Quick Start for User Defined Models

Last Updated: June 11, 2013

PowerWorld Corporation

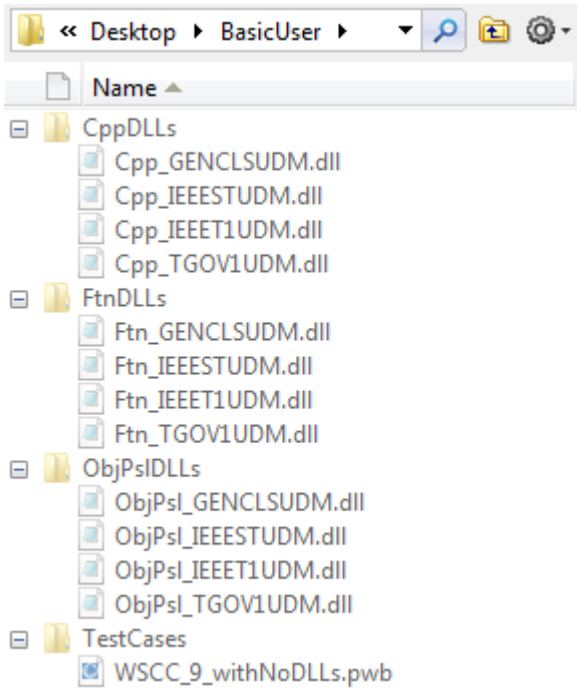
PowerWorld Corporation
2001 South First St
Champaign, IL 61820
(217) 384-6330
http://www.powerworld.com
info@powerworld.com

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. 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 0 is geared towards basic users, and section 3 will provide a brief note for advanced developers.

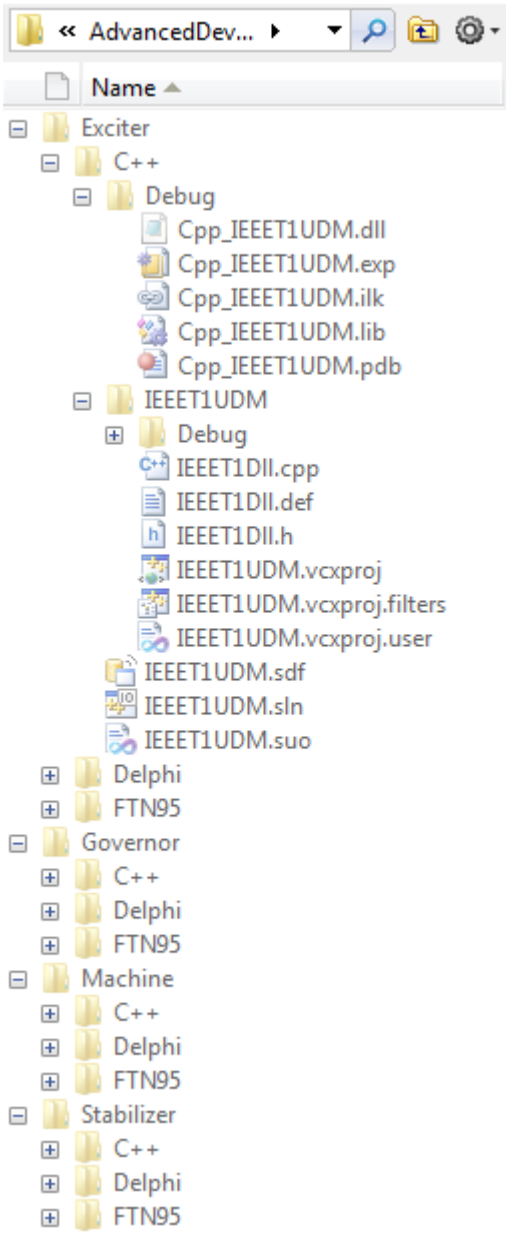
1. Before you begin…

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



The BasicUser folder contains four folders. The folders CppDLLs, FtnDLLs and ObjPslDLLs have identical DLLs of the GENCLS machine, IEEEST stabilizer, IEEET1 exciter and TGOV1 governor. In addition, the TestCases folder contains a case for illustrating the use of these DLLs.

