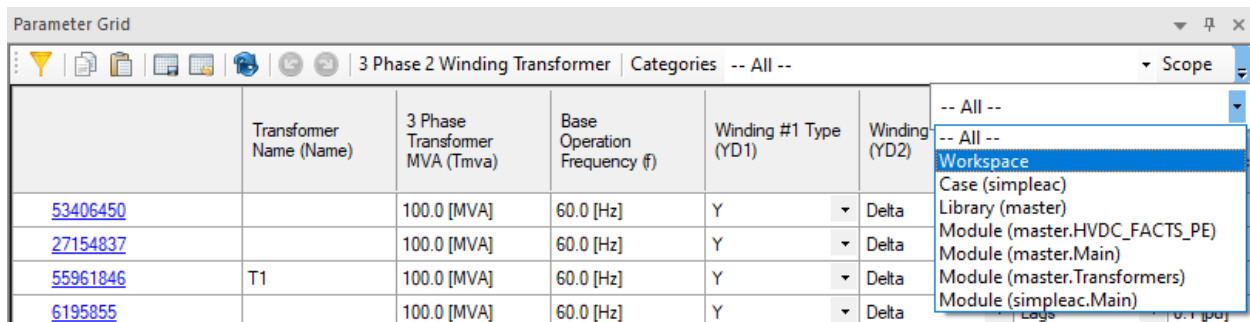
When launching a simulation set containing two or more projects that are set to manually start with an integrated debugger, PSCAD will now issue a single dialog to allow selection on whether to manually start or not, for all projects in the simulation set.



In previous versions, individual dialogs would pop-up for every project set to manually start. If users so desire, there is a custom button on the above dialog that provides the ability to set project launch configurations on an individual basis.

The Parameter Grid Goes Workspace Wide (#8952)

The parameter grid feature is now even more powerful. In addition to the ability to filter results by component parameter category page, it now boasts a selectable *Scope* drop list, which can limit results by project type (i.e., case or library), or by any module that exists in the workspace. Users may even set the scope of results across the entire workspace, to include results from all modules in all loaded projects.

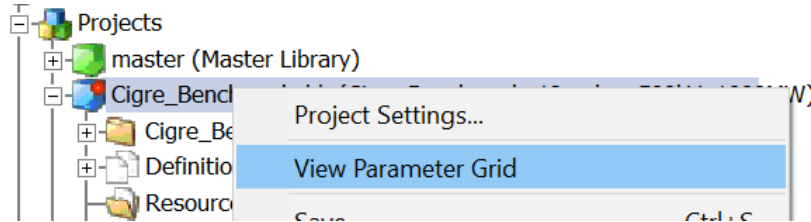


Other Parameter Grid Enhancements (#9114, #9235, #9199)

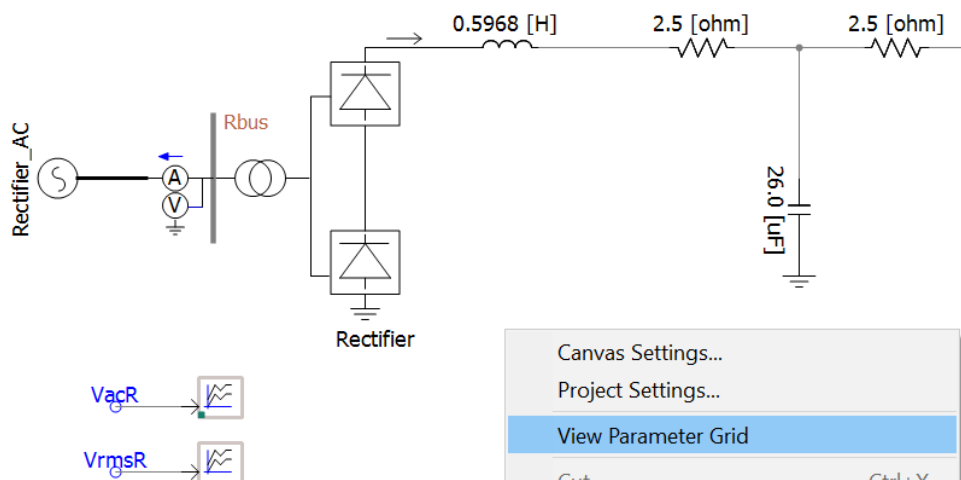
The following describes various other enhancements that have been added to the Parameter Grid:



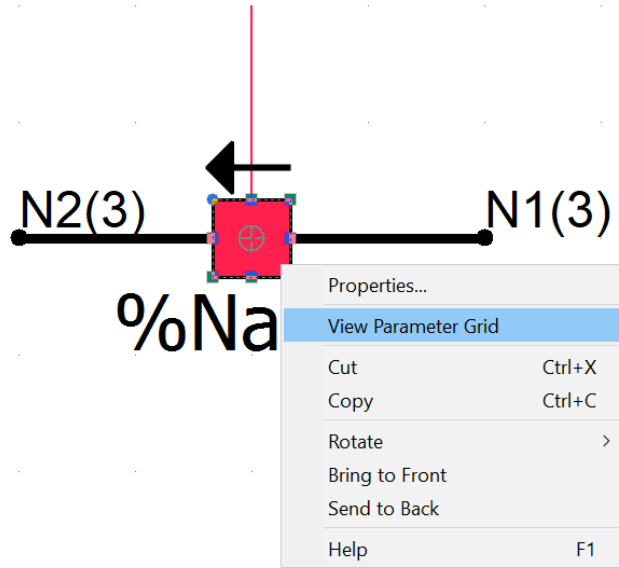
- **Project Settings:** Project settings dialog contents are now supported in the parameter grid.



- **Canvas Settings:** Canvas settings dialog contents are now supported in the parameter grid (right-click on any schematic canvas).



- **Component Graphics Objects:** Component definition graphic object properties are now supported in the parameter grid.



- Viewing Only Selected Components:** A new scope has been added so when viewing parameter grid results, you can focus only on **selected** components. View the parameter grid as normal, select which components you want to focus on and select the scope as *Current Selection*.

Parameter Grid

Summing/Differencing Junctions Scope Current Selection

Name (Name)	Data Path Type: (DPath)	Signal Dimension (Dim)	Input A (A)	Input	Input E (E)	Input F (F)	Input G (G)
1855092531	Integer	1	disable	subtra	able	subtract	disable
368323577	Integer	1	disable	disable	able	add	disable
892274134	Integer	1	disable	subtra	able	disable	disable

Context Menu:

- All --
- Workspace
- Current Selection**
- Case (ParallelMultirun1)
- Case (ParallelMultirun2)
- Case (SerialMultirun)
- Library (master)
- Module (master.CSMF)
- Module (master.Main)
- Module (master.Meters)
- Module (ParallelMultirun1.Main)
- Module (ParallelMultirun2.Main)
- Module (ParallelMultirun2.PMRCon
- Module (SerialMultirun.Main)

Wire Mode Refined (#8740)

Wire mode has been refactored under the hood for better performance, stability, and ease of use.





Python Interpreter Upgrade (#9779, #10119)

The embedded Python interpreter has been upgraded from Python 3.7 to Python 3.12, allowing embedded scripts access to new features of the language. These include a simpler syntax for dictionary union operations, the walrus operator (`:=`), debug-style f-strings (`f"{expr=}"`), and improved type-hint syntax to name a few.



Additional Python-Related Updates (#9579, #10031, #10042, #10050, #10139)

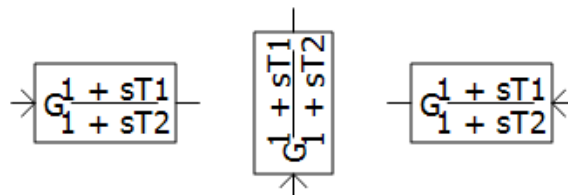
The following functionalities have been added existing Python scripting capabilities.

- **Paste Definitions:** *Paste Definition* & *Paste Definitions w/ Dependents* is now fully implemented. Previously the record function added these commands to the script, but they were added as comments (preceded by #), as they were not fully implemented.
- **Node and Branch Search:** Users may now search for a specific node or a branch, given the node or branch number and the subsystem.
- **Update Python Library Button:** A new button has been added to the Tools tab in the ribbon control bar, allowing users to update the mhi.pscad Python Automation Library as a per-user package.
- **Port Specification:** Users can now create a configuration file, `~/.mhi.pscad.py`, which can be used to specify a port or port range for Python automation to use.

Mirror/Rotate Restriction Control for Component Definition Graphic Objects (#83, #9880, #10034)

Component graphic objects may now have of their graphics orientation locked to prevent mirroring and to restrict rotating of those objects.

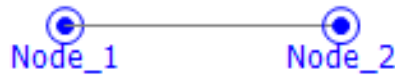
A new option is added to the component definition graphic object properties called *Orientation Lock*, which can be used to lock enable the locking.



Short-able XNodes (#8499)

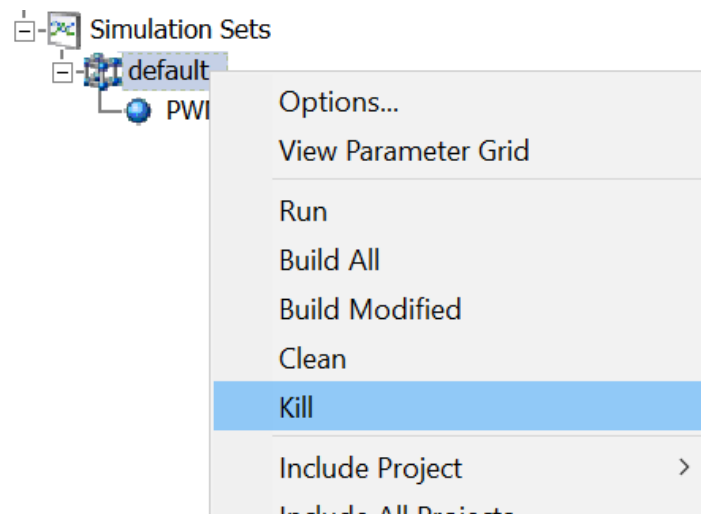


XNodes may now be shorted together (i.e., without a defined branch between them). This was an issue for users with module schematics resulting in shorted XNodes, based on compile-time scenarios. A preprocessing step is now added to identify shorted XNodes, which are then mapped to the same node following global node mapping.



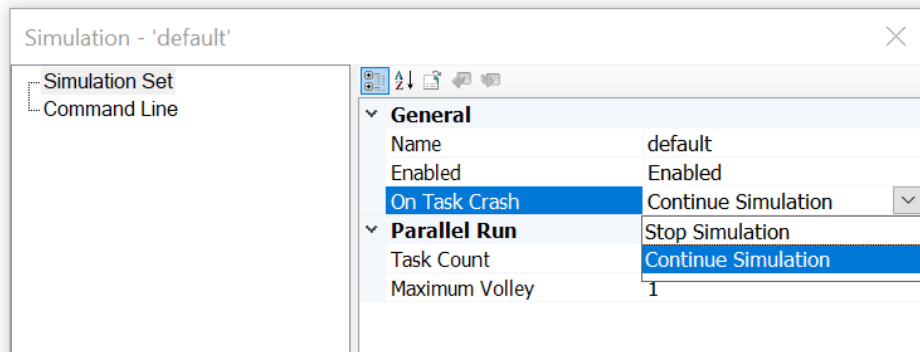
Easy Kill of Launched Simulation Processes (#9103)

If PSCAD crashes during a large simulation, or is unable to stop all running simulation processes, it can be difficult to track these down and kill them manually. A kill option has been added to the simulation set, which will find all simulation processes pertaining to that set by name and kill them. This will be done regardless of whether they were started by the current instance of PSCAD or not.



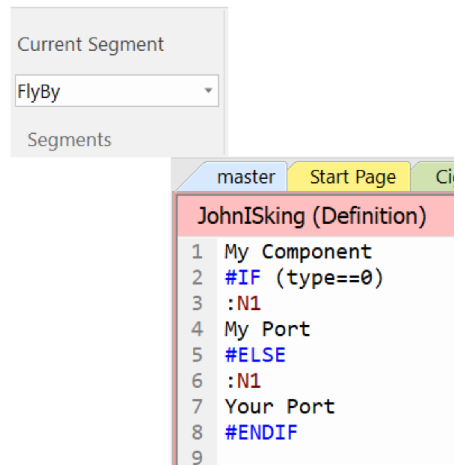
New Option to Automatically End Simulation Set on Task Crash (#9137)

If a single EMTDC instance crashes or dies during a simulation set run, that simulation set will become stuck, continuously waiting for the dead task. This could put an exceedingly large amount of stress on a machine until the task is manually killed. No longer. We have added a simulation set option that will automatically shut down the entire set, if one or more tasks crash or die.



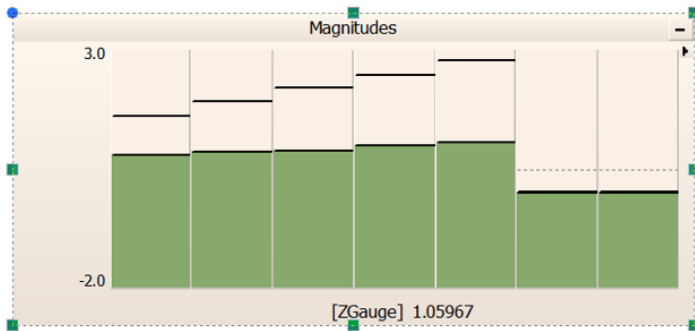
Conditional Statements in Flyby Script Segments (#1546)

Conditional statements (#IF, #ELSE, #ENDIF, #CASE) are now allowed in the FlyBy component definition script segment. This allows users to change the flyby display over custom components, depending on user input. This is especially important when say connection ports vary in dimension and/or type.



Instrument Properties (#9175)

All display instruments (i.e., *PolyMeter*, *PhasorMeter* and *Oscilloscope*) now possess their own unique properties that may be adjusted independently of their associated *Output Channel*. In previous versions, display limits were based on the minimum and maximum display limit parameters of the output channel linked to the device. Note that when an instrument is first created, their default property values are still derived by default from the associated output channel.



