

Quick Start for User Defined Models

Saurav Mohapatra, PhD, PE
saurav@powerworld.com

Last Updated: March 13, 2025



PowerWorld Corporation
2001 South First St
Champaign, IL 61820
(217) 384-6330
<http://www.powerworld.com>
info@powerworld.com

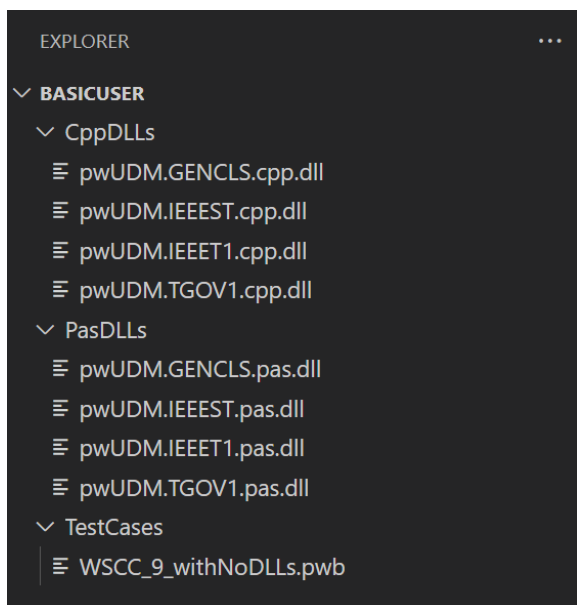
Overview	3
1. Before you begin... ..	3
2. Basic Users	4
3. Advanced Developers	11

Overview

The purpose of this document is to provide a step-by-step guide to using the User Defined Models tab, which is located in the Options menu of the Transient Stability Add On in PowerWorld Simulator. The user defined model feature has been made available from version 17 onward. Initially, this was made available as 32-bit DLLs, and then later as 64-bit DLLs when PowerWorld moved to 64-bit only. This tutorial aims to provide the basic steps for navigating the tabs of interest, loading, inserting, and using user defined models for transient analysis. Section 2 is geared towards basic users, and section 3 will provide a brief note for advanced developers.

1. Before you begin

Download the sample files and projects from the PowerWorld website. The content is shown below.

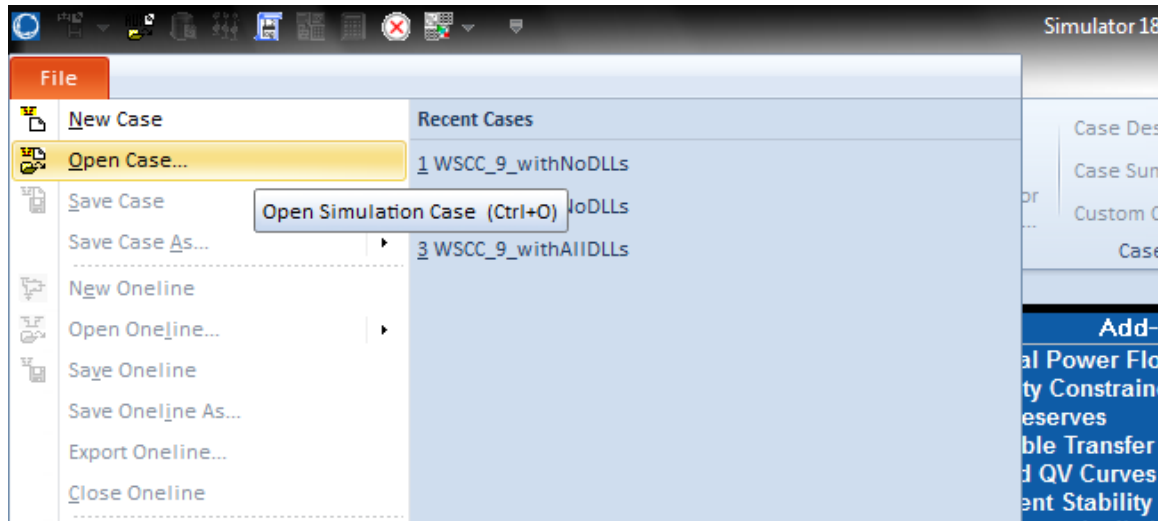


The BasicUser folder contains four folders. The folders CppDLLs, and PasDLLs have functionally identical DLLs of the GENCLS machine, IEEEEST stabilizer, IEEEET1 exciter and TGOV1 governor. In addition, the TestCases folder contains a case for illustrating the use of these DLLs.