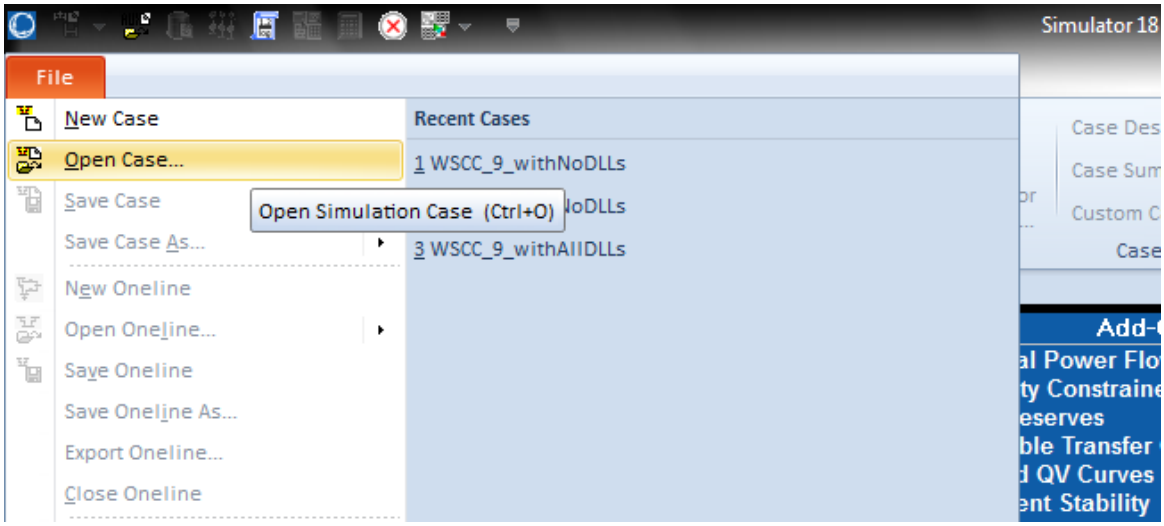
Corresponding to the DLLs from the BasicUser folder, the AdvancedDeveloper folder contains various subfolders containing the source code for generating the respective DLLs. The project structure of Cpp_IEEET1UDM is shown on the right-side of this page.



