**Adding a New Module**

Last Updated 8/16/2019 by JWM (Rule tables section only)

Further review/updates needed

*This article assumes the reader has significant experience modifying existing siumulation modules.*

**Create Microsoft Visual C++ files:**

Create a new solution in the Microsoft Development Environment:

* File->New->Blank Solution…: Enter name (same as the simulation module being added -- e.g. “MySimulation”) and modify location to D:\TrimSim. The “Solution will be created at…” line should read D:\TrimSim\MySimulation

Create a new project in the solution:

* Right-click on the solution and select Add->New Project…
	+ Visual C++ Project
		- Win32 project
* Enter name (same as the module) & modify location (you will probably need to take the “MySimulation” directory specification out of the default location). The “Project will be created at…” line should read D:\TrimSim\MySimulation, not D:\TrimSim\MySimulation\MySimulation.
* Click “OK”
* On the subsequent popup window (Win32 Application Wizard), click “Application Settings”:
	+ Check DLL and Empty Project: Selecting an empty project is important, especially if you already have files in this location where the project is to be created. The Wizard will create <MySimulation>.cpp and .h here if empty project is not selected and wipe out any files (without warning) that exist here with the same name.
	+ Click “Finish”

Add initial version of source code:

* Create the files MySimulation.cpp and MySimulation.h in the D:\TrimSim\MySimulation directory. These can be blank, or copies of existing files.
* Right click on “Source Files->Add->Add Exisitng Item” and select the newly-created \*.cpp file.

Modify project properties:

* Right click on the project name and select “Properties “:
	+ General: Use MFC in shared DLL for both the debug and release configurations.
	+ Debuging->Command: “D:\TrimExe\debug\Trim.exe” for the debug configuration
	+ Debugging->Working Directory: “D:\TrimExe\debug\” for the debug configuration
	+ C/C++->Preprocessor->Preprocessor Definitions: add \_AFXEXT for both the debug and release configurations.
	+ Linker->General->Output File: “D:\TrimExe\Debug\MySimulation\_1\_0” for the debug configuration (this needs to be changed if you want the debug version to have some other version number besides 1\_0).
	+ Linker->Input->Additional Dependencies: “D:TrimEXE\TrimDLL\Debug\TrimDLL.lib” for the debug configuration and “D:\TrimEXE\TrimDLL\Release\TrimDLL.lib” for the release configuration

Copy the D:\TrimSim\MySimulation directory (and its contents) to the Source\TRIMSim folder of the ASPE-TRIM server.

**Add the new simulation to the Release tool:**

Edit your copy of “Release.ini” (usually in the WINNT folder). Add MySimulation to the initial list of simulations (add it in alphbetical order, and renumber all simulations after it). After the initial list, add the following lines (again, in alphabetical order):

[MySimulation]

Directory=D:/TrimSim/MySimulation

Version=1\_0

Save Release.ini, and run the release tool to release version 1\_0 of MySimulation with “Developers only” status. Your initial code is now safely stored in the configuration management files.

Make the same change to the version of “Release.ini” in the Source\Release folder of the ASPE-TRIM server.

# Create CTD rule tables for the simulation:

Create four new CTD tables:

MySimulationNational

MySimulationArray

MySimulationState

MySimulationStateArray

as empty (i.e. structure only, no data) copies of:

ShellNational

ShellArray

ShellState

ShellStateArray

The simplest way to do this is to issue a query similar to the following for each of the four (National, State, Array and StateArray) rules tables that need to be created:

“Create ctd.[NewModuleName]National like ctd.ShellNational”.

**CAUTION: “State” and “Sequence” in the state, array and StateArray tables must be defined as “float” data types. “Integer” data types would also be acceptable but would not be consistent with the way other rules tables are defined. Defining these columns as character data types (e.g., “varchar”) will result in errors in the way the data are shown in the interface and read into TRIM3 simulations and errors in saving edits to state-level rules.**

After creating the tables, add any additional rules needed by the new simulation.

**Add data to the CTD for an initial run of the new simulation:**

Add MySimulation to the CTD.Simulations table (this simply registers the new simulation):

insert into ctd.simulations (simulation) values(‘MySimulation’)

Add a ctd.runs record for an initial run by using the interface to create a blank run. Then manually create a corresponding record in the ctd.rundetails table as follows (make sure that runid and simulationid correspond with the blank runid):

insert into ctd.rundetails

(runid,simulation,simulationid,module)

values('InitialMySimulation’,'MySimulation','InitialMySimulation\_',

'MySimulation\_1\_0')

Manually populate the four new CTD rule tables with values for initial SimulationID defined above. For national rules, only 1 record is required. For state rules, 56 records are required (one for each of the 56 fips codes). The national array table requires 1 record per sequence for each simulation setup, and the state array table requires 1 record per sequence for each state, so that the total number of records for that table for each simulation setup = number of states \* number of sequences per state. Also add any necessary records to the ctd.Variablelist table for the SimulationID.

NOTE: The maximum sequence for the national array and state array table should correspond either to the maximum sequence defined in the header file of the new module or with the default values defined by the TRIM3 frame, which is 60. (See MAX\_BRACKET\_SIZE in the stdafx.h file.) If a module defines a maximum bracket size less than that—it cannot be greater—and more array elements are needed at a future time, then those array elements may be added (up to the maximum of 60 defined in the frame code) for every existing setup using a Perl script available to TRIM3 administrators. (**TRIM3 Navigator -> Internal Scripts and Procedures -> Scripts -> Expand array tables (increase the sequence number) and add any missing sequences**)

# Summary Tables

## Add a subdirectory named **MySimulation** under wwwroot/SummaryTables, and add a \*.php file for each table produced by the simulation.

# Result Variables

If not already done, add the VariableList rules “AnnualOutput” and (if appropriate) “MonthlyOutput”.

**Documentation**

## Add a subdirectory named MySimulation under wwwroot/Documentation, and create a file named “Main.html” to hold the documentaion. Use the file “wwwroot/documentation/shell/main.html” as a template.

**Misc**

Add at least three records to the CTD.Terms table for the new module for the following Term, SourceTable, and Category fields:
“AnnualOutput” “VariableList” “Output”
“MonthlyOutput” “VariableList” “Output”
“SimulationMode” “*NewModule*National” “Other”