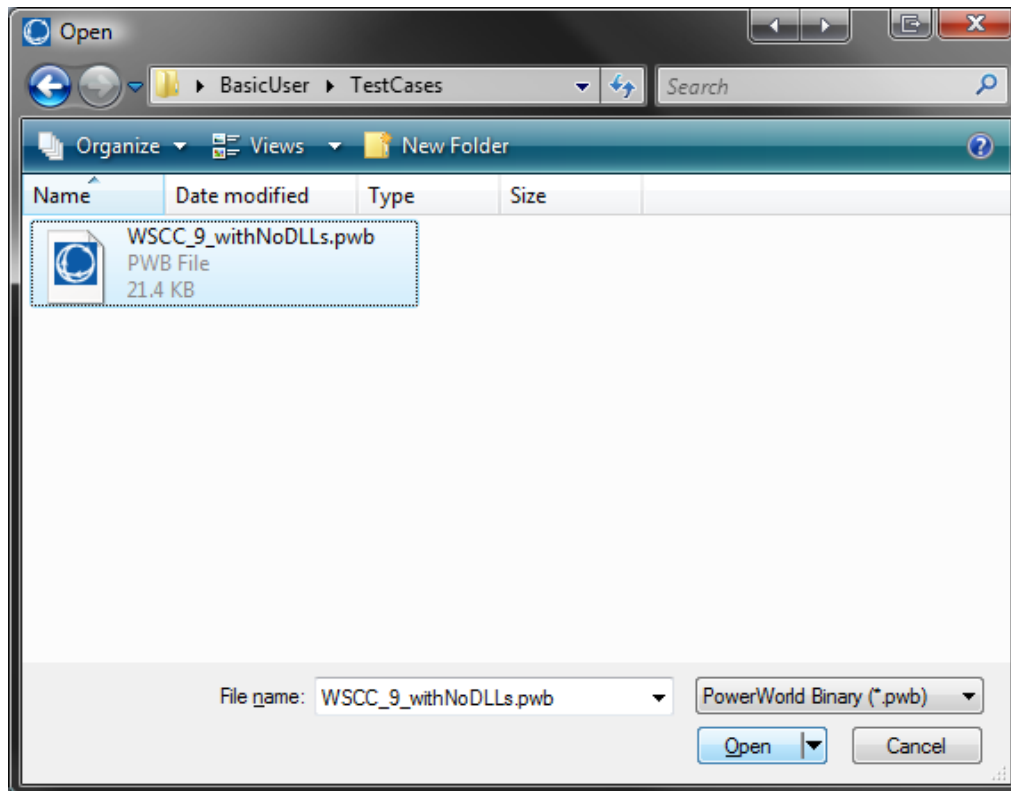
2. Basic Users

It is likely that a basic user will receive a set of DLL files, each with one transient model of interest. (The development of these DLLs is left to the advanced developer) This section will outline the steps in PowerWorld Simulator for loading, inserting and using user defined models.