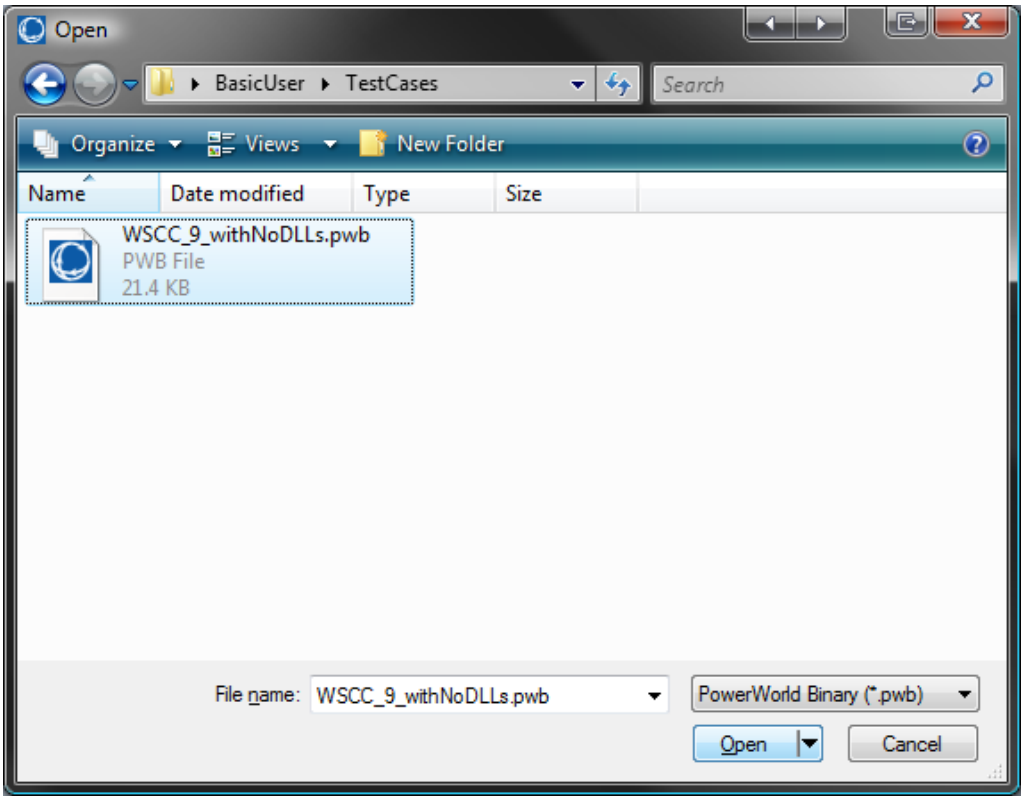
2. Basic Users

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.

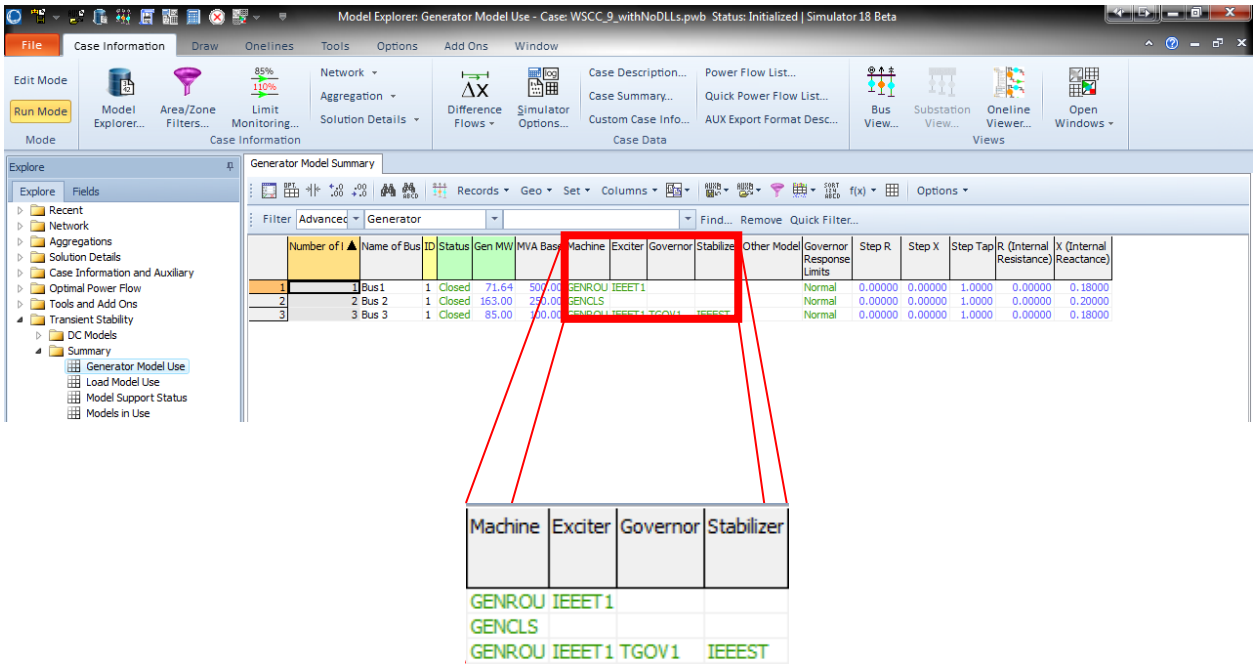
1. Start PowerWorld Simulator, and navigate to File > Open Case…