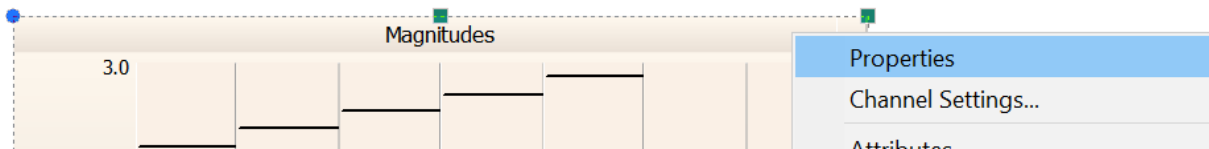
Polymer ✕

General

Name	
Title	Magnitudes
Scrollable	Non-Scrollable
Processor Core Activity	No Display
Color	■ 135, 169, 107
Default Maximum Display Limit	3.000000
Default Minimum Display Limit	-2.000000

Default Maximum Display Limit
Enter a value to control the maximum display limit on devices.

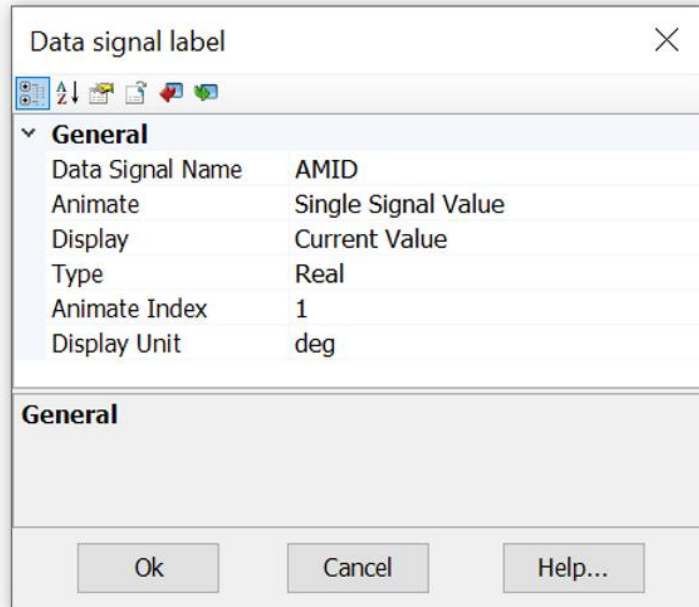
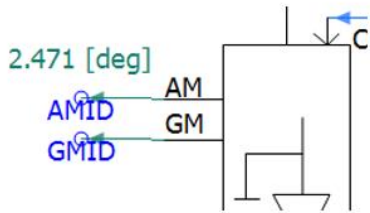
Instrument properties may be adjusted at any time: Prior, during, or post simulation.



Animated Data Labels (#9124)

Data labels may now be set to display the value of their associated signal. Utilizing the *Animated Graphics* functionality in PSCAD, data label parameters may be adjusted to display the dynamic signal value during the run, or the maximum or minimum signal value that occurred over the entire run.

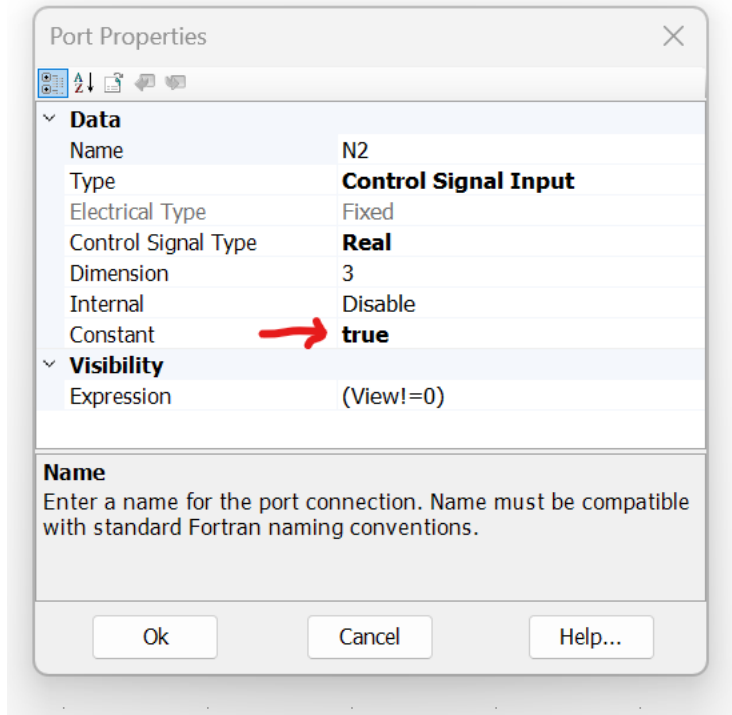
Simply ensure that you have enabled *Animated Graphics* in the project settings. Then, adjust the data label parameters to display what is needed.



Component Ports May Now Be Defined as Constant (#10054)

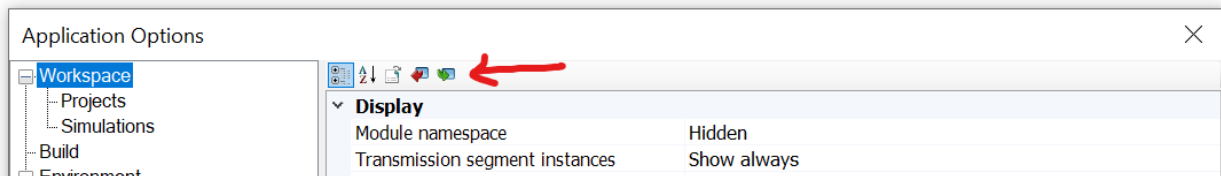
Ports within component definitions may now be set as constant when the port is defined as non-electrical. If the port is operating as an input, it will only accept a constant signal. If functioning as an output, it will create a constant signal.

Master library components that generate constant signals, such as the constant tags, rank, Delta-T, etc., have already been modified in v5.1.



Application Options Import/Export to *.csv File (#9345)

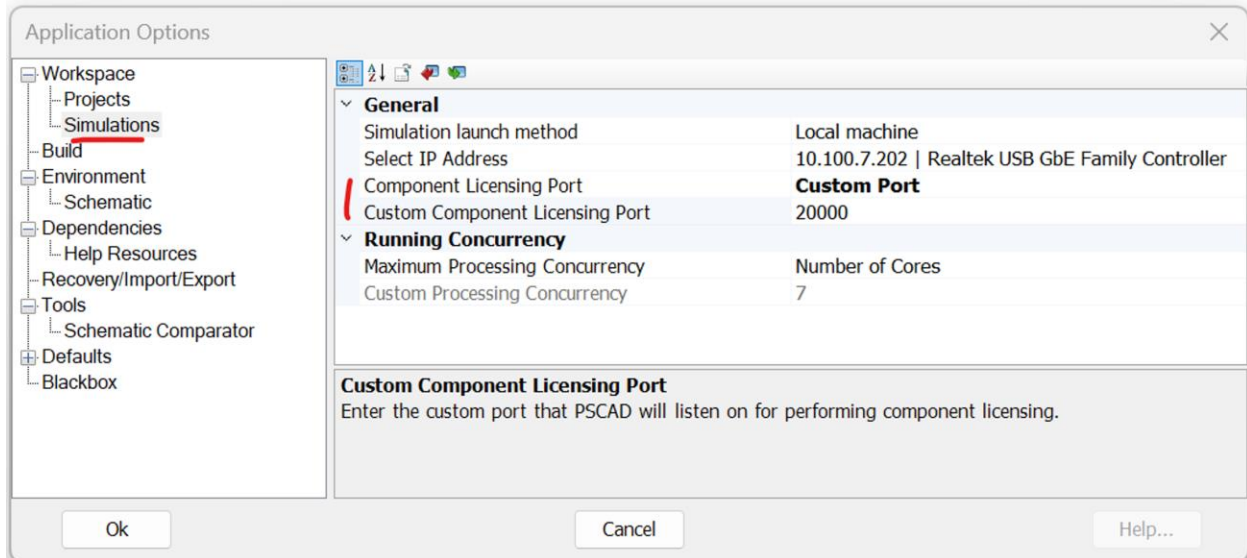
Users may now export or import their application options, to or from a *Comma-Separated Values (*.csv)* file. This comes in handy when the exact same application option settings need to be setup over multiple machines on the network. Simply export the options to a *.csv file on the first machine, and then import to all others.



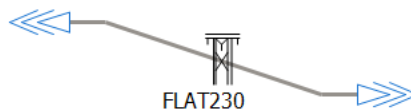
Miscellaneous Addressed Deficiencies (#5858, #8223, #8694, #8744, #8946, #8971, #9095, #9109, #9200, #9391, #9431, #9452, #9489, #9510, #9620, #9734, #9738, #9741, #9904, #9905, #10012, #10139, #10218)

In addition to the above, the following deficiencies have also been addressed:

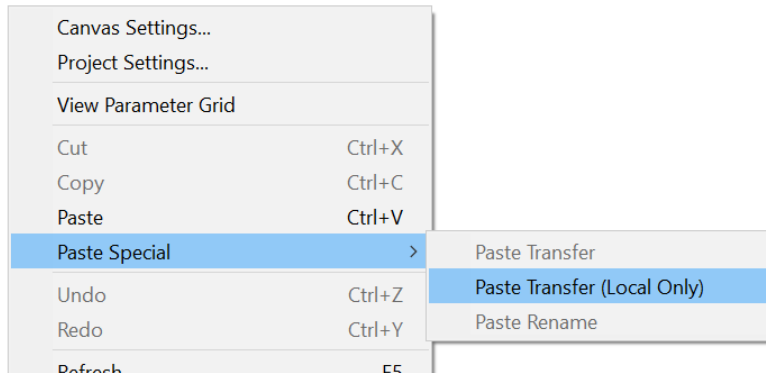
- **Loaded Project Limit:** The total number of projects per workspace has been increased to 1024.
- **Component Licensing Port Specification:** A port number to be used by Component Licensing may now be directly specified in the Application Options. The default is set to any port.



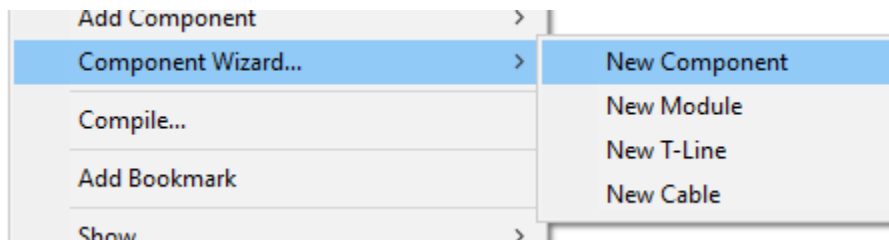
- Snapshot and Map File Names/Paths:** PSCAD and EMTDC now support full, Windows 260-character file paths and names for snapshot (*.snp) and map (*.map) files. Previously these file names were limited to 34 characters maximum, and therefore could only be assumed to reside in the temporary folder (thereby avoiding long absolute paths). Also, PSCAD can now provide (and EMTDC accept) full snapshot and map file paths via the command prompt.
- PNI and Foreign Transmission Segments:** When compiling a PNI simulation that includes foreign transmission segments (i.e., the segment properties are located on another computer), the resulting LCP output files (*.tlo/*.clo) are now compiled into the local folder on the computer being used to launch the simulation.



- Paste Transfer (Local Only):** A new paste transfer option has been added which, when selected, ensures that only definitions local to the project where the copy action occurred, are transferred to the destination project. That is, all definitions that are part of referenced library projects are not included in the paste transfer.



- Component Wizard Quick Options:** When accessing the component wizard from the schematic canvas, new menu items have been added for quick configuration of the wizard to specific component types.

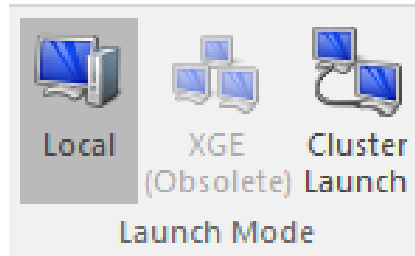


- Jumbo-Sized Canvas and PSCAD v4.6:** PSCAD v5.1 now supports jumbo-sized canvases on *Save As v4.6* format. Using a tile-sweep algorithm, PSCAD will split any Jumbo canvas it encounters into smaller canvases, while producing minimal changes to the structure of the schematic circuit. These smaller canvases are stored as module definitions in the project definition list and must be added connected within the project manually in PSCAD v4.6. For example, if a jumbo module canvas is named *MyCanvas*, the supplementary modules created will be called *MyCanvas_1*, *MyCanvas_2*, etc.
- Workspace Save and Open Projects:** When saving a workspace, PSCAD now additionally tracks and saves which projects were being viewed at the time of save. So, when the workspace is reloaded, only these projects (viewed during the last session) will be re-opened in the schematic canvas view. This is helpful in instances when a workspace contains many projects, but only a few are being focused upon.
- *.inf/*.map File Updates:** The *.inf file (and the output channel part of the *.map file) have been updated to include the full module hierarchy path to where the output channel is in the project. This extra information was not included at all in versions before v5.1.0.