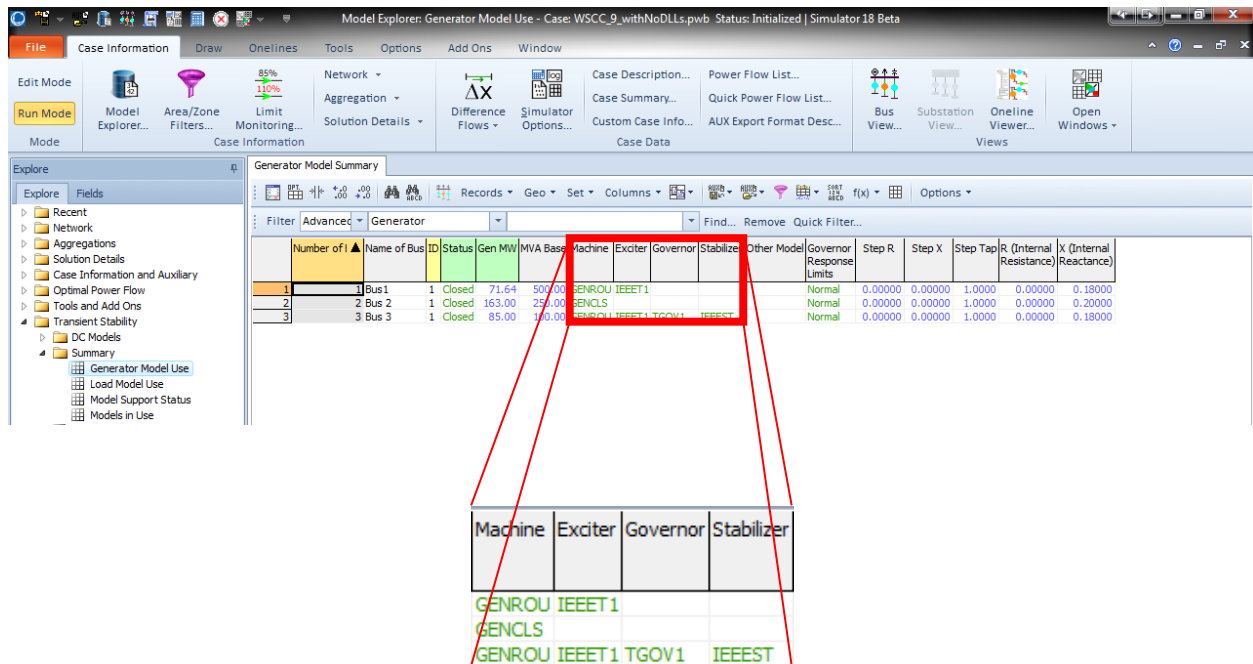
1. Start PowerWorld Simulator, and navigate to File > Open Case...



2. Navigate to BasicUser > TestCases folder. Select the WSCC_9_withNoDLLs.pwb file, and click Open.

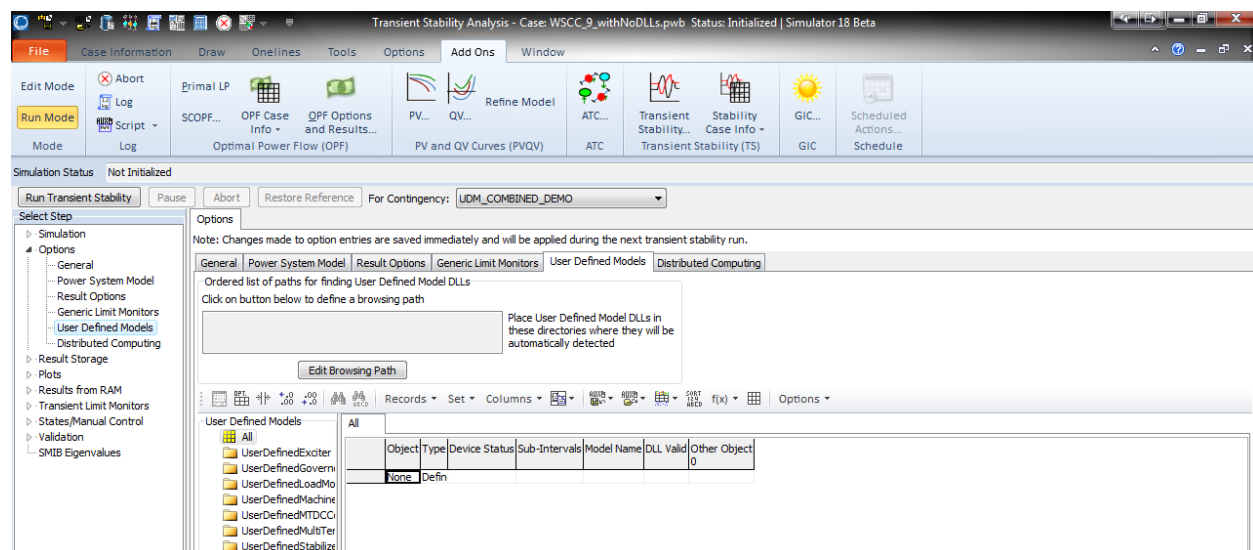


- Once the case loads, navigate to the Case Information tab. In Model Explorer, navigate to the Transient Stability > Summary > Generator model use option in the left-side menu.