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

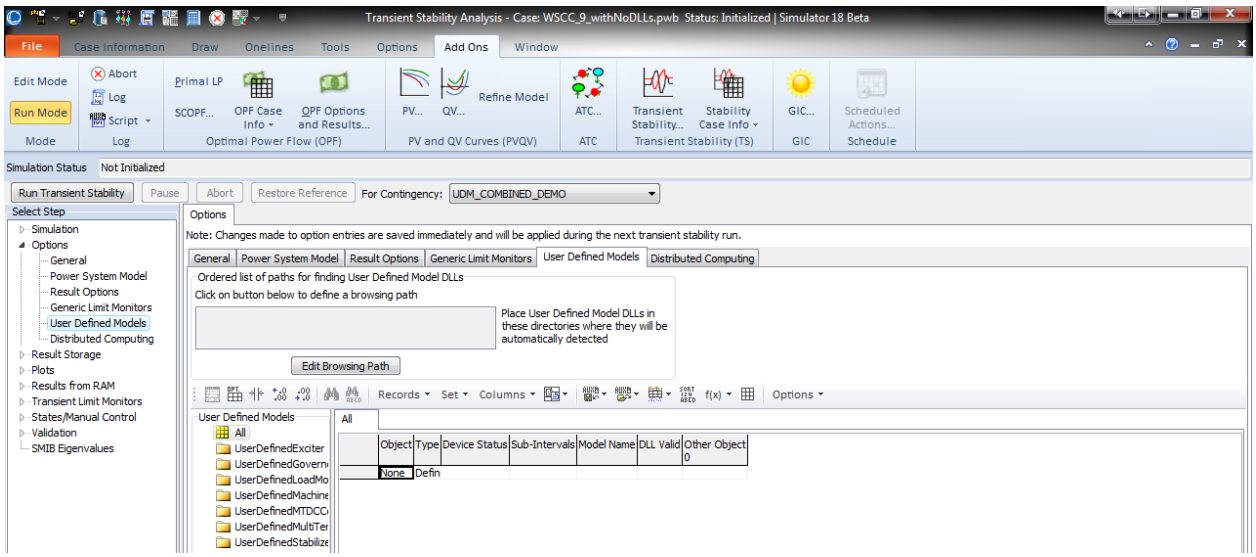


3. Once the case loads, navigate to the Case Information tab. In Model Explorer, navigate to the Transient Stability > Summary > Generator model use option in the left-side menu.



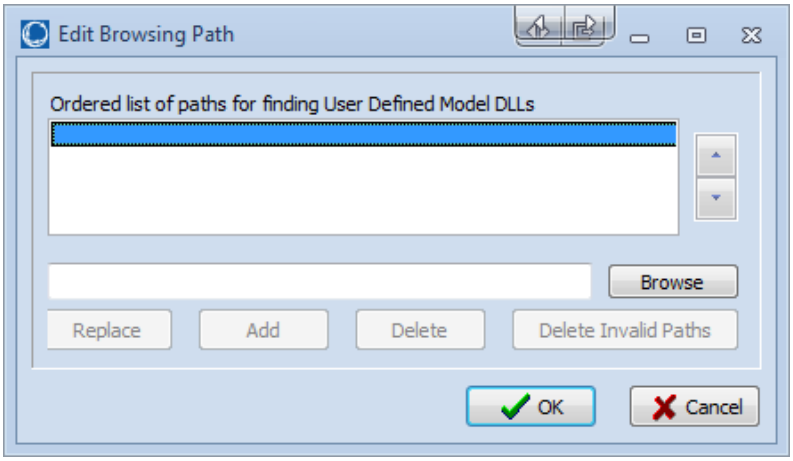
In the Generator Model Summary tab the machine, exciter, governor and stabilizer models in use are displayed. These models are built-in models supported by PowerWorld Simulator. In the subsequent steps, user defined models will be inserted to replace these models.