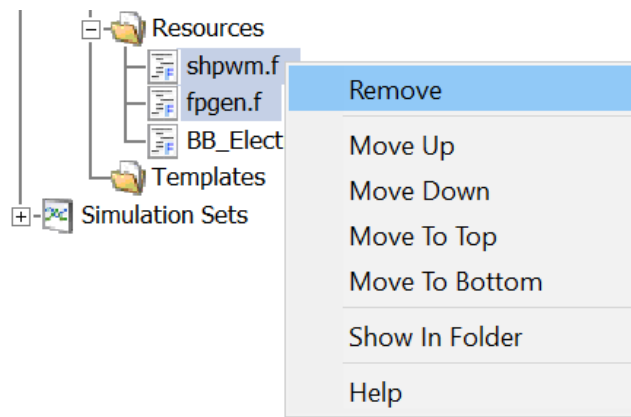
PGB(1)	Output	Desc="3 Phase Source Current:1"	Group="Source Current"	Max=1.50	Min=-1.50	Units="kA"	Path="Main(0):3 Phase Source Current"
PGB(2)	Output	Desc="3 Phase Source Current:2"	Group="Source Current"	Max=1.50	Min=-1.50	Units="kA"	Path="Main(0):3 Phase Source Current"
PGB(3)	Output	Desc="3 Phase Source Current:3"	Group="Source Current"	Max=1.50	Min=-1.50	Units="kA"	Path="Main(0):3 Phase Source Current"



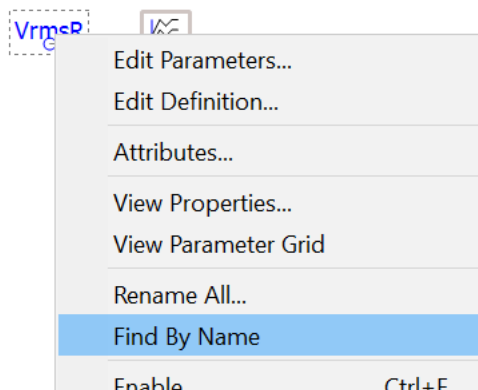
- Launch Mode:** New, more sensible buttons have been added to the ribbon Tools tab, for the easy facilitation of Local (local machine) or Cluster Launch (multiple machines using the Cluster Launch System (CLS)) types of simulations.



- Resources Branch Multi-Select:** Users may now per perform actions (such as Remove) on multiple resources simultaneously.

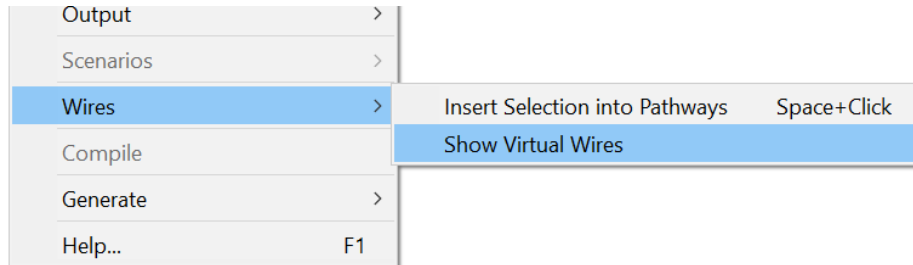


- Quick Search by Name:** PSCAD now allows users to quickly search for other components that possess the same name (from the component *Name* parameter). This may be used to quickly search for t-line or cable ends, data labels, node labels, etc. The results of the search are provided in the *Search Results* pane, complete with navigable links.

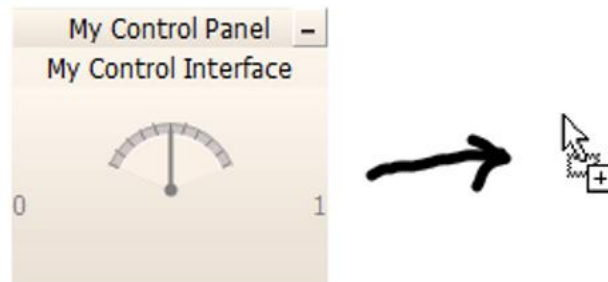




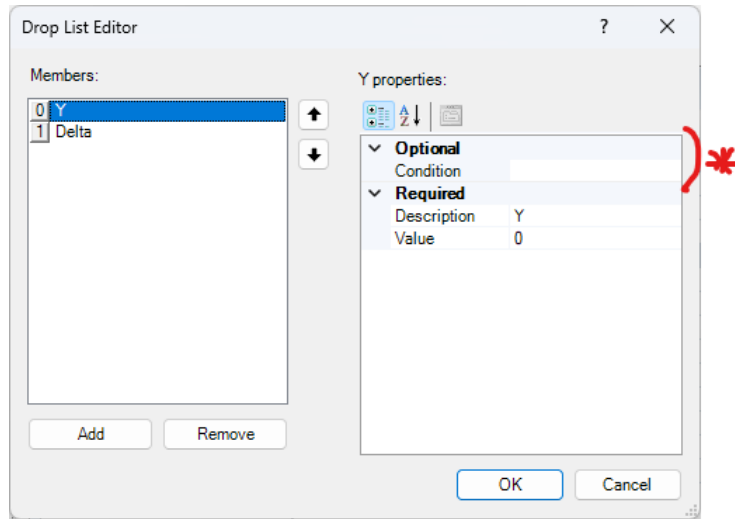
- **Component-Based Virtual Wires:** Regardless of whether or not Virtual Wires are enabled on a schematic canvas, users can now quickly display virtual wires specific to a single component. Simply right-click on a component and select Wires | Show Virtual Wires.



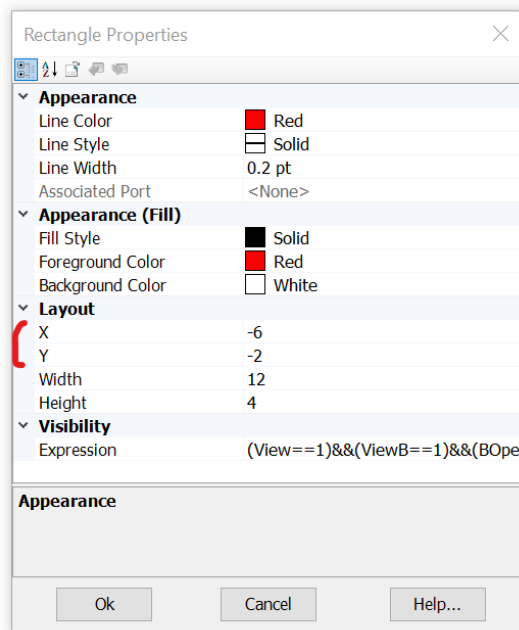
- **Undo/Redo in Component Definition Script Section:** Full undo/redo support has been added to the component definition editor script section. Both *Ctrl + z* and *Ctrl + y* hotkeys are functional, plus new buttons were added to the *Scripts* ribbon tab.
- **Control Panel Creation via Drag and Drop:** Control interfaces in control panels may now be dragged onto a blank part of the schematic canvas to create a new control panel (*Ctrl + left-mouse drag*).



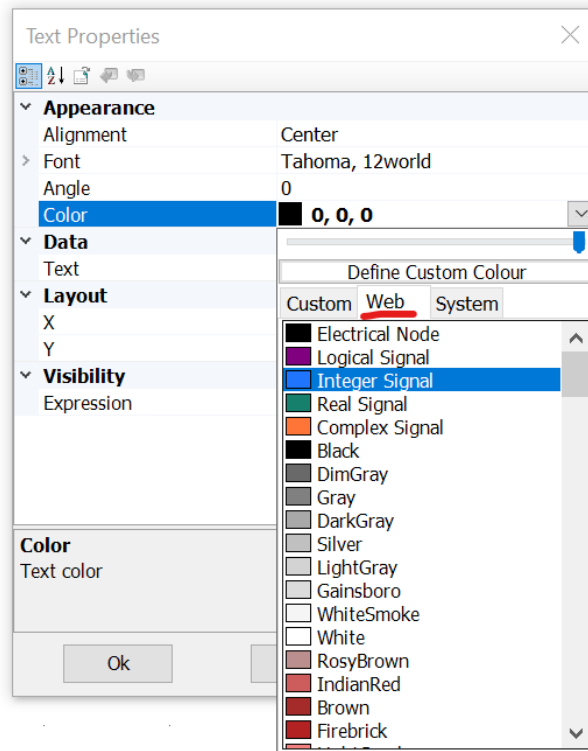
- **Drag and Drop Control Interfaces Between Control Panels:** Control interfaces may now be copied and pasted between control panels via drag and drop (*Ctrl + left-mouse drag*).
- **Individual Parameter Choice Field Item Conditional Statements:** Conditional statements may now be associated with individual items (collection) in a parameter choice field. This is beneficial when it is necessary to filter the collection list based on the values of other parameters.



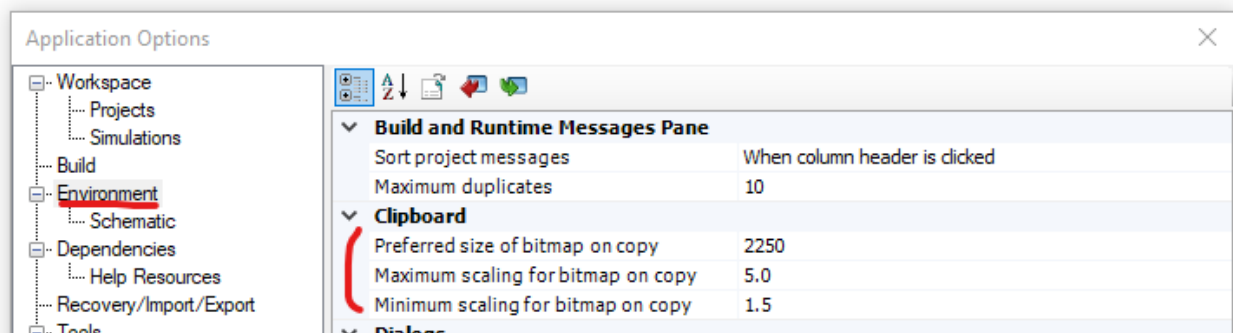
- **Double and Single Quotes in Conditional Statements:** Both double and single quotes are now processed properly as part of conditional statements. For example, `View == "0"` or `View == '0'`.
- **Component Definition Graphics Position:** Users may now adjust the position of component graphics directly via their individual properties.



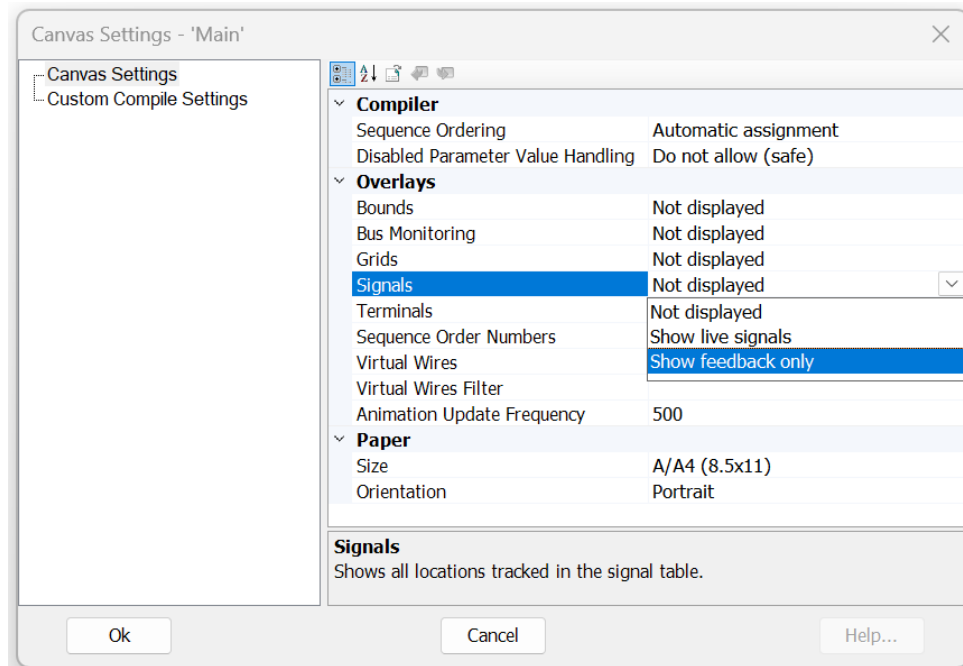
- **PSCAD System Colours Added:** Users may now select PSCAD system colours directly (i.e. integer blue, logical magenta, etc.).



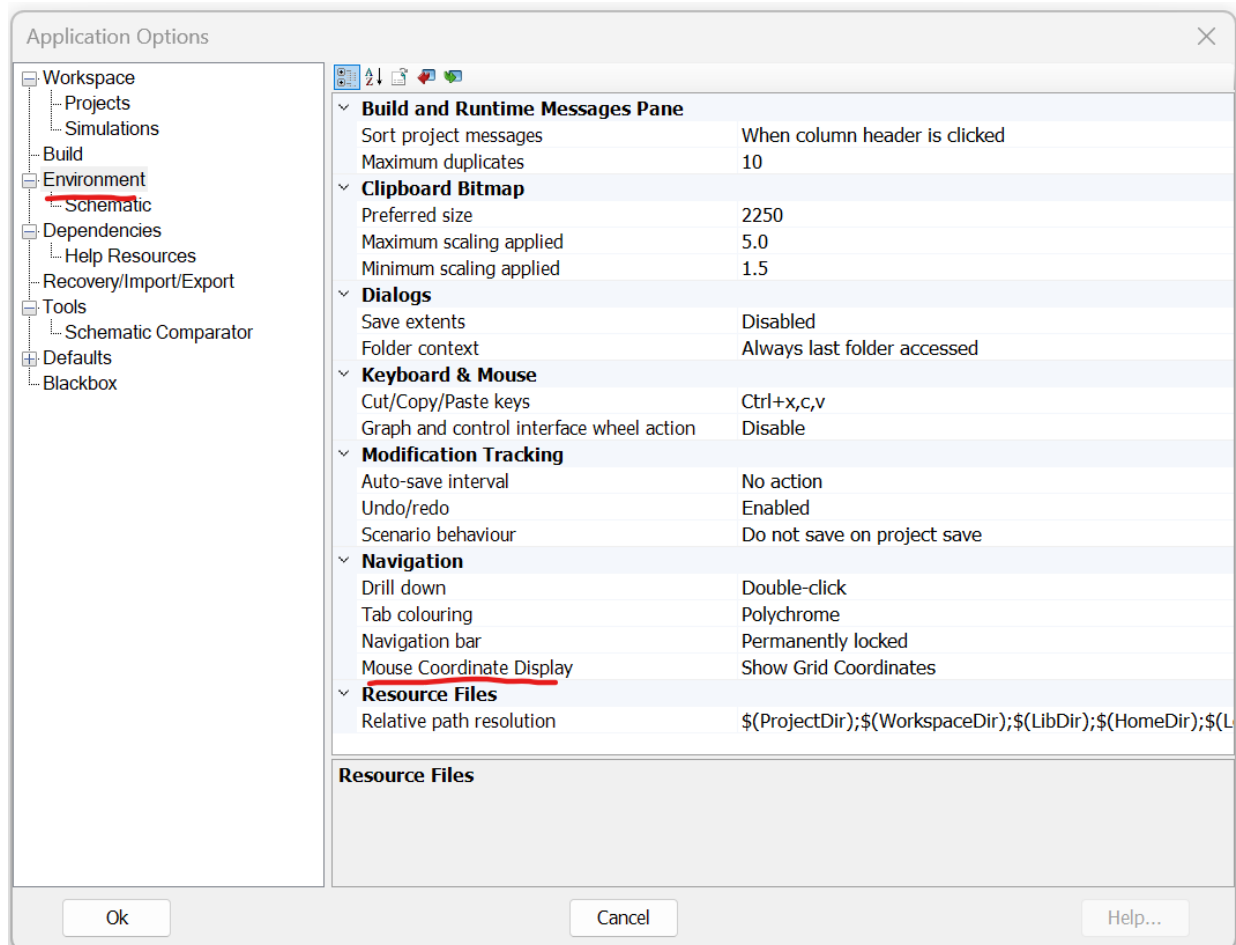
- **Bitmap Scaling Added to Application Options:** New application options have been added to the dialog *Environment* category, which provide some adjustability to how bitmaps are handled by the clipboard when copied from PSCAD.
 - Preferred Size in Pixels
 - Maximum and Minimum Scaling



- **Displaying Feedback Signals:** The *Signals* option in the module Canvas Settings has been updated to include a 'Show feedback only' option. This can help provide visual indicators where feedback loops exist on the Schematic.