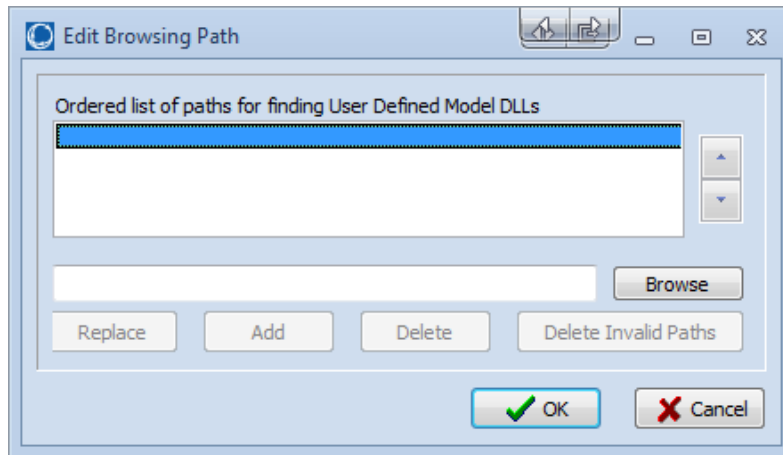


In the Generator Model Summary tab the machine, exciter, governor and stabilizer models in use are displayed. These models are built-in models supported by PowerWorld Simulator. In the subsequent steps, user defined models will be inserted to replace these models.

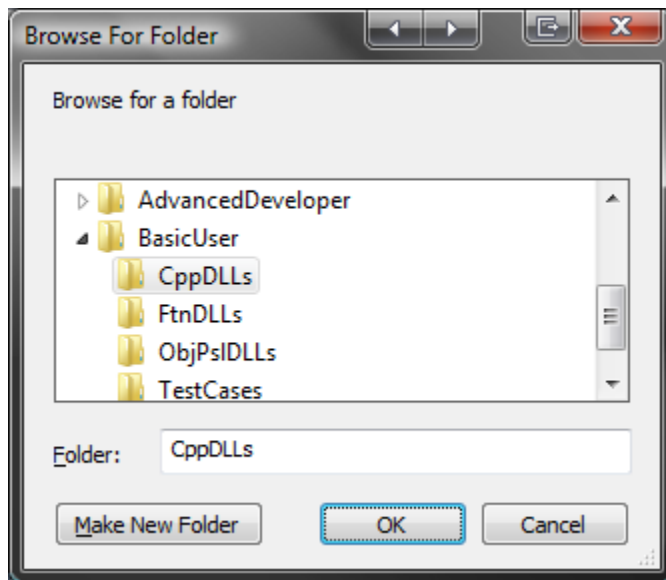
- Ensure that Simulator is in Run Mode. Navigate to the Transient Stability in the Add Ons tab. In the left-side menu, click on Options > User Defined Models. This is currently empty because no user defined models have been selected/inserted.



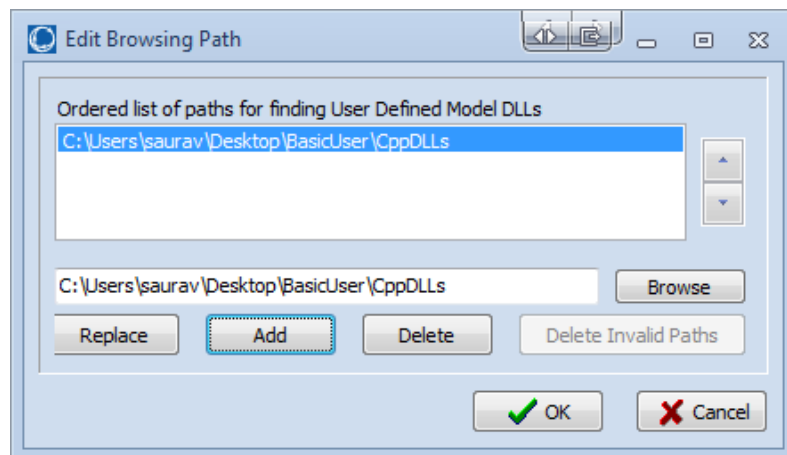
5. Click on the Edit Browsing Path button.
6. In the window below, click Browse.