4. 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.

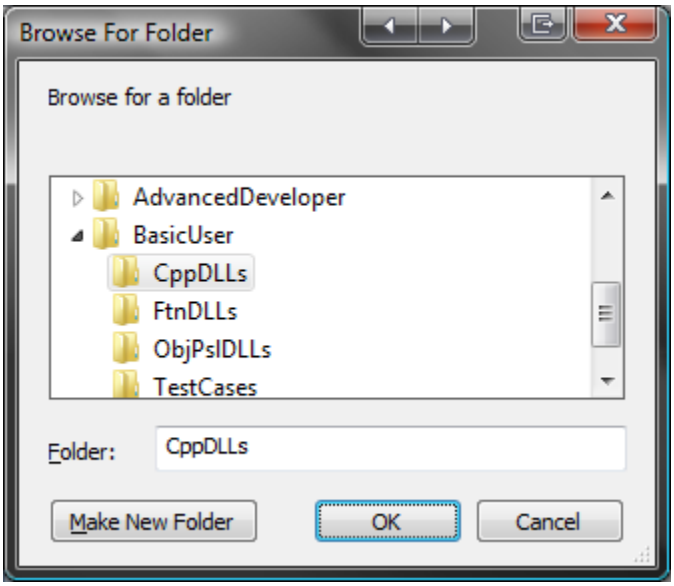


5. Click on the Edit Browsing Path button.

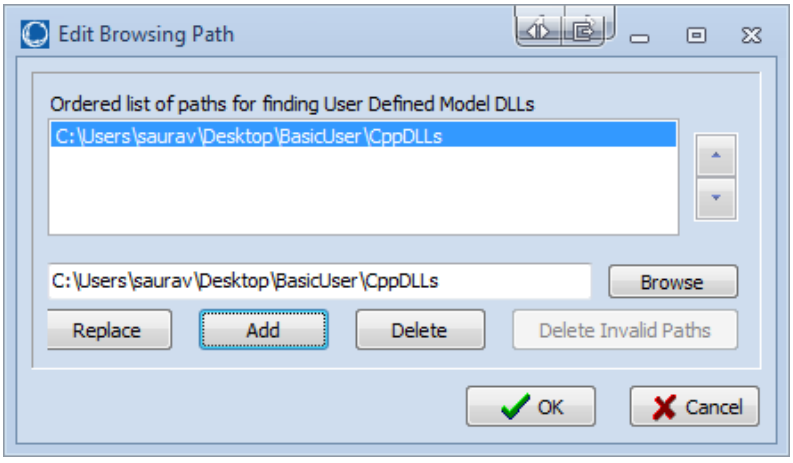
6. In the window below, click Browse.



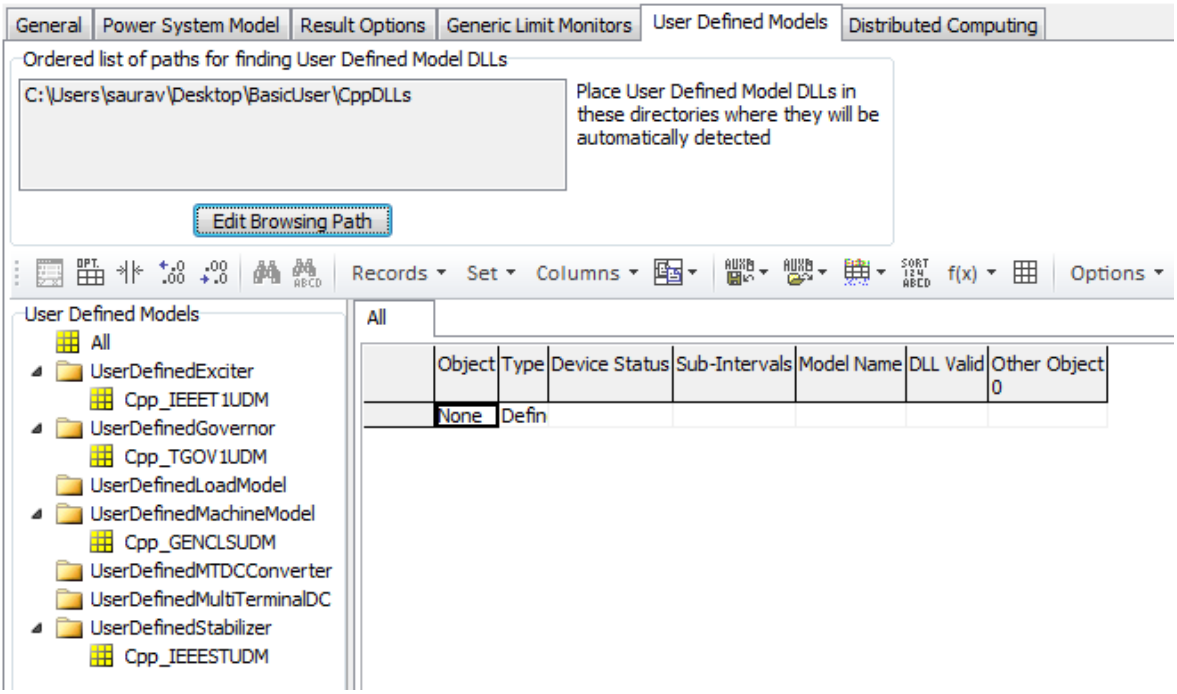
7. Then, browse for the BasicUser > CppDLLs folder. Click OK.



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.



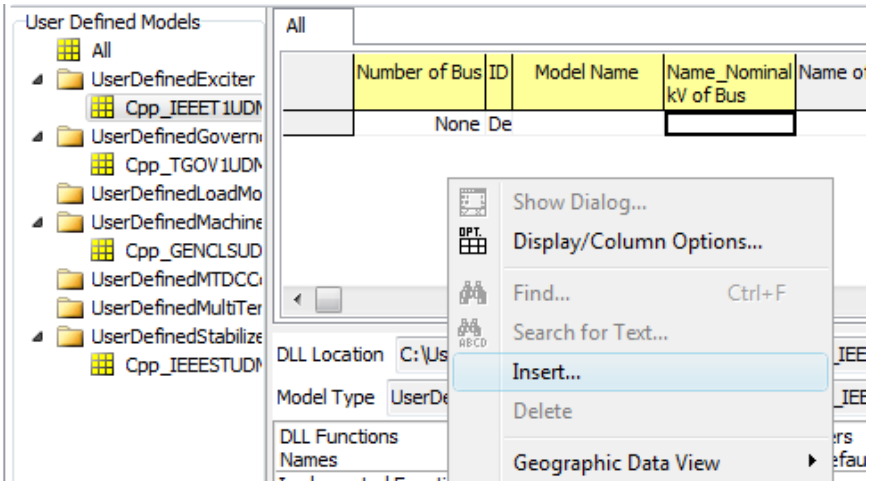
10. Click on Cpp_IEEET1DM, and the following description should appear in the lower pane of this tab. This shows the DLL Functions' Names, Extra Objects Types, Default Parameter Values, States Description, Hard-Coded Signals Variable and Algebraic Variable Description. There should be no error message 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\Cpp_IEEET1UDM.dll