- **Grid Coordinate View:** The coordinate view in the status bar may now toggled between mouse coordinate and grid coordinate view. Grid coordinate view helps when setting up Python scripts that involve direct canvas grid coordinates.



Bug Fixes

The following is a list of bugs that were discovered and fixed since the latest, previously released product.

1. When in wire mode, points can no longer be added to the wire outside the canvas bounds (#8347).
2. Blackbox-generated object files are now properly copied to the Blackbox folder.
3. Variables defined in the Computations script segment have been increased from a maximum limit of 8, to 32-characters in length (#8855).
4. Resource files in the project Resources branch may now be ordered via the resource context menu (move up, down, top or bottom) (#9044).
5. Print preview mode now allows you to scroll through all schematic canvases available to print (#8793).



6. The #VERBATIM script directive now works as intended (#9338).
7. PSCAD now detects Line Constants Program errors, even if the *.tlo/*.clo file was successfully generated (#9089).
8. Sticky wires now properly undo/redo when moved (#9453).
9. Cleaned up validation techniques for all parameter types. All parameters now accept literal values, empty values, variables, global substitutions (#9211).
10. **PSCAD no longer crashes when unloading a workspace if one library is dependent on another, and the dependant library is unloaded in such a way as a navigation event happens while the workspace is unloading (#9597).**
11. PSCAD will no longer create additional, empty subsystems in instances where two different buses of are touching (#9609).
12. In the component definition parameter editor, the move to bottom function now works properly (#9503).
13. A copied selection is now pasted wherever the mouse is currently pointing (when using ctrl+c/ctrl+v) (#9559).
14. Sub-sections of the xy plot frame (such as marker sliders) can no longer be cut from the greater pane (#9395).
15. Separate, redundant menu resources in the xy plot and graph panel frames have been removed (#9394).
16. Cut/Copy from the context menu of an xy plot now functions properly when right-clicking in the center of the frame (#9397).
17. 'Null Object' error notification window no longer appears after the deletion of all parameters in a component definition category (#9460).
18. Deleting a parameter within the parameter editor, via the ribbon Delete button, now functions properly (#9464).
19. Jumbo canvas size now an available within the transmission line/cable cross-section editor (#9465).
20. Run button in the ribbon no longer shows empty dropdown when no simulation sets exist in the workspace (#9490).
21. It is no longer possible (via a convoluted series of steps) for a transmission line or cable to become referenced to an invalid definition (#9462).
22. Invalid ribbon bar functions are now disabled while within the transmission line/cable editor (#9467).



23. When a selection containing two or more modules, which are dependent on each other via radio links, the radio link receiver, transmitter definition name is now updated properly (if the transmitter definition name indeed changes) (#9283).
24. When cutting a curve from an xy plot panel, the actual curve is not removed, instead of the entire xy plot panel (#9577).
25. Project files that possess an invalid namespace name will now be renamed to a valid name and loaded without interruption (#9647).
26. **Running a Python script containing a `pscad.logout()` command, no longer crashes PSCAD (#8949).**
27. Component definition script window now properly displays a blank page if no script segments exist in the definition (as opposed to the last viewed script segment from another component) (#9528).
28. Empty unit parameter values are now allowed in output channel components (#9326).
29. **Printing from the script, map or make file pane no longer crashes PSCAD (#8937).**
30. Application options are now saved to the user profile file when the dialog is closed (Ok button pressed). Previously, application options were only saved on application exit (#9272).
31. Arrow keys now moves the selection, rather than panning the canvas, when single or multiple components are currently box selected (#8459).
32. Copying and pasting a port in the component wizard no longer creates a new port without a name (#9712).
33. Blackbox now properly handles electric networks where no branches exists (but nodes are present) (#9702).
34. If the signal of a module output parameter is used as an output parameter of a different type in a script function call, the signal is now properly aliased before being passed into the function call (#9515).
35. **PSCAD will no longer crash if the *Enable PMR on PNI* option is enabled, and the user combines PMR-I using master/slave configurations and radio links, while using PNI transmission line sections (#9564).**
36. Curve menu placement is no longer based on the original view when opened, but the current view (this leads to the menu appearing in the wrong place) (#9590).
37. An error message is no issued if the user attempts to delete the default global substitution set (which is not allowed) (#9314).
38. Graph pane speed bar menu, plus some other smaller areas around the PSCAD application have been fixed to display properly on high DPI monitors (#7024).



39. License workgroup number has been added to the About Box display, for ease in determining license particulars (#8738).
40. The order the mutual lines are processed in can be arbitrary and may not be what the user expects. For example, if a π section or other component is created from the mutually coupled line, where the order of the transmission segment becomes important, the connection order is not obvious. Connection order information is now displayed on mutually coupled lines (#6738).
41. Animated substitutions, as well as mixing animated substitutions and static substitutions in conditional statements in a single text graphic, will now draw correctly (#9135).
42. When recording a script, the automation library no longer rejects the `project.save_as(...)` argument created as invalid (#9107).
43. It is no longer possible to enter an invalid equation into a numerical, component parameter type (#9208).
44. Dragging of multiple selected port in the component wizard is now possible (#9715).
45. Removed redundant, isolated node build warnings source from the cable interface components (and any others that have nodes only connected via matrix-fill scripts) (#9428).
46. A 'Do not show this again' checkbox was added to the 'Existing project namespace has been modified' dialog. This will help when loading large workspaces, where the project filenames have been changed prior to load (filename and namespace needs to be synched for case projects) (#9729).
47. **A simulation run will no longer stall if a plot step is entered that is not an integer multiple of the time step (#9736).**
48. Loading multiple cases at once no longer leads to failed definition reference linking (#9689).
49. Pasting a control object with multiple scenarios no longer causes the selected scenario (at the project level) for the control to have its values overwritten by the base scenario values (#9705).
50. When pasting a control object with a scenario into a case that does not have that scenario, the scenario now becomes visible in the scenario dropdown menu (#9701).
51. If you switch a scenario in the project, while actively viewing controls properties, the properties now update to reflect the change of scenario (#9700).
52. The contents of Scripts pane will now update if the *.py file being viewed is edited and saved externally (#9768).
53. PSCAD no longer issues a warning when an integer parameter is entered as a valid negative value (#9780).
54. Project and application-level substitutions (ex. $\$(Namespace)$) are now available anywhere that global substitutions are used (#9786).



55. **PSCAD will no longer crash when using Python to blackbox a module, and the blackbox creation fails (#9800).**
56. Using the Esc key to close any of the various text file viewing panes (ex. Scripts, Map, Make, etc.), no longer closes the pane view permanently (#9810).
57. **PSCAD no longer crashes if the description of a module is more than 79 characters long (#9773).**
58. Added a single, customizable dialog to handle multiple simulation set tasks that are set to launch from an integrated debugger. This alleviates the need for multiple dialogs (#9775).
59. Animated graphics can consume a significant amount of processing time in a simulation, especially when an expensive to draw object like a graph is sitting on top of an animated component. This specific situation is now handled more intuitively, resulting in simulation speed up (#9834).
60. Invalid EMTDC constant PI changed to the valid PI_ in the script segment, EMTDC Constants symbol list (#9431).
61. A choice parameter default value list on a high DPI monitor, while editing a component definition, will now display properly (#9611).
62. PSCAD now validates its state on startup and on application settings change. If any file required for startup is missing, the user is now notified (#9842).
63. PSCAD now issues a build warning if duplicate sequence numbers are detected (can occur in manually sequenced modules) (#9843).
64. Opening a PSCAD case with a curve id='6710', no longer causes an unexpected New Layer dialog to appear, interrupting the case load (#9877).
65. When specifying a snapshot in the Project tab of the ribbon control bar, the proper 'Save As' dialog is now opened (rather than the 'Open' dialog) (#9831).
66. Rotation of text labels within the component definition graphics editor now function properly, including hotkeys and ribbon buttons (#9901).
67. A transmission segment is no longer forced to re-solve on a simulation re-run after the project file is saved, and there were no modifications made to the transmission segment (#9723).
68. Unmodified, mutually coupled transmission segments are no longer forced to re-solve whenever the simulation is re-run (#9846).
69. Undo/redo now functions properly when adding/removing control interfaces to/from control panels, as well as adding/removing graphs to/from graph panels (#9924).
70. It is no longer possible to drag cable or t-line cross-section definitions from the project definitions list, onto the schematic canvas (#9926).



71. The component wizard now properly releases a newly created, floating component (attached to mouse pointer) if the user decides to go back to the wizard to re-create it (#9912).
72. A newly created component definition (in the project definition list), using the component wizard, is now properly removed/added when the action is undone or redone (#9881).
73. It is no longer possible to add a dead curve to a graph from a component that is not an output channel (#9907).
74. Fixed some minor issues with folder access tracking functionality (#9930).
75. Transmission line and cable definitions are now properly relinked to their respective instances following a definition name change. This had caused issues with instance parameter changes not sticking (#9826).
76. Inter-project paste transfer operations may now be properly undone or redone. Previously, undo/redo simply deleted the instance, not the definitions (#9929).
77. Components newly created via the component wizard or blackbox may now be undone while in a floating state (includes both instance and definition) (#9932).
78. A newly blackboxed component is now properly removed/added when the action is undone or redone (#9940).
79. PSCAD now provides a build warning if two or more identical sequence numbers are detected on a manually ordered, schematic canvas (#9843).
80. Ctrl + left click to select/deselect multiple schematic canvas objects, now functions properly (#9816/#9951).
81. **EMTDC will no longer dead lock if ALL the following criteria are met: There are multiple co-simulation components in a single case; each co-simulation component is connected to the same client id, but different channels; the co-simulation timestep is not the same as PSCAD; the order in which PSCAD processes the co-simulation components is different than the tool.**
82. Components existing as part of an invisible layer, no longer appear visible when printing the schematic canvas (#9954).
83. Recording a script in PSCAD can produce the line: `task = simset.add_task("project_name")`. This function will no longer fail when the script is run (#9976).
84. Two or more instances of PSCAD, each running a project of the same name in debug mode, will now launch multiple instances of MS Visual Studio environments (#10010).
85. Wire mode (and other selection modes) now automatically shut off on creation of a component (ex. clicking a new component button in the ribbon) (#9949).
86. Implicit casting is now properly performed when casting any signal type to complex, or from a complex-type signal to a logical (#9963).



87. Undo/redo now functions properly after component nudging operations (arrow keys) (#8081).
88. Adjusting sticky notes arrows now mark project as dirty and are included in undo/redo (#9910).
89. **Mapping of x-nodes now works properly when there are multiple x-nodes of the same name, but different enabled values (#9996).**
90. The *Echo runtime parameters and options* checkbox now functions properly and will display the runtime parameters in the Runtime Messages when checked (#9978).
91. The co-simulation interface will no longer stall if the co-simulated application exits early. When finalize is called, it will resend the last information with an indefinite expire time. This will ensure that the simulation can continue to run on the last information sent (#9975).
92. Blackbox now checks for the presence of complex signals. If the blackbox compatibility option is set to any version prior to v5.0.0, blackbox will fail, as complex signals are supported only in v5.0.0 onwards (#9984).
93. A plot step smaller than the time step can no longer be entered in the ribbon projects tab. If such a plot step is entered, it is immediately adjusted equal to the time step (#8342).
94. Fixed a rare situation involving a source contention going unnoticed by the netlist compiler. If a data label source by an import tag, is placed on a signal generated by another import tag, no error is issued (#10081).
95. Compare tool definition lists are now presented in alphabetical order (#9970).
96. The run button now remains disabled when selecting to start externally from an integrated debugger (#9936).
97. The consolidator now supports multiple EMTDC platforms. So, when *.dll files are involved, all required *.lib files are included as resources (#10062).
98. Custom layer settings are now properly saved with the project (#10140).
99. Right-clicking while dragging an object on the schematic canvas no longer leaves artifacts (#7749).
100. In the specific instance where a module with a fixed electrical node port, that is connected to an x-node inside on its schematic canvas, which is attached to a switched electrical node, is blackboxed, the node defined by the port will be labeled as a switched node. Prior to this release, there was an inconsistency between a compiled module and a blackboxed module (#10182).
101. Blackboxing any module containing a multimeter that uses a \$VDC:A(1):A(2) statement (i.e. is measuring an instantaneous voltage), will now produce correct, measured results. A simple indexing issue was discovered and corrected (#10190).
102. Drawing layers (incl. quick enable/disable) now function properly within the t-line/cable editor canvas (#10196).