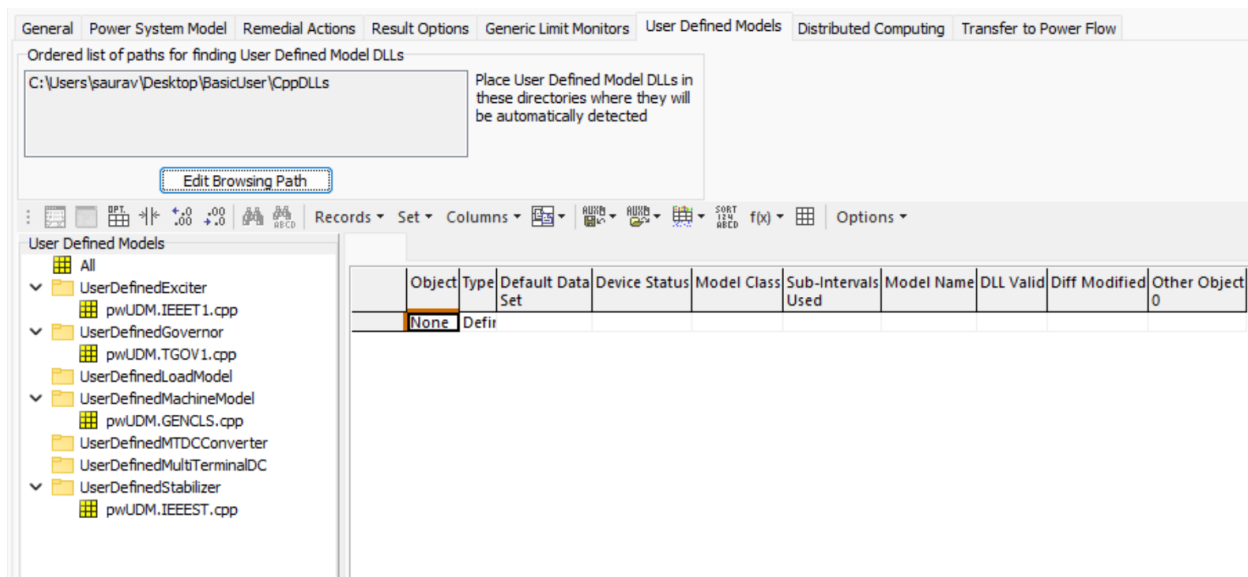
7. Then, browse for the BasicUser > CppDLLs folder. Click OK.