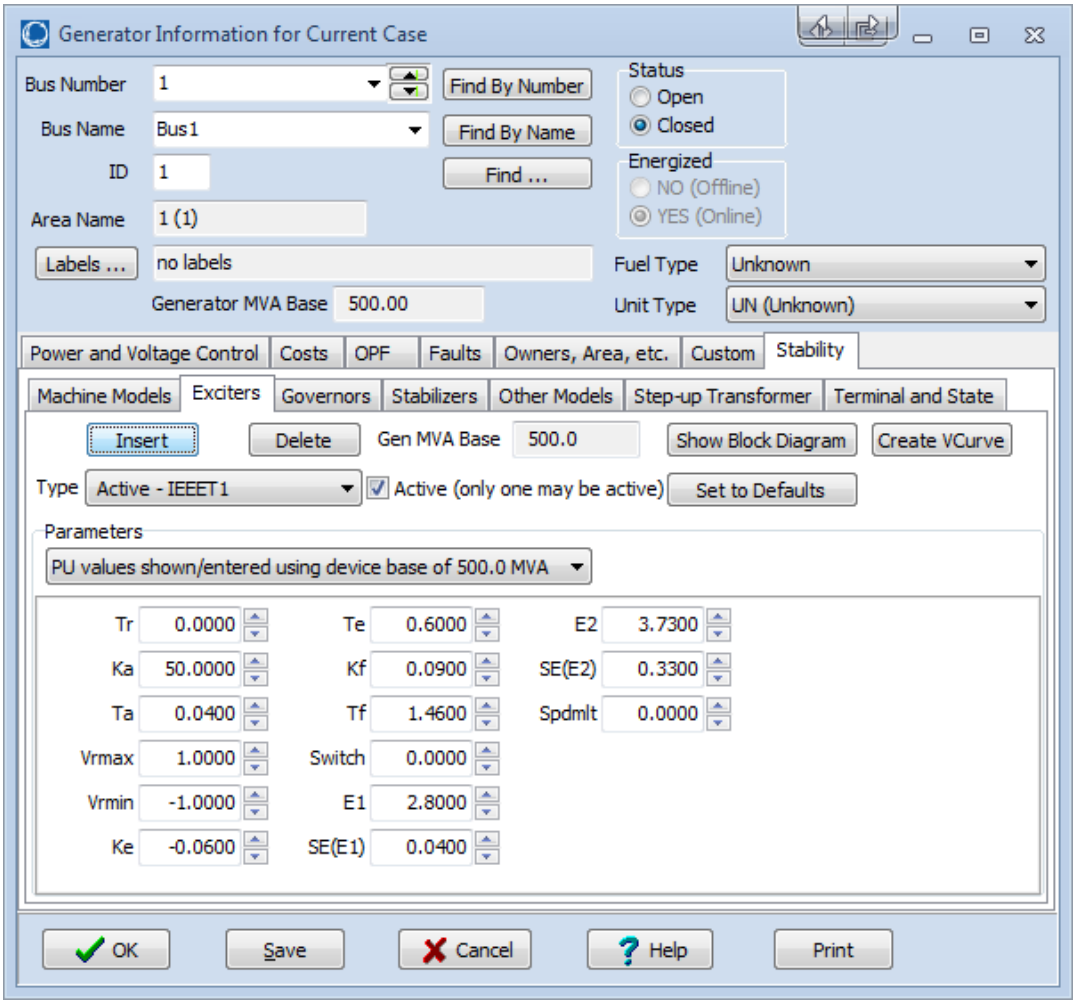
Model Type: UserDefinedExciter Model Name: Cpp_IEEET1UDM

