ChangeParameters_and_Rerun

February 17, 2016

1 Example of running Source with changed parameters

We're going to change the impervious fraction parameter in Sacramento and run Source with the original and changed values

In [1]:
```
# Some steps required until these Python modules are properly installed...
import sys
sys.path.append('../Modules')
sys.path.append('../../../../veneer-py')
# Get the Source scripting components (veneer) and GBR specific stuff
import gbr
```

In [2]:
```
# Point the system at a particular output directory...
gbr.init('D:/Beckers/outputs/Scenario 1/')
```

In [3]:
```
# Initialise the Veneer (Source scripting tool)
v = gbr.veneer()
```

In [4]:
```
# A path to the variable of interest... .* means 'give me all'
accessor = 'scenario.Network.GetCatchments().*FunctionalUnits.*rainfallRunoffModel.theBaseRRMod
```

In [5]:
```
existingValues = v.model.get('scenario.Network.GetCatchments().*FunctionalUnits.*rainfallRunoff
existingValues
```

Out[5]:
```
[0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
 0.01,
```

```
         0.01,
         0.01,
         0.01,
         0.01,
         0.01]
```

In [6]:
```
# Run with those original values

# First, set the name of the run
v.model.set('scenario.CurrentConfiguration.runName','RUN_ORIGINAL_PCTIM')
```

In [7]:
```
# Its a good idea to set some options in Dynamic Sednet to prevent the results window appearing
# Also, to make it automatically override existing results
v.configureOptions({'ShowResultsAfterRun':False,'OverwriteResults':True})
```

Out[7]:
```
{'Exception': None, 'Response': None, 'StandardError': '', 'StandardOut': ''}
```

In [9]:
```
# Also, lets switch on the performance options
v.configureOptions({'RunNetworksInParallel':True,'PreRunCatchments':True,'ParallelFlowPhase':Tru
v.model.sourceScenarioOptions("PerformanceConfiguration","ProcessCatchmentsInParallel",True)
```

Out[9]:
```
{'Exception': None,
 'Response': {'Value': True,
  '__type': 'BooleanResponse:#FlowMatters.Source.Veneer.ExchangeObjects'},
 'StandardError': '',
 'StandardOut': ''}
```

In [10]:
```
# Now, lets run the model... When this cell executes in Python, the run window should appear i
v.run_model()
```

Out[10]:
```
(302, 'runs/1')
```

In [11]:
```
# NOTE: The above output (eg runs/1) is a point to retrieving the 'normal' Source results - ie
# We don't need that for GBR/Dynamic Sednet, because we can get to the summarised results
```

In [12]:
```
# Lets take a quick look at those results...
results_original = gbr.Results('RUN_ORIGINAL_PCTIM')
results_original.queries.regional_export('t/y')
```

In [13]:

Out[13]:

SummaryRegion	agbot	agmid	agtop
Constituent			
Ametryn	0.000000e+00	0.000000e+00	0.000000e+00
Atrazine	0.000000e+00	0.000000e+00	0.000000e+00
Flow	5.618641e+08	4.826120e+08	2.044276e+08
N_DIN	1.076096e+05	9.206070e+04	3.776663e+04
N_DON	1.076096e+05	9.206070e+04	3.776663e+04
N_Particulate	5.590488e+05	3.949711e+05	2.687237e+05
P_DOP	9.987315e+04	8.674313e+04	3.633789e+04
P_FRP	9.987315e+04	8.674313e+04	3.633789e+04
P_Particulate	NaN	2.748961e+05	1.606461e+05
Sediment - Coarse	0.000000e+00	0.000000e+00	0.000000e+00
Sediment - Fine	4.943001e+06	3.917739e+06	2.476492e+06
Tebuthiuron	0.000000e+00	0.000000e+00	0.000000e+00

In [14]:
```
# NOW... Lets change the Pctim parameter...
# We'll use the same accessor -- but this time we'll use it to set values
accessor
```

Out[14]:
```
'scenario.Network.GetCatchments().*FunctionalUnits.*rainfallRunoffModel.theBaseRRModel.Pctim'
```