8. Click Add. The path name should appear in the list. Then, click OK.



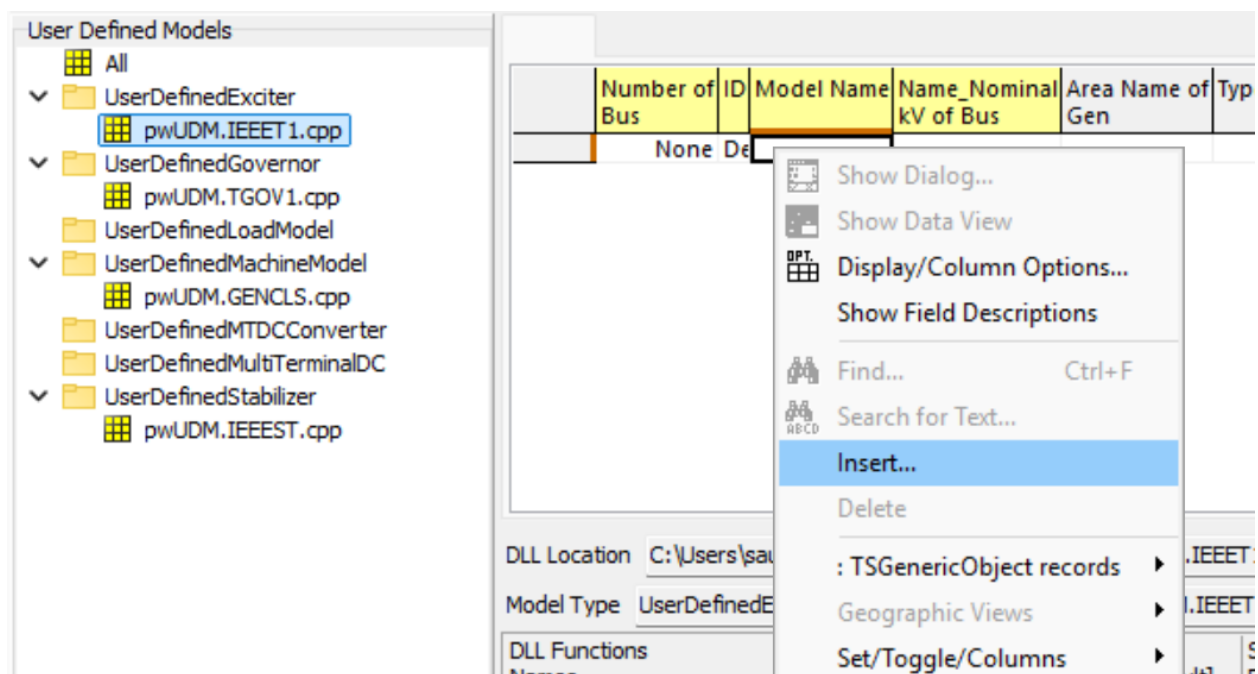
9. Simulator scans the folder in the browsing path(s), and automatically loads and validates the DLLs. They are also categorized based on their model types. In this case, four DLLs are shown to have been loaded.



10. Click on pwUDM.IEET1.cpp, and the following description should appear in the lower pane of this tab. This shows the DLL Functions' Names, Extra Objects Types, Default Parameter Values, States Description, Hard-Coded Signals Variable and Algebraic Variable Description. There should be no error message shown, and the dll should be ready for use. At this point, the user defined model is recognized by Simulator can be handled just like any built-in model.

DLL Location C:\Users\saurav\Desktop\BasicUser\CppDLLs\pwUDM.IEEET1.cpp.dll					
Model Type UserDefinedExciter		Model Name pwUDM.IEEET1.cpp			
DLL Functions	Extra Objects	Parameters	States	Hard-Coded Signals	Algebraics
Names	Type (Description)	Name [Default]	Description	Variable [Description]	Variable [Description]
Implemented Functions		0: Tr [0]	0: EField	0: Vref (Voltage Reference)	
modelClassName		1: Ka [50]	1: Sensed Vt	1: Initial Efield	
DLLVersion		2: Ta [0.04]	2: VR	2: Field Current	
parameterName		3: Vrmax [1]	3: VF	3: Voltage Compensated	
stateName		4: Vrmin [-1]		4: Generator Speed Deviation (pu)	
allParamCounts		5: Ke [0.06]		5: Voltage Generator Terminal	
getDefaultParameterValue		6: Te [0.06]		6: Stabilizer Input (Vs)	
SubIntervalPower2Exponent		7: Kf [0.09]		7: Flag for active OEL (1 if active)	
initializeYourself		8: Tf [1.46]		8: Over Excitation Limiter Signal	
getNonWindUpLimits		9: Switch [0]		9: Flag for active UEL (1 if active)	
calculateFofx		10: E1 [2.8]		10: Under Excitation Limiter Signal	
PropagateIgnoredStateAndInput		11: SE(E1) [0.04]			
ExciterEfieldOut		12: E2 [3.73]			
Optional Functions Omitted		13: SE(E2) [0.33]			
OtherObjectClass		14: SpdmIt [0]			
OtherObjectDescription					
signalSelection					
getStringParamDefaultValue					
TimeStepEnd					
TimeStepEndAction					

- Ensure you have clicked on the pwUDM.IEEET1.cpp, and then right click anywhere in the top-right pane, or on the word "None". Then, select Insert.