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

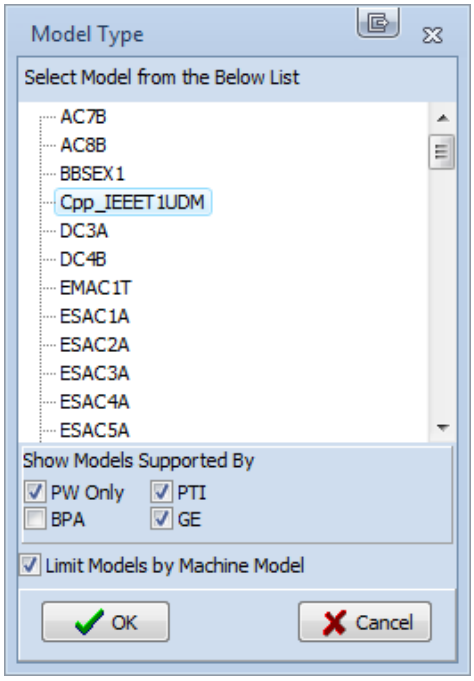
11. Ensure you have clicked on the Cpp_IEEET1DM, and then right click anywhere in the top-right pane, or on the word “None”. Then, select Insert.



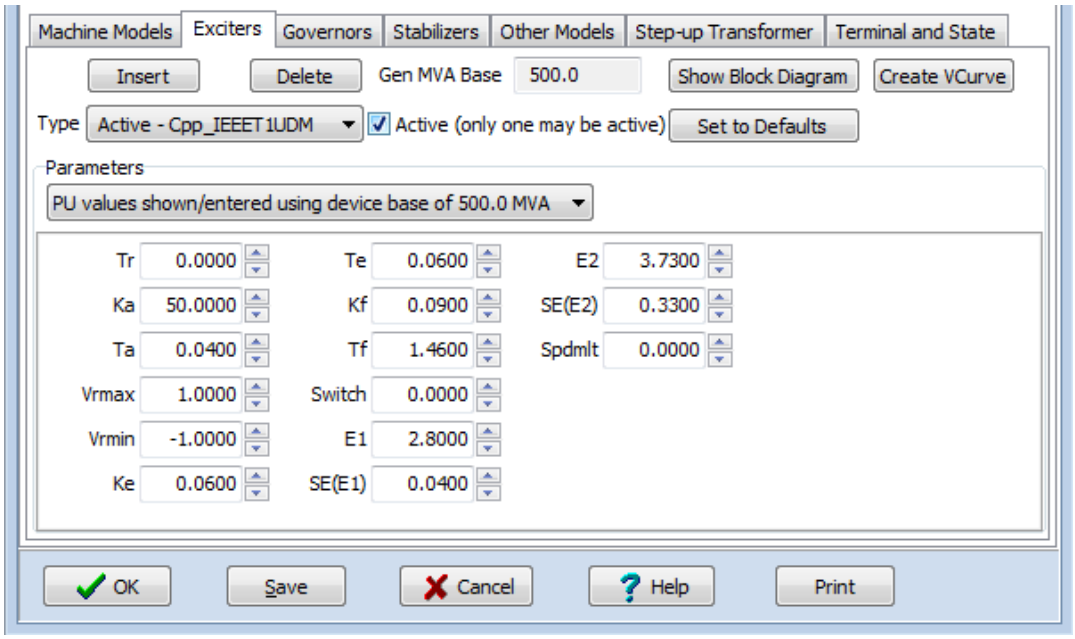
12. The Generator Information dialog box should be shown. Here, the built-in IEEET1 exciter model at Bus 1 will be replaced by an Cpp_IEEET1UDM user defined exciter model. Click Insert.



13. Here, select Cpp_IEEET1UDM. Click OK.



14. Now the Cpp_IEEET1UDM user defined exciter model has been inserted at Bus 1, and is ready for use. 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.



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

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 sub-folder 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.

The following are the folders of interest in the following programming languages (IDEs)

1. Object Pascal (Embarcadero® Delphi® XE Version 15.0)

   ~Debug\Win32

2. Visual C++ (Microsoft® Visual Studio 2010)

   ~\Debug

3. Fortran (Microsoft® Visual Studio 2010 with Silverfrost FTN95 plug-in)

   ~Debug\Win32

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