- 103. The parameter grid will now enforce edits if switched to a new parameter grid view, while still in edit mode on the previous view (#10208).
- 104. Blackbox now properly parses directives, such as \$#DIM, in the model-data sections of custom components existing in the module being blackboxed (#10160).
- 105. Paste rename will now scan the original name for an existing suffix number and use that as its starting point (instead of simply appending a '_#'). For example, Sig_1234 becomes Sig_1235 (#10084).
- 106. Renaming the definition of a transmission line or cable, from the workspace tree, no longer causes instances connected to it to forget their parameter values (#9903).
- 107. Component graphic objects can now be pasted properly after copying to the clipboard (#10231).
- 108. Copy/paste functionality now functions consistently with previous versions of PSCAD: A selection always takes precedence over hover when copying. Hover only (with no selection) will copy the hovered object (#10232).
- 109. Python scripts containing Unicode characters will now be displayed properly when viewing in the Scripts pane (#10240).
- 110. Fixed erroneous results that can occur when using the comparison tool (#10248).
- 111. **Mutual coupling now functions properly when including the 'Simplified Underground Cable' component (#10133).**
- 112. If a sticky wire is connected to a bus component AND other components or a non-sticky wire, the bus is now given connection priority. So, the sticky wire will only move with the movement of the bus, if a bus is present at the connection point (#10207).
- 113. Cleaning from the context of the simulation set, now properly clears the build messages in each simulation task (#10234).
- 114. A negative integer is now accepted as a valid integer when entered in the integer constant tag component (#10053).
- 115. Annotations and sticky notes are now supported properly on T-Line and Cable cross-section canvases. Previously the presence of either component would cause solve errors (#9409).
- 116. The C-language interface code, auto-generated by the Component Wizard, now uses ISO C binding when linking Fortran files, as a cleaner, simpler way to bind to the C than the previous manual binding method (#9270).
- 117. MANA element/index order is now consistent and correct between blackbox-generated Fortran and the project Data files, when multiple MANA components appear on the same module canvas being blackboxed (#10353).
- 118. The blackbox option allowing users to set a specific target project (i.e. one other than the current project) has been removed entirely. This functionality can be manually performed by



copying blackbox-generated definitions and resource files to another project (or use Python) (#9059).

- 119. MANA branches are now treated properly by blackbox when MANA components exist within a multiple-level module hierarchy (#10336).
- 120. Wires are now drawn in the correct position following a reversal of its vertexes (#10307).
- 121. Switch and dial actions, during Python script recording, are now properly formatted (#10396).
- 122. PSCAD will now auto-select a valid Fortran compiler in instances when the user has installed an updated version of PSCAD, but the previous version user settings file was set to a compiler that is no longer supported (#9350).
- 123. When using the create compiled library feature (or when blackboxing), PSCAD now chooses the least recent Fortran compiler, within a compiler family, when generating an *.obj or *.lib file. This ensures that the compiled file is compatible with all compilers in that family (#10442).
- 124. Plot step display in the ribbon control bar will now change accordingly if the plot step is changed via the *var_plotstep* component (#10126).
- 125. NDProtocol handles are no longer accumulates when running very large PNI simulations consecutively over RDMA. This accumulation was causing performance issues (#10507).
- 126. **PSCAD no longer crashes if the meta-file (*.psmx) is corrupted (#10572).**
- 127. On paste-transfer of a component that references a definition without a namespace, the components definition reference is now correctly resolved (#10636).
- 128. Transmission segment namespace reference is now properly modified when the namespace name changes (#10625).



PSOUT FILES AND READER UTILITY

Bug Fixes

The following is a list of bugs that were discovered and fixed since the latest, previously released product.

1. **When converting *.psout format to older *.out format, the converter now exports properly with 10 traces plus a domain, rather than 9 traces plus a domain. This caused a serious issue when importing into Enerplot (#9539).**
2. *.psout files may now be read from or written to by multiple programs simultaneously. This means, for example, that you may now update *.psout file data in Enerplot, while PSCAD is generating the data for the same file (#9695).
3. **When writing EMTDC output as *.psout format, while using the GFortran compiler, stored animated graphics information is no longer corrupted (#9682).**
4. **Active graphics traces are no longer overwritten after 128 active graphics actions, when stored within the *.psout file format (#9770).**
5. The PSOUT reader now overrides the Microsoft Windows Regional Settings on Export. Regional settings can lead to invalid files, as the format may not match expected file format (ex. using ‘,’ instead of ‘.’ as a decimal) (#9995).
6. The PSOUT Utility output converter now properly formats *.out file names identical to that written by EMTDC: Specifically, with regards to the file number increment, when multiple *.out files are generated (#10027).



EMTDC/MASTER LIBRARY

Algorithmic Enhancements

Subsystems Solved in Parallel

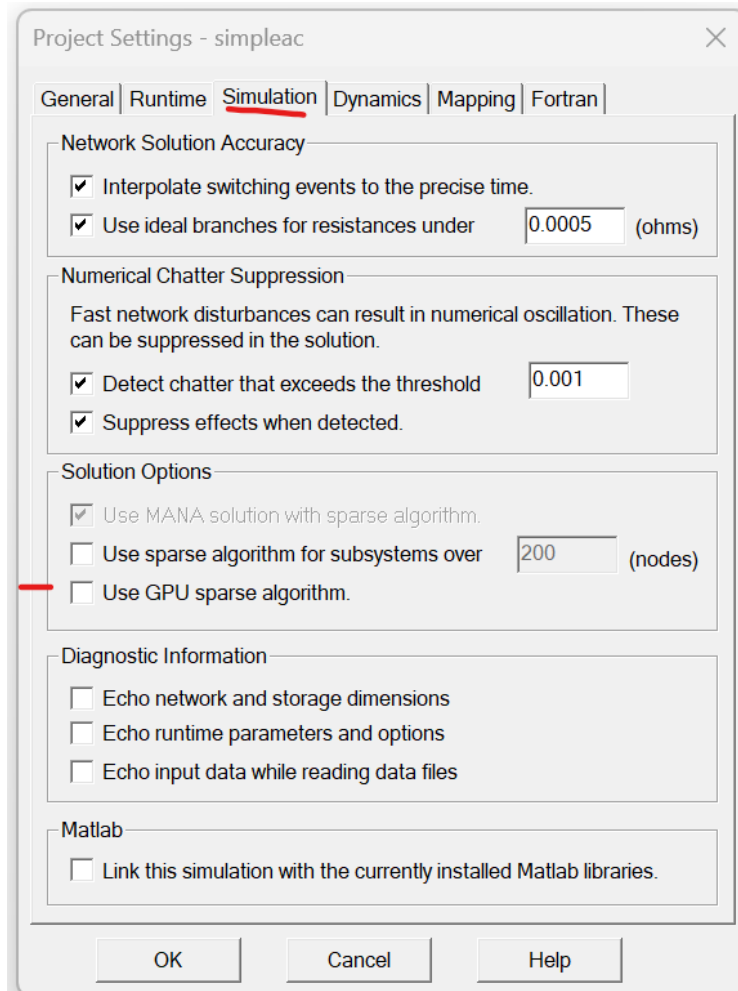
Subsystem solving is now multi-threaded, meaning several subsystems are solved simultaneously on different threads, enhancing simulation speed of multi-subsystem projects.

Improved Ideal Branch and Transformer Processing

Ideal branch and transformer processing is now improved by utilizing cached memory for some of the computations, enhancing simulation speed. A speed improvement is specifically noticeable in projects that contain many ideal branches and/or transformers.

GPU Sparse Algorithm Enabled for EMTDC (#10037)

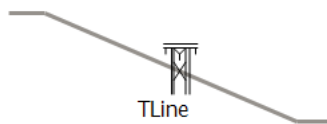
EMTDC may now utilise GPU processing power via an alternative sparse algorithm, in addition to the legacy sparse algorithm already inherent to EMTDC. To enable this feature, simply toggle the associated project setting.



New/Upgraded Models

Multi-Rate (or Multi-Timestep) Transmission Line Upgrade (#6469)

The transmission line configuration component has been upgraded to support multi-rate, parallel network interface (PNI) simulations (i.e. different simulation timesteps at each of its ends). There are no additional parameter settings to configure. Simply connect the PNI simulation as you normally would and run.

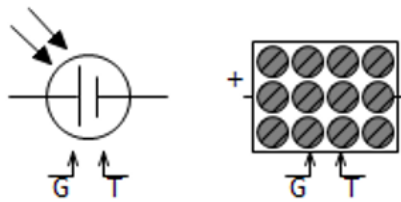




Note that at the time of release, multi-rate functionality is only available when using *foreign ends*. In other words, only PNI simulations that are run from within a single instance of PSCAD may make use of multi-rate (PNI simulations between multiple instances of PSCAD, using *alien ends* are not supported). The only other limitation is that the transmission line model being used must be selected as the Bergeron model.

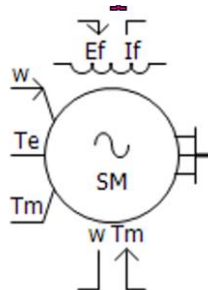
New Photo-Voltaic (PV) Source (#9922)

A newer PV source model (*photovoltaic_source2*) is added to the master library. Unlike its predecessor, this component can take typical data available in datasheets. The data required by the previous model was not easily available to the user.



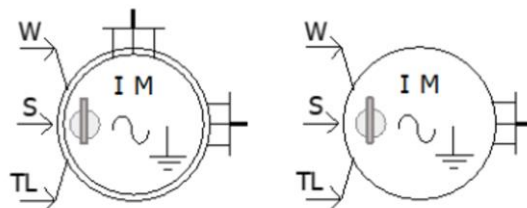
New Synchronous Machine with Simplified Input

An alternate synchronous machine model with a simplified interface, is included in the master library. The model uses the same routine as the original *synchronous_machine* model; the difference is only in the interface.



New Universal Induction Machine with Simplified Input (#8953)

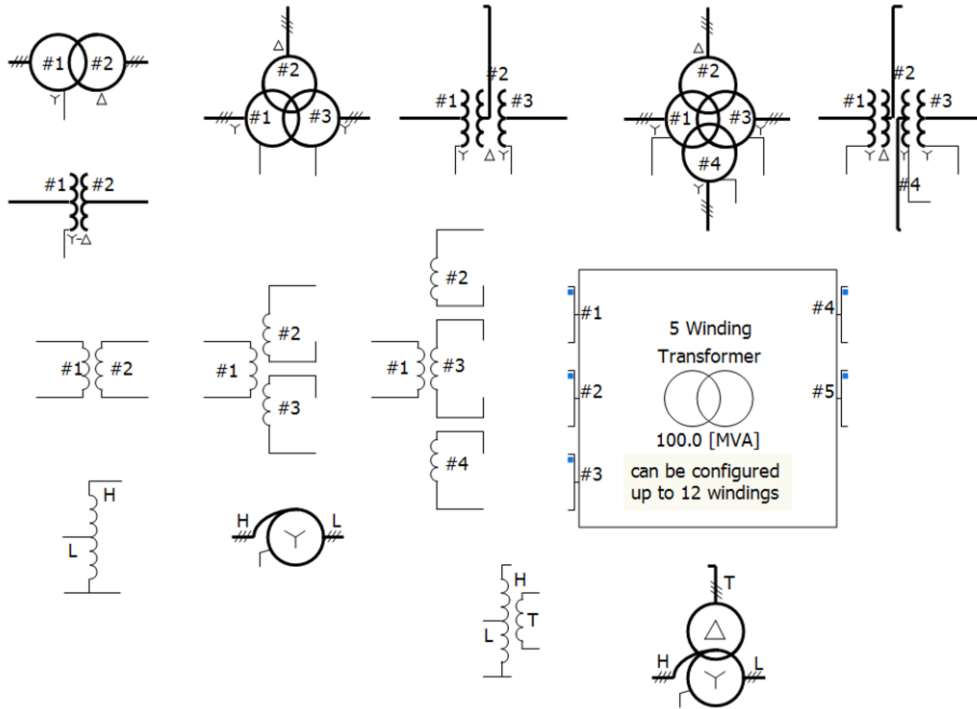
This component is a universal version of the induction machine model (introduced with v5.1.0), which can be used to model either a wound-rotor or a squirrel-cage induction machine. Data input via parameters has also been simplified somewhat. It also supports Type 40 format.





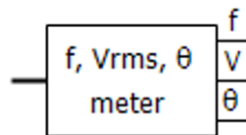
All Classical Transformers, Including Auto Transformers, Updated

All classic transformers in the master library (except for the 1-Phase, N-Winding Transformer) have been updated to include a more simplified set of input parameters.



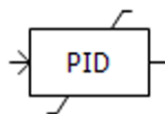
Frequency/Phase/Magnitude Meter Component

New meter component added that calculates the fundamental frequency (f), RMS magnitudes (V), and the phase angles (θ) of a single-phase or a three-phase AC voltage input.



New PID Controller Component

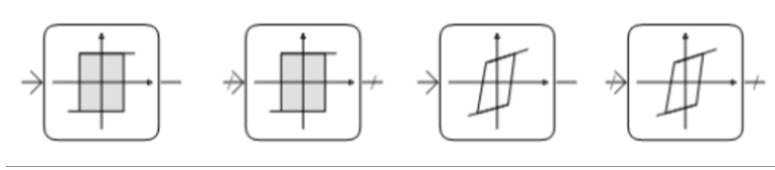
A new component that models a Proportional – Integral – Differential (PID) controller is added to the master library.



New Hysteresis Controller Component

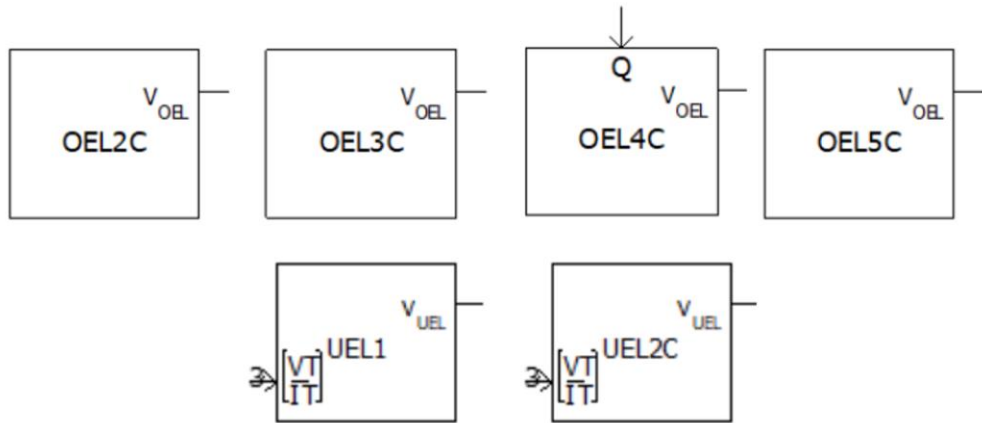


A new hysteresis controller that takes an input and produces an output based on predefined thresholds, has been added to the master library. The component switches the output between two states when the input crosses these thresholds, preventing rapid fluctuations.



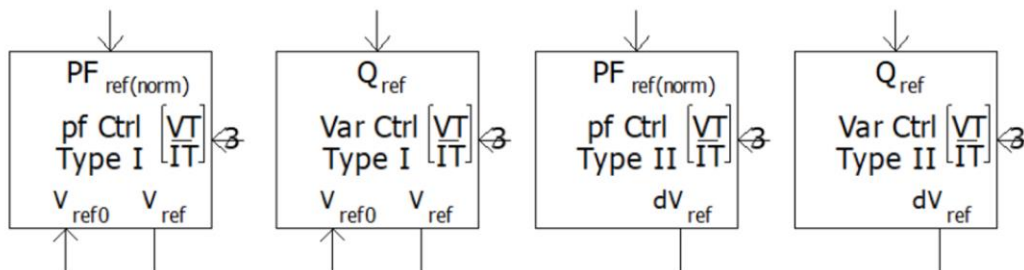
New Over and Under-Excitation Limiter Components

Two new components that model over and under-excitation limiters (conform to the IEEE Standard 421.5-2016) have been added to the master library.



New Power Factor and Reactive Power (VAR/PF) Controller and Regulator Models

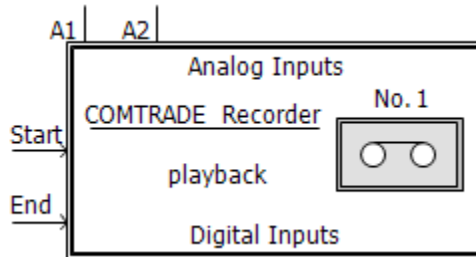
Two new components that model Power Factor and Reactive Power (VAR/PF) Control (conform to the IEEE Standard 421.5-2016) have been added to the master library.



New COMTRADE Recorder Component (#9808, #9809)

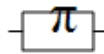


The old RTP/COMTRADE Recorder component has been replaced by a new, COMTRADE only component in the master library. All references and functionality to the Real Time Playback (RTP) product, and COMTRADE 91 formats have been removed. Only COMTRADE 99 formatted output is supported.



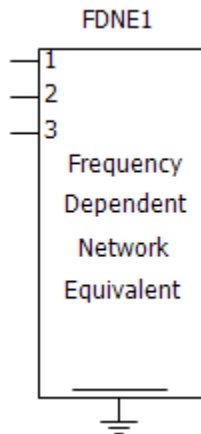
Single-Phase PI Section (#6669)

A new component definition called *pi_section* has been added to the v5.1 master library. This component can be used as either a single-phase or three-phase pi-section.



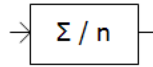
FDNE Component Upgraded to Use Optimized Matrix Storage (#10076)

A new version of the FDNE component definition called 'fdne2' has been added to the master library (the original 'fdne' definition is now deprecated). The new version includes an optimized storage matrix.



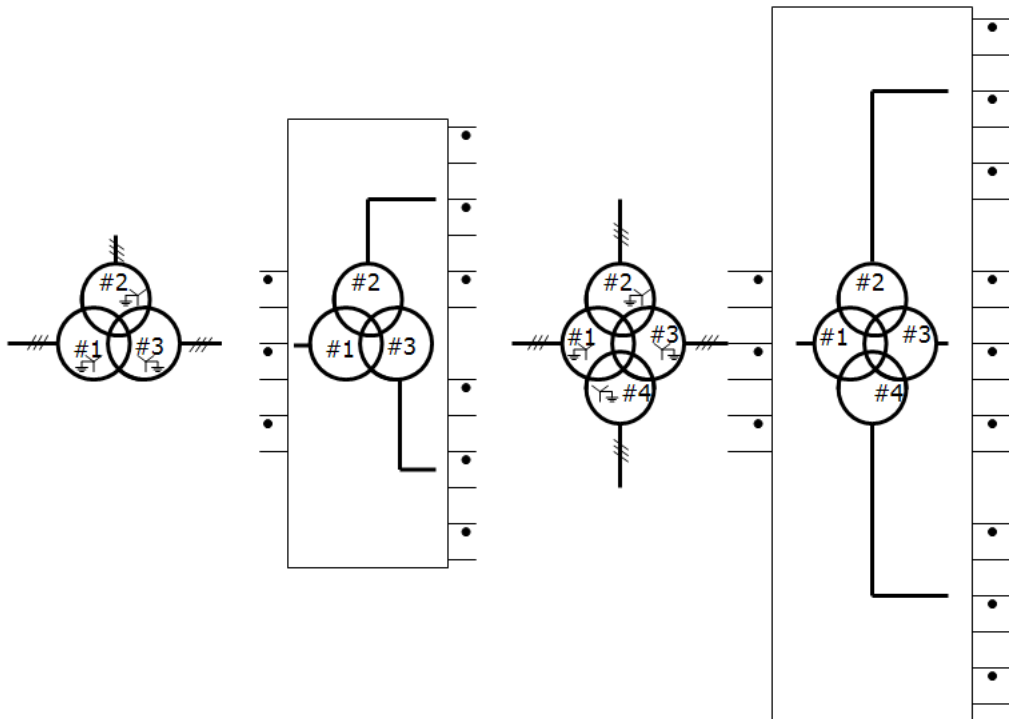
Moving Average Filter (#8764)

A new component definition called *moving_average* has been added to the v5.1 master library. Given a defined moving window and number of samples within it, this component takes a moving average of its input signal.



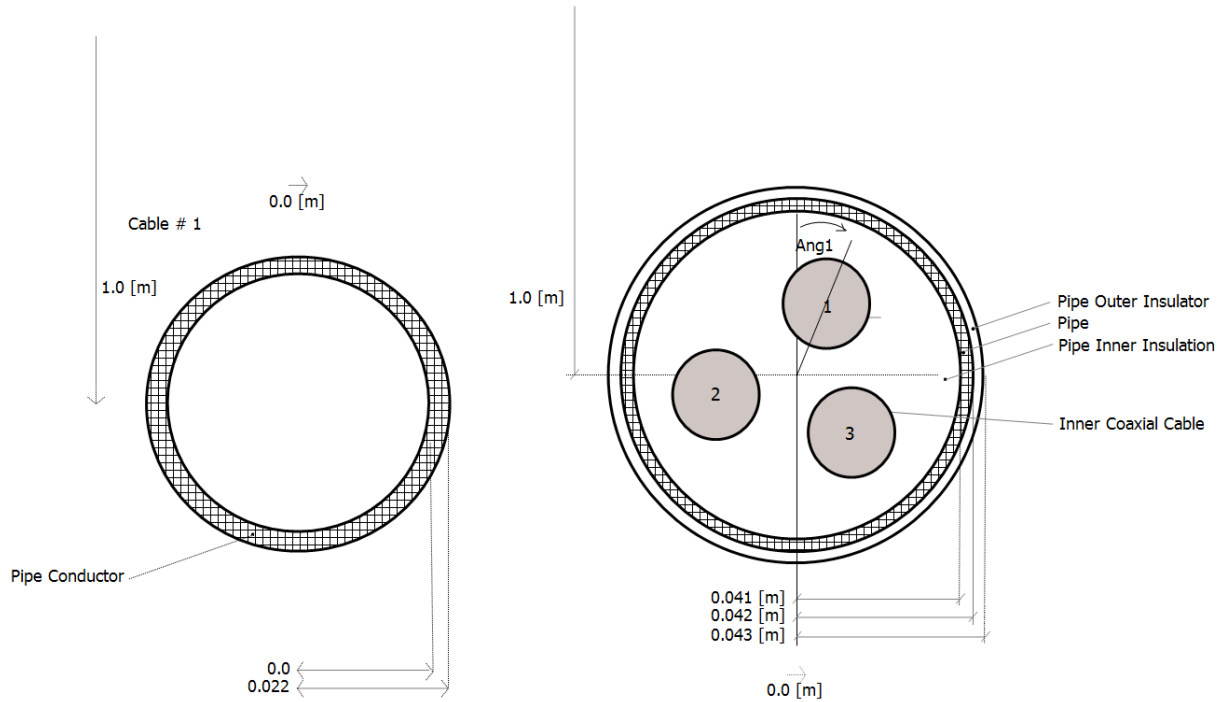
New Duality-Based Transformers (#10381)

Two new, duality-based transformer components have been added to the master library: The duality-based, 3-Phase 3-Winding and the 3-Phase 4-Winding Transformers.



New Pipe-Type Cables (#10369)

Two new pipe-type cable components have been added to the master library, both based on the existing pipe-type cable: The first is a pipe-only, and the second, a simplified pipe-type component. Both are added to alleviate some headaches in data entry.



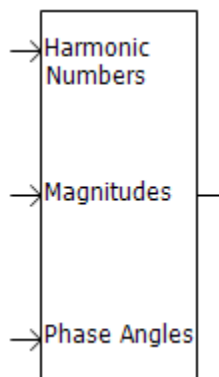
Three New PLL-Based Examples Added (#7299)

Three comprehensive example projects, which demonstrate the usage of the PLL master library component, have been added to the list of released examples. They reside within the CSMF/PLL folder.

Functionality Deficiency Fixes

IEC Standard Input Added to the Harmonics with Different Magnitudes Component (#7873)

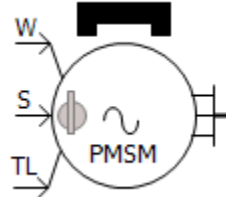
The IEC Standard has been incorporated into the input format for the 'harmdiffmag' component. Specifically, the format follows IEC Standard 61400-21-3, specifically Tables 1 and 2 on Page 23.





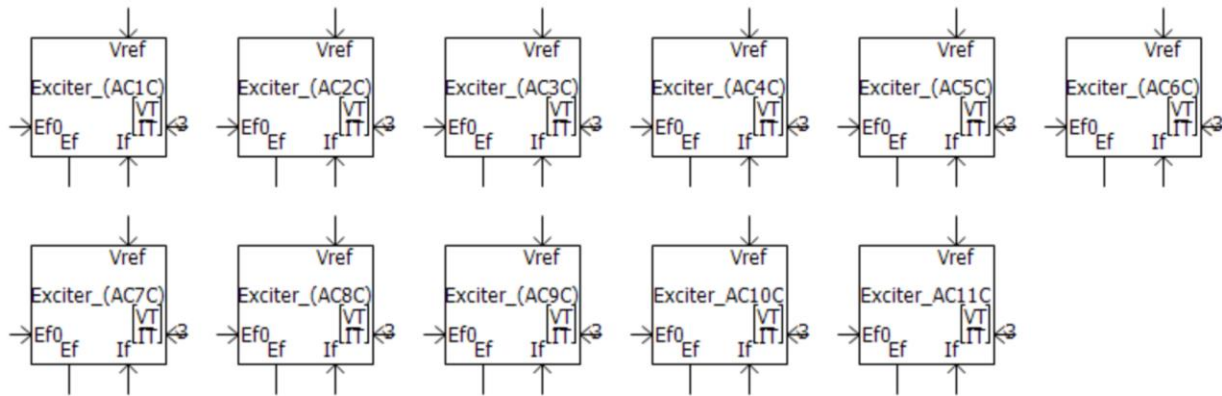
New Data Entry Format for the Permanent Magnet Machine (#2855)

The PMSM now includes two different data input formats: *Equivalent Circuit* and *Dynamic Model*. Equivalent Circuit format represents the electrical characteristics of the machine using parameters like resistance (R) and reactance (X). Dynamic Model format describes the time-varying behavior of the machine, including key aspects such as transient time constants and transient reactance.



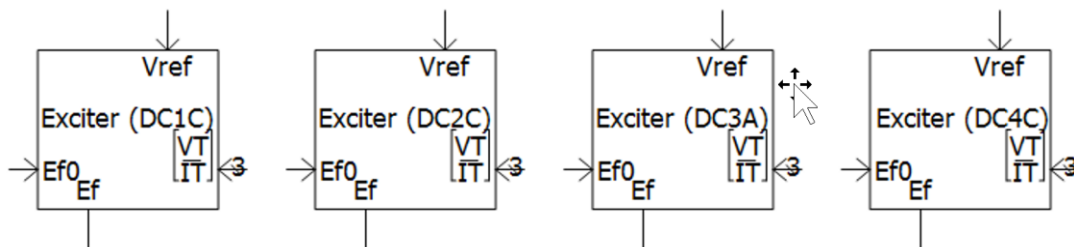
AC Exciters (2016) Update (#10441)

The AC exciter component has been updated with some minor parameter changes.



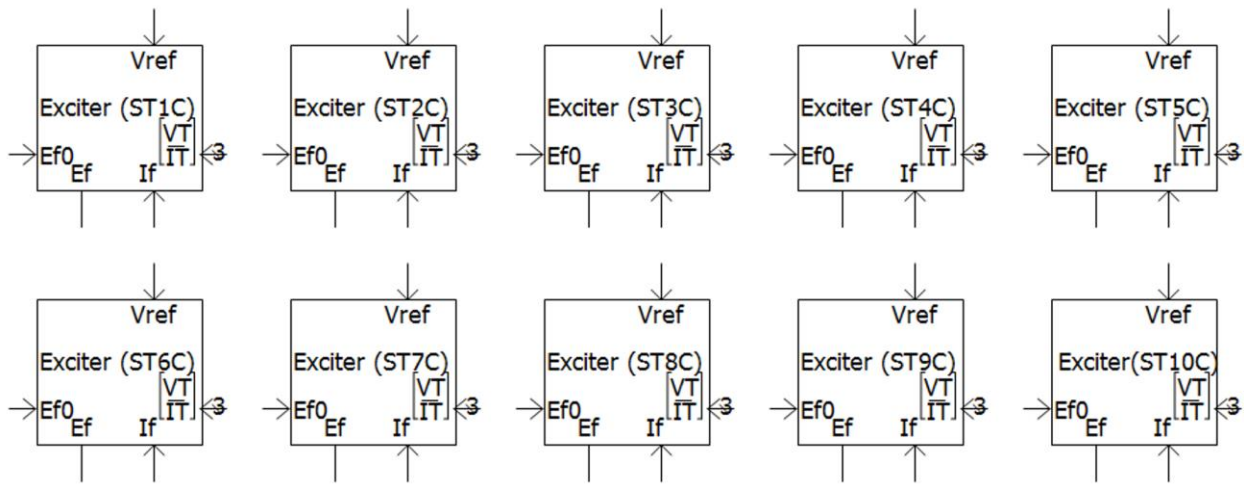
DC Exciters (2016) Update (#10441)

The DC exciter component has been updated with some minor parameter changes.



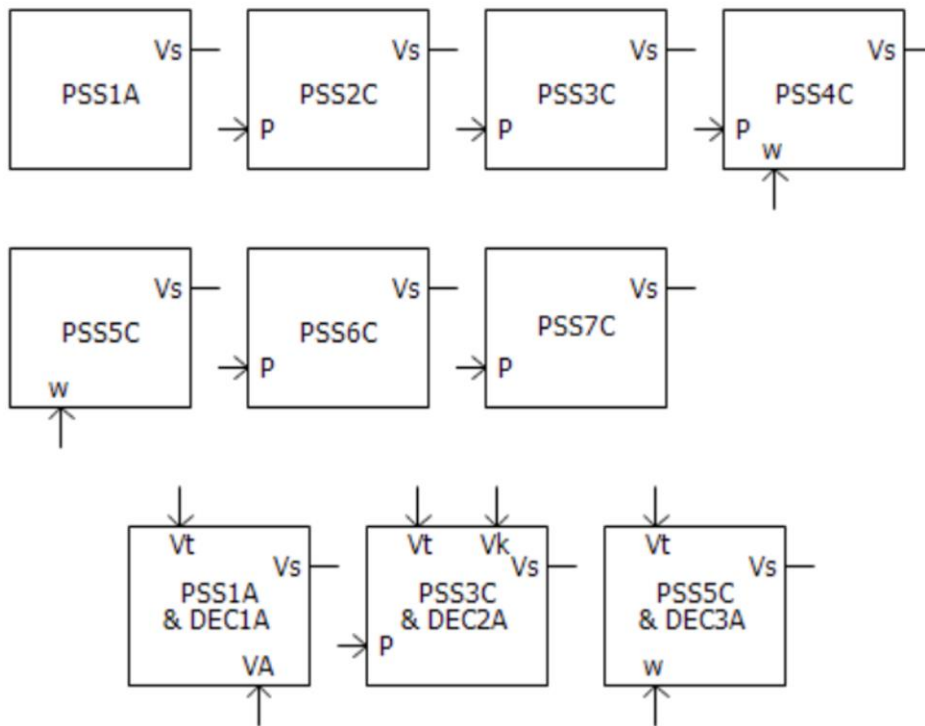
Static Exciters (2016) Update (#10441)

The Static exciter component has been updated with some minor parameter changes.



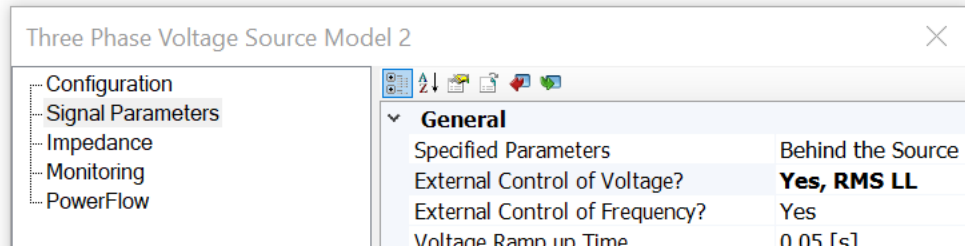
Power System Stabilizers (2016) Update (#10562)

The Power System Stabilizer component has been updated to conform to the IEEE Standard 421.5-2016.



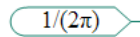
RMS Option for External Control of Voltage Source (#2885)

External voltage control in this component may now be accomplished via a line-to-line RMS signal, in addition to the existing line-to-ground peak format.



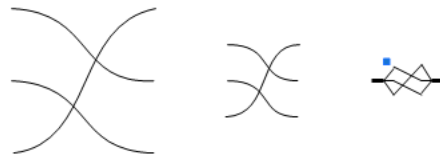
Greek and Math Symbols Displayed in EMTDC Constant Tag (#9068)

PSCAD V5 supports Unicode, so the graphics in the *emtconst* master library component have been updated to display actual Greek characters and math symbols, instead of spelling them out.



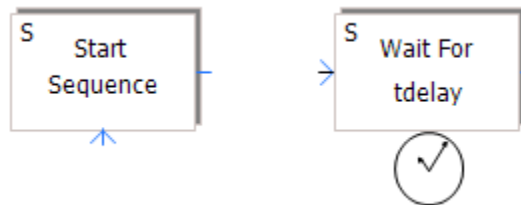
Twist Component Compact View (#8811)

The new Twist component now has a compact view that matches with compact breakout, as well as a single line view. An additional option is also provided to break the nodes into two, distinct sets by inserting an ammeter between them. This would be helpful when both ends are connected to breakout components.



Sequencer Start and Wait Components Now Edge Triggered (#9517)

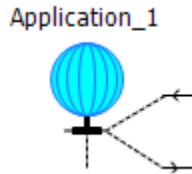
The Sequencer Start and Sequencer Wait master library components have been modified to function on and edge trigger, rather than a level trigger. This was done to resolve a sequencing issue that could occur under the right conditions.



#COSIM Directive in Co-Simulation Component (#8948)



A new directive has been added to the co-simulation component to act as an alternative identifier to the component definition name. This allows users to create and use their own, custom co-simulation components.



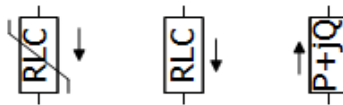
```
#COSIM $client_id $channel_id $s_dim $r_dim
```

Where:

1. *Client ID*: The ID of the application, with which it will be co-simulating.
2. *Channel ID*: The ID of the channel in the application.
3. *Send Size*: The size of the data that this component will send to the other application in terms of total of real numbers.
4. *Receive Size*: The size of the data that this component will receive from the other application in terms of number of real numbers.

Current Measurement Added to RLC Branch Components (#9517)

Components *var_impedance*, *rtcbranch* and *fixed_load1* are modified to provide current measurement through the branch as an internal output. The graphics of each component was also updated to show the direction of current measurement.



Default Capacitive Branch Behaviour in Variable Impedance (#9523)

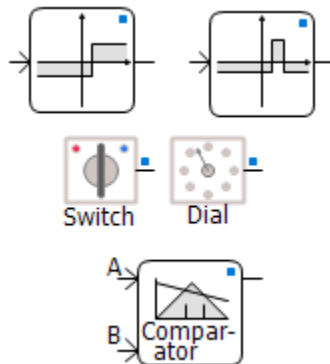
The *var_impedance* component has been modified to give the user a choice on how to treat zero capacitance as a short-circuit or an open-circuit. It is defaulted to a short-circuit to provide backwards compatibility with PSCAD v5.0.





Graphical Display Indicates Conversion of Output to Nearest Integer (#9956)

Some components in the master library (ex. compar, compare, range, var_pot, var_switch, etc.) have the option to convert the output to the nearest integer. Now, a graphical indicator (a small blue box) when this option is applied.



Bug Fixes

1. Added new parameter to the dependent sources to allow an easy way to invert the components direction up or down (#9242).
2. The FFT component now has a proper id call, the lack of which was causing generated messages to point to other components (#8944).
3. The FFT component now limits the tracked frequency to the maximum permissible, set by the project time step, avoiding a runtime error (#8966).
4. The Multimass component now supports fixed power share, in addition to fixed torque share (#8837).
5. **Fixed an issue with the PGBNAME function in EMTDC, which caused a simulation to get 'stuck' when many output channels exist in a project (#9673).**
6. Unrealistic saturation parameter data in the AC and DC exciter components is now caught and the user is notified with an error message (#9919).
7. When a constant signal is used as an analog input to the COMTRADE recorder component, the COMTRADE file generated is now correctly written (#9913).
8. The correct information message and time step value are now provided in situations where the COMTRADE recorder component's recording time step is less than the simulation time step (#9909).
9. When a multiple run with a tandem line, with other lines present is launched, the simulation no longer fails on the 2nd run (#9934).



10. The FWAV22 routine and its variants no longer creates runtime message flak for minor imbalance of phase magnitudes (caused a negative influence on execution speed) (#9942).
11. The fault component now displays N instead of G if the fault neutral point is not directly grounded (ex. A->N instead of A->G) (#9855).
12. EMTDC no longer emits multiple PSCAD_AGR2 statements in runtime messages (#9961).
13. Interpolated output of the delay component no longer gives values outside the input range, when subjected to certain conditions (#9255).
14. A chatter removal step is now only applied, when necessary, when starting from a snapshot. Previously persistence of this step created a slight difference in results (#10046).
15. The UMEC transformer model now produces a correct, user-supplied saturation curve (#2825).
16. Both source models 1 and 2 are now giving the full voltage when the ramp-up time, or time constant, is set to zero (#9977).
17. The computed matrix sparsity limit has been increased by upgrading to INTEGER*8 from INTEGER*4 (which could result in errors if the number of nodes per subsystem is close to 50,000) (#10162).
18. An issue with the LOGICAL storage incrementor has been corrected in ST1C exciter component (#10144).
19. A new choice parameter was added to the surge generator component to allow specification of how maximum steepness is defined, when configured for the CIGRE standard. This allows for arbitrary changes in peak current amplitude (#6352).
20. The current source component default parameter values have been modified to be more realistic (#7229).
21. GFortran compilers no longer fail when the *Fileread* component is set to output the last value, or interpolate, and the simulation length is longer than the data file (#6358).
22. The FDNE order is now properly defined when the component set to input user-defined ABCD parameters (#7153).
23. An additional voltage option is added to the synchronous machine component. The voltage rating may now be specified as line-to-line or line-to-neutral (#4492).
24. The parameter *Capacitor Leakage Resistance* for the *MMC Full Cell* and *MMC Half Cell* components have been updated to allow variable signal input (#6613).
25. The *Ramp Up Method* parameter in both the 3-phase and 1-phase voltage source model 1 can now be selected as First Order Lag or Linear (#10029).
26. The Frequency Scan component will now properly process when t-lines or cables are present that are configured to use manual data entry with multiple frequencies (#10057).



27. The *MMC Half Cell* component has been fixed so that the capacitors of each module can be initialized correctly, as set in the parameter 'Sum of Capacitor Voltages at Time zero' when 'Partial Blocking' is enabled (#10132).
28. Fixed an insufficient storage allocation issue in the ZDelay component script, which would arise if several delay components were being used (#10146).
29. The FDNE now functions properly when used with multiple frequency scanners and FDNEs (#8774).
30. Line current measurement parameters in the 3-Phase, 2-Winding Transformer are now always enabled (and give the correct value) regardless of the winding configuration (#10264).
31. Upper value maximum of leakage impedance has been relaxed to allow more realistic data entry (i.e. > 1.0 pu) in some applications (#10194).
32. The xy transfer function component expects x-axis data to be entered in ascending order. However, no check was being done to inform user if this is not so. A check has now been implemented to produce a warning (#10261).
33. The Rank component now gives the correct output when used inside BEGIN routines (#10311).
34. Spring constant entry in p.u. for the multi-mass component now provides accurate input, as verified via eigenvalue analysis (#10250).
35. The z-domain, nth order transfer function component no longer encounters an error when handling multiple dimension input signals (#10394).
36. The 'Conductors to Eliminate' parameter of the Pipe-Type Cable component will no longer affect the cable calculation even if it is disabled (#10471).
37. Detailed output file viewer now displays the correct units in the detailed output drop list (#10451).
38. **Exciter components no longer allow SCL (oel) to go as take-over and SCL (uel) to go to the summation point at the same time (not allowed according to the standard) (#10441).**
39. **EMTDC results are no longer slightly different when a project is started from a snapshot, compared with one started from the datafile (or t=0.0). Logic for an additional interpolation step is now performed only if needed (#6225).**
40. **The ST6C-type exciter now properly accounts for the influence of the PSS input (#10530).**
41. Fixed an error in parameter visibility logic in the xfmr_Nw transformer: The leakage reactance (4-7) parameter was never visible when the transformer is set to 7 windings (#10635).
42. Some projects no longer generate linking errors related to *frexp*, which is included in *bhlibucrt.lib* (from Visual Studio) and *libmmt.lib* (from Intel) (#10666).



- 43. The frequency scan master library component now works properly when there exist t-lines or cables in the network, configured to use the multiple-frequency, manual-data entry method (#10673).
- 44. The ic_engine master library component no longer fails if number of misfiring cylinders is greater than 3 (#10778).



LINE CONSTANTS PROGRAM (LCP)

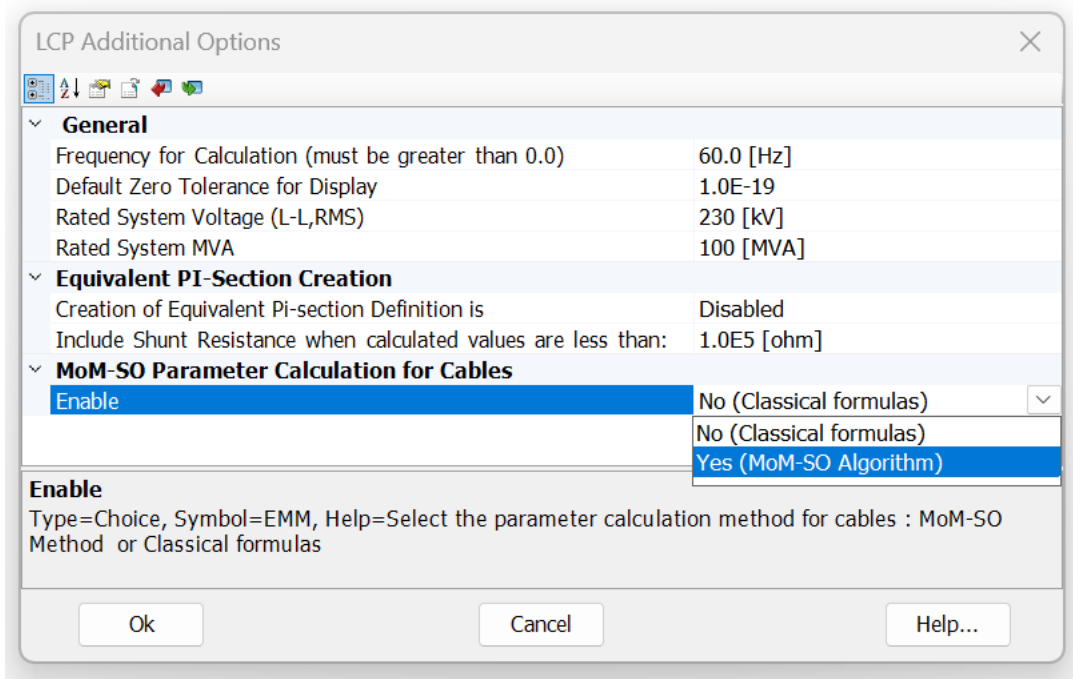
Algorithmic Enhancements

MoM-SO Method Added (#9102, #10157)

Users may now opt to use MoM-SO methodology, instead of the classical formulas, to more accurately solve cable constants where proximity effects in the conductors are prominent.

MoM-SO is a semi-analytical method used to compute cable parameters more broadly, considering proximity effects for the correct prediction of electromagnetic transients in cables that are placed close to one another, such as submarine cables.

MoM-SO is controlled in PSCAD via the Additional Options component: An optional component that is placed on the cable cross-section, along with the model, ground return, and the cable itself.



Bug Fixes

1. Modified formatting of LOAD FLOW RXB DATA to show three decimals, instead of two (#9987).
2. Pi-section component creation now functions properly when the line/cable is set to manual data entry at multiple frequencies (#10058).
3. Fixed a singularity problem that would occur when underground cables and aerial lines exist within the same right-of-way, due to an incorrect starting index (#10156).



4. Long-line corrected data entered in the Manual Entry of Y,Z component, is now handled properly (#9357).
5. When the Bergeron line model is used with the Manual Entry of Y,Z component, it is now considers the frequency for loss approximation properly, in the generated constants (*.tlo) file (#9989).
6. When the Bergeron line model is used with the Manual Entry of Y,Z component, it is now considers the travel time interpolation and reflectionless line properties properly, in the generated constants (*.tlo) file (#9938).
7. The LCP now issues a proper error message when a user attempts to solve a transmission segment contained the Frequency-Dependent (Phase) model with the Manual Entry of Y,Z component, when the Manual Entry component is not set to multiple frequency input (#10201).
8. The correct port numbers are now used in the Matrix Fill section when the transformer is set to delta-lead configuration. This is for both 3-phase, 2-winding and 3-phase, 3-winding models (#10326).



LICENSING AND UTILITIES

Bug Fixes

1. Installed Intel Fortran compilers are now listed in order of their release date when viewing selecting a compiler (#10364).
2. All support required for the GFortran 13.2 compiler has been added (#10005).
3. All support required for the Intel Fortran 19.2 compiler has been added (#10041).
4. All support required for the Intel Fortran (ifx) 2025 compiler has been added (#10314).
5. All support required for the Intel Fortran 2024.0 OneAPI compiler has been added (#9878).
6. A new setting, when enabled, will set a license certificate to be returned only when the last running instance of PSCAD, attached to that certificate, is closed (#10325).
7. A new setting, when enabled, will set PSCAD to first check to see if a license certificate already exists, before trying to acquire a certificate. This will help alleviate 'stuck' licenses in many scenarios (#10320).



APPENDIX: COMPLETE LIST OF NEW MACROS

- **\$(FileVersion)**: Substitutes the last version of PSCAD used to save the current state of the project file (ex. 4.6.3).
- **\$(WorkspaceName)**: Substitutes with the workspace name.
- **\$(Scenario)**: Substitutes the name of the active scenario. Failing that, it will substitute the base scenario name.
- **\$(WorkspaceDir)**: Substitutes with the directory for workspace.
- **\$(WorkspaceTempDir)**: Substitutes with the directory for temporary workspace level resources. This is where a co-simulation configuration file will reside when launching from a simulation set.
- **\$(VersionName)**: The name of the version of PSCAD that is being run
- **\$(Version)**: The name of the version of PSCAD that is being run
- **\$(HelpURL)**: The location of the PSCAD help on the internet.
- **\$(HelpURLScripting)**: The location of the PSCAD Scripting help on the internet.
- **\$(HomeDir)**: The installation folder for this PSCAD installation
- **\$(HomeURL)**: The home URL for the Website for this product.
- **\$(LanguageDir)**: The folder for language specific content.
- **\$(LibDir)**: The location of the custom library folder.
- **\$(LocalDir)**: The folder for the user local resources for this user.
- **\$(LocalHost)**: The local host IP Address
- **\$(PscadPlatform)**: The x86 or x64 Platform that version of PSCAD is using.
- **\$(PublicDir)**: The location of the public documents folder on this machine.
- **\$(RootKey)**: The Key name for the Registry entry for this installation.
- **\$(TempDir)**: The location for the temporary directory for this user.
- **\$(UserDir)**: The user directory for this user.
- **\$(Configuration)**: Will resolve to 'Debug' or 'Release' depending on the debug status of this build of this project.
- **\$(CreateDate)**: The date that this case was created.
- **\$(CreateTime)**: The time that this case was created.
- **\$(Creator)**: The user who created this case.
- **\$(ReviseDate)**: The last day that this file was changed.
- **\$(ReviseTime)**: The last time that this file was changed.



- **\$(Reviser)**: The last user who changed this case
- **\$(Namespace)**: The namespace of this case.
- **\$(Scenario)**: Substitutes the name of the active scenario. Failing that, it will substitute the base scenario name.
- **\$(ProjectDir)**: The folder path that this case resides in.
- **\$(ProjectFile)**: The name of the file that this case is saved in.
- **\$(LanguageDir)**: Location for language specific
- **\$(Compiler)**: Environment Variable name for the currently selected Fortran version
- **\$(EmtPlatform)**: Environment Variable name for the platform of the currently selected Fortran version
- **\$(CompilerFullName)**: Environment Variable name for the name of the currently selected Fortran version
- **\$(CompilerKey)**: Environment Variable name for the key of the currently selected Fortran version
- **\$(CompilerVersion)**: Environment Variable name for the version number of the currently selected Fortran version
- **\$(CompilerExe)**: Environment Variable name for the executable location of the currently selected Fortran version
- **\$(CompilerBatch)**: Environment Variable name for the batch file name of the currently selected Fortran version
- **\$(CompilerArgs)**: Environment Variable name for the batch file arguments of the currently selected Fortran version
- **\$(FortranInstallDir)**: Environment Variable name for the installed folder path of the currently selected Fortran version
- **\$(FortranExePath)**: Environment Variable name for the path to the executable of the currently selected Fortran version
- **\$(FortranSuffix)**: Environment Variable name suffix to use for compiled file of the currently selected Fortran version
- **\$(CPlatform)**: Environment Variable name for the platform of the currently selected 'C' version
- **\$(CFullName)**: Environment Variable name for the name of the currently selected 'C' version
- **\$(CKey)**: Environment Variable name for the key of the currently selected 'C' version
- **\$(CVersion)**: Environment Variable name for the version number of the currently selected 'C' version



- **\$(CExe)**: Environment Variable name for the executable location of the currently selected 'C' version
- **\$(CInstallDir)**: Environment Variable name for the installed folder path of the currently selected 'C' version
- **\$(CExePath)**: Environment Variable name for the path to the executable of the currently selected 'C' version
- **\$(CSuffix)**: Environment Variable name suffix to use for compiled file of the currently selected 'C' version
- **\$(MatlabPlatform)**: Environment Variable name for the platform of the currently selected MatLab version
- **\$(MatlabFullName)**: Environment Variable name for the name of the currently selected MatLab version
- **\$(MatlabKey)**: Environment Variable name for the key of the currently selected MatLab version
- **\$(MatlabVersion)**: Environment Variable name for the version number of the currently selected MatLab version
- **\$(MatlabExeDir)**: Environment Variable name for the executable folder location of the currently selected MatLab version
- **\$(MatlabDllDir)**: Environment Variable name for the DLL folder location of the currently selected MatLab version
- **\$(MatlabPath)**: Environment Variable name for the path to the currently selected MatLab version
- **\$(MatlabLibs)**: Environment Variable name for the library folder location of the currently selected MatLab version
- **\$(MatlabExe)**: Environment Variable name for the full executable path
- **\$(EmtFolder)**: Environment Variable name for the EMT location for the currently selected compilers
- **\$(EmtLinker)**: Environment Variable name for the Linker executable location for the currently selected compilers
- **\$(EmtLinkerPath)**: Environment Variable name for the Linker location for the currently selected compilers
- **\$(BuildPath)**: Environment Variable name for the build path for where the simulation is built
- **\$(ProjectFortranAdditonalOptions)**: Environment Variable name additional Fortran compile options provided.
- **\$(ProjectCAdditonalOptions)**: Environment Variable name additional 'C' compile options provided.



- **\$(ProjectLinkerAdditonalOptions):** Environment Variable name additional Linker compile options provided.
- **\$(ProjectPreprocessorAdditonalOptions):** Environment Variable name additional preprocessor options provided.
- **\$(DefaultCompiler):** The default value to use for the environment variable for the Fortran Compiler
- **\$(DefaultEmtPlatform):** The default value to use for the environment variable for the Fortran Compiler Platform
- **\$(DefaultCompilerFullName):** The default value to use for the environment variable for the Fortran Compiler Name
- **\$(DefaultCompilerKey):** The default value to use for the environment variable for the Fortran Compiler Key
- **\$(DefaultCompilerVersion):** The default value to use for the environment variable for the Fortran Compiler Version
- **\$(DefaultCompilerExe):** The default value to use for the environment variable for the Fortran Compiler Executable name
- **\$(DefaultCompilerBatch):** The default value to use for the environment variable for the Fortran Compiler Batch File
- **\$(DefaultCompilerArgs):** The default value to use for the environment variable for the Fortran Compiler Batch File Arguments
- **\$(DefaultFortranInstallDir):** The default value to use for the environment variable for the Fortran Compiler Installed Directory
- **\$(DefaultFortranExePath):** The default value to use for the environment variable for the Fortran Compiler Executable Path
- **\$(DefaultFortranSuffix):** The default value to use for the environment variable for the Fortran Compiler suffix extension
- **\$(DefaultCPlatform):** The default value to use for the environment variable for the 'C' Compiler Platform
- **\$(DefaultCFullName):** The default value to use for the environment variable for the 'C' Compiler Name
- **\$(DefaultCKey):** The default value to use for the environment variable for the 'C' Compiler Key
- **\$(DefaultCVersion):** The default value to use for the environment variable for the 'C' Compiler Version
- **\$(DefaultCExe):** The default value to use for the environment variable for the 'C' Compiler Executable Name



- **\$(DefaultInstallDir):** The default value to use for the environment variable for the 'C' Compiler Installed Path
- **\$(DefaultCExePath):** The default value to use for the environment variable for the 'C' Compiler Name Executable Path
- **\$(DefaultCSuffix):** The default value to use for the environment variable for the 'C' Compiler Name suffix extension
- **\$(DefaultMatlabPlatform):** The default value to use for the environment variable for the MatLab Compiler Platform
- **\$(DefaultMatlabFullName):** The default value to use for the environment variable for the MatLab Compiler Name
- **\$(DefaultMatlabKey):** The default value to use for the environment variable for the MatLab Compiler Name Key
- **\$(DefaultMatlabVersion):** The default value to use for the environment variable for the MatLab Compiler Name Version
- **\$(DefaultMatlabExeDir):** The default value to use for the environment variable for the MatLab Compiler Name Executable Folder path
- **\$(DefaultMatlabDllDir):** The default value to use for the environment variable for the MatLab Compiler DLL Folder path
- **\$(DefaultMatlabPath):** The default value to use for the environment variable for the MatLab Compiler path
- **\$(DefaultMatlabLibs):** The default value to use for the environment variable for the MatLab Compiler Libraries
- **\$(DefaultMatlabExe):** The default value to use for the environment variable for the MatLab Exe location
- **\$(DefaultEmtFolder):** The default value to use for the environment variable for the compiler selection EMT Folder Path
- **\$(DefaultEmtLinker):** The default value to use for the environment variable for the compiler selection Linker Executable Name
- **\$(DefaultEmtLinkerPath):** The default value to use for the environment variable for the compiler selection Linker Executable Path