- The Generator Information dialog box should be shown. Here, the built-in IEEET1 exciter model at Bus 1 will be replaced by an pwUDM.IEEET1.cpp user defined exciter model. Click Insert.

Generator Information for Current Case

Bus Number: 1 Find By Number

Bus Name: Bus1 Find By Name

ID: 1 Find ...

Area Name: 1 (1)

Labels ... no labels

Generator MVA Base: 500.00

Status: ☐ Open ☒ Closed

Energized: ☐ NO (Offline) ☒ YES (Online)

Fuel Type: Unknown

Unit Type: UN (Unknown)

Power and Voltage Control Costs OPF Faults Owners, Area, etc. Custom Stability

Machine Models Exciters Governors Stabilizers Other Models Step-up Transformer Terminal and State

Insert Delete Gen MVA Base: 500.0 Show Block Diagram Create VCurve

Type: Active - IEEE1 ☒ Active (only one may be active) Set to Defaults

Parameters

PU values shown/entered using device base of 500.0 MVA

Tr	0.0000	Te	0.6000	E2	3.7300
Ka	50.0000	Kf	0.0900	SE(E2)	0.3300
Ta	0.0400	Tf	1.4600	Spdmlt	0.0000
Vrmax	1.0000	Switch	0.0000		
Vrmin	-1.0000	E1	2.8000		
Ke	-0.0600	SE(E1)	0.0400		

OK Save Cancel Help Print

13. Here, select pwUDM.IEEET1.cpp. Click OK.

Model Type

Select Model from the Below List

- IEEX2A
- IVOEX
- MEXS
- PLAYINEX
- pwUDM.IEEET1.cpp**
- REXS [Translate to REXSYS in PTI]
- REXS1
- REXSYS [Translate to REXS in GE]
- SCRX
- SEXS_GE [SEXS in GE]
- SEXS_PT1 [Translate to SEXS in GE] [S]
- ST10C
- ST1C [ESST1C in GE]

Show Models Supported By

☒ PW Only ☒ PTI

☐ BPA ☒ GE

☐ Both GE and PTI

☐ Only Show Used Models

☒ Limit Models by Machine Model

OK Cancel

14. Now the pwUDM.IEEET1.cpp user defined exciter model has been inserted at Bus 1, and is ready for use. Ensure the Active checkbox is selected. The default parameter values have been loaded from the DLL, which can be altered if needed. For now, leave the values as the defaults, click Save, and then OK.

Machine Models | **Exciters** | Governors | Stabilizers | Other Models | Terminal and State | Results | Step-up Trans

Insert | Delete | Gen MVA Base 500.0 | Show Block Diagram | Create VCurve

Type: Active - pwUDM.IEEET1.cpp | ☒ Active (only one may be active) | Set to Defaults

Parameters

Tr	0.00000	Ke	0.06000	E1	2.80000
Ka	50.00000	Te	0.06000	SE(E1)	0.04000
Ta	0.04000	Kf	0.09000	E2	3.73000
Vrmax	1.00000	Tf	1.46000	SE(E2)	0.33000
Vrmin	-1.00000	Switch	0.00000	Spdmlt	0.00000

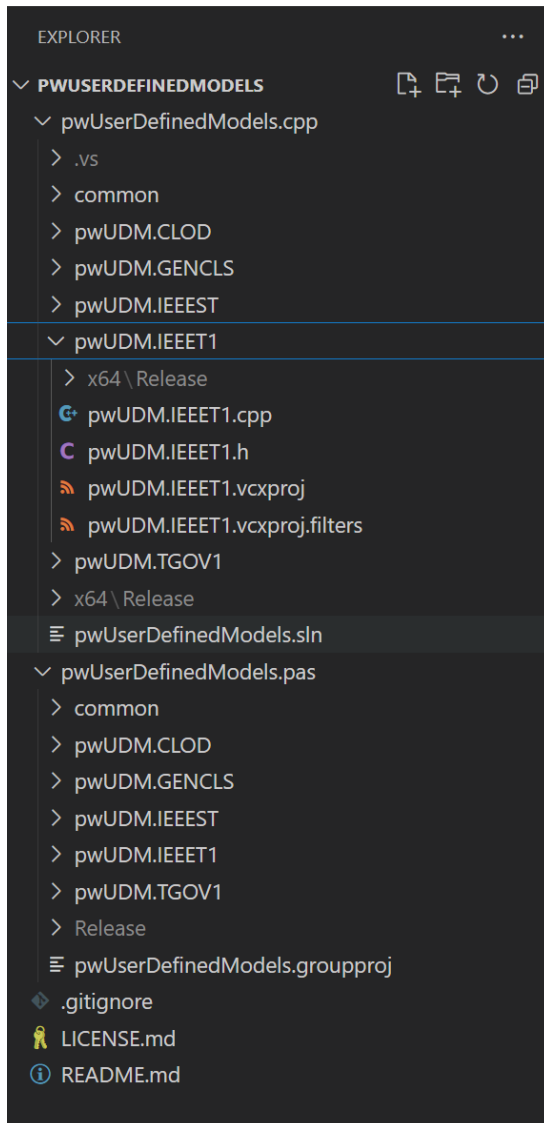
15. The model is now ready for use. This test case already contains a Transient Contingency named UDM_COMBINED_DEMO, which contains a fault at Bus 7. To execute the simulation, click on the Run Transient Stability button.

3. Advanced Developers

The recommended method to develop custom models using the user defined model specification is by forking the public GitHub repo mentioned below. This repo will be maintained as customers have more requests and will provide ready-to-use functions and helper files to make the process streamlined.

<https://github.com/PowerWorld-Corporation/pwUserDefinedModels>

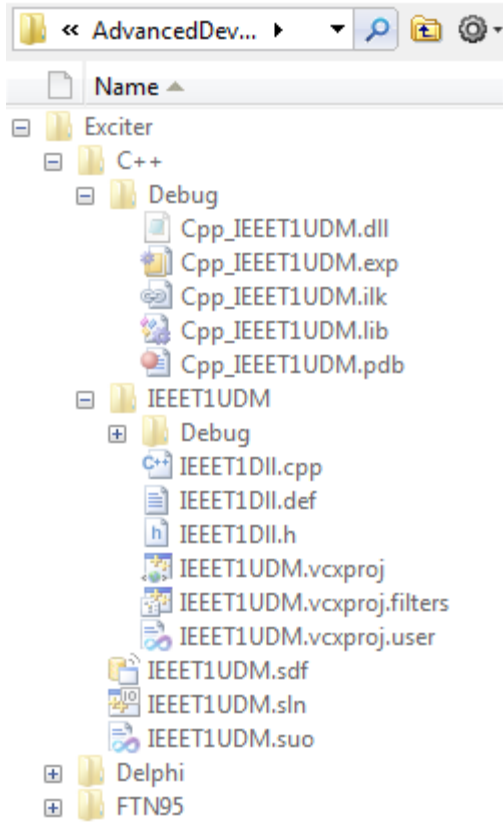
The screenshot below gives you an idea of the project structure.



For debugging, the pwrworld.exe process can be attached to the debugger being used. This can also be done by specifying the location of the pwrworld.exe file (from the installation directory of PowerWorld Simulator, version 17 or later) as the launch target during the debugging process.

Note: While re-building DLLs during the debugging process, ensure that the attached PowerWorld Simulator is closed

Corresponding to the DLLs from the BasicUser folder, the AdvancedDeveloper folder contains various subfolders containing the source code for generating the respective DLLs. The example project structure of is shown below. **Note: This was first developed in 32-bit, and all projects will need to be retargeted to 64-bit if users choose to use the older example files. in the AdvancedDeveloper folder.**



For developers planning to create DLLs for new models, debugging will need to be done from the IDE in which the DLL is being built. Similar to instructions in Section 2, developers can point directly to a subfolder within a DLL project folder. This will eliminate the need to copy-paste the DLL file to a different location, and will help with a smoother debugging experience.

Examples have been discontinued in Fortran. If Fortran is firm requirement for your custom development, please contact PowerWorld.