In [16]:
```
# We can set every 'instance' of Pctim - ie every FU in every subcatchment - to a single value
#
# v.model.set(accessor,0.5)
#
# or we can pass in a list of values
#
# v.model.set(accessor,[0.2,0.3,0.5,0.4,1.0],fromList=True)
#
# Now... If your list of values is shorter than the number of instances... (ie # subcatchments
# then the list will be 'recycled'... That is, the list will be reused repeatedly until values
# instances...
#
# ie... Given that the Becker's model has 5 FUs, [0.2,0.3,0.5,0.4,1.0] is the equivalent of gi
# five functional units in each subcatchment...
v.model.set(accessor,[0.2,0.3,0.5,0.4,1.0],fromList=True)
```

In [19]:
```
# Lets check what happened...
v.model.get(accessor)
```

Out[19]:
```
[0.2,
 0.3,
 0.5,
 0.4,
 1,
 0.2,
 0.3,
 0.5,
 0.4,
 1,
 0.2,
 0.3,
 0.5,
 0.4,
 1,
 0.2,
 0.3,
 0.5,
 0.4,
 1,
 0.2,
 0.3,
 0.5,
 0.4,
 1]
```

In [20]:
```
v.model.set('scenario.CurrentConfiguration.runName','RUN_CHANGED_PCTIM')
```

In [21]:
```
v.run_model()
```

Out[21]:
```
(302, 'runs/2')
```

In [22]:
```
results_changed = gbr.Results('RUN_CHANGED_PCTIM')
results_changed.queries.regional_export('t/y')
```

Out[22]:

SummaryRegion	agbot	agmid	agtop
Constituent			
Ametryn	0.000000e+00	0.000000e+00	0.000000e+00
Atrazine	0.000000e+00	0.000000e+00	0.000000e+00
Flow	5.985996e+09	5.482349e+09	5.046765e+09
N_DIN	1.318299e+06	1.218539e+06	9.657002e+05
N_DON	1.318299e+06	1.218539e+06	9.657002e+05
N_Particulate	2.604056e+07	2.483552e+07	3.695302e+07
P_DOP	1.265883e+06	1.177535e+06	9.466321e+05
P_FRP	1.265883e+06	1.177535e+06	9.466321e+05
P_Particulate	1.114083e+11	1.114078e+11	1.370545e+11
Sediment - Coarse	0.000000e+00	0.000000e+00	0.000000e+00
Sediment - Fine	2.457229e+08	2.394227e+08	3.207062e+08
Tebuthiuron	0.000000e+00	0.000000e+00	0.000000e+00

In [24]:
```
# Now that we've done both runs, we probably want to put the parameter back to normal...
v.model.set(accessor,existingValues,fromList=True)
```

In [25]:
```
# Now... Lets run a results comparison...
differences = gbr.DifferenceResults('RUN_ORIGINAL_PCTIM','RUN_CHANGED_PCTIM')
differences.queries.regional_export('t/y')
```

Out[25]:

SummaryRegion	agbot	agmid	agtop
Constituent			
Ametryn	0.000000e+00	0.000000e+00	0.000000e+00
Atrazine	0.000000e+00	0.000000e+00	0.000000e+00
Flow	-5.424132e+09	-4.999737e+09	-4.842337e+09
N_DIN	-1.210689e+06	-1.126478e+06	-9.279335e+05
N_DON	-1.210689e+06	-1.126478e+06	-9.279335e+05
N_Particulate	-2.548151e+07	-2.444055e+07	-3.668429e+07
P_DOP	-1.166010e+06	-1.090792e+06	-9.102942e+05
P_FRP	-1.166010e+06	-1.090792e+06	-9.102942e+05
P_Particulate	NaN	-1.114075e+11	-1.370543e+11
Sediment - Coarse	0.000000e+00	0.000000e+00	0.000000e+00
Sediment - Fine	-2.407799e+08	-2.355049e+08	-3.182297e+08
Tebuthiuron	0.000000e+00	0.000000e+00	0.000000e+00

In [ ]:
```
# As might be expected, increasing the impervious fraction increases flow and consequently the
```

In [ ]: