Adding information to:

<http://www.apsim.info/Wiki/Developing-Models-for-Apsim.ashx>

The [Input]-tag can locate the variable in a module if that module has the same ouput tag, eg

In module A

[Input]

private double AddingNO3=0;

In module B

[Output]

[Units("g")]

double AddingNO3=0;

Events with simple types:

In module A:

DoubleType ting = new DoubleType();

[Event]

public event DoubleDelegate AddN03;

[EventHandler]

public void OnProcess()

{

//do stuff

ting.Value = 0.1;

AddN03.Invoke(ting);

}

In module B

[Output]

[Units("g")]

double AmountOfNO3 = 0;

public void OnAddN03(DoubleType added)

{

AmountOfNO3 += added.Value;

}

In Grossmargin.xl we have:

<ApsimToSim>

<component name="[GrossMargin.name]" executable="[dll]">

<executable>[dll]</executable>

<componentInterface>%apsim%\Model\dotnetcomponentinterface.dll</componentInterface>

<InitData>

[Model]

</InitData>

</component>

</ApsimToSim>

This should look like so the interface variables would be used by the modules

<ApsimToSim>

<component name="[slurrytankAfterRest.name]" executable="[dll]">

<executable>[dll]</executable>

<componentInterface>%apsim%\Model\dotnetcomponentinterface.dll</componentInterface>

<InitData>

[Model]

<NCost>[GrossMargin.NCost]</NCost>

<NApplicationCost>[GrossMargin.NApplicationCost]</NApplicationCost>

<MinimumProtein>[GrossMargin.MinimumProtein]</MinimumProtein>

<ProteinIncrement>[GrossMargin.ProteinIncrement]</ProteinIncrement>

</InitData>

</component>

</ApsimToSim>

If you would like to use c++ you can make it into an DLL. Make a new visual c++ project called win 32 console application. During the wizard you should select DLL and Empty projct. Then add a new class to the project.

The header could look like

#pragma once

#include <stdio.h>

extern "C"

{

\_\_declspec(dllexport)int mains(int b);

}

And the cpp could look like:

#include "wrapper.h"

int mains(int b)

{

int d=0;

d+=b;

return d;

}

Now jump back to the c# project. Add under the pre -compiler area:

using System.Runtime.InteropServices;

within you class you can now load the DLL via:

[DllImport(@"c:\FooWrapper.dll")]

You also need to write witch function that can be access like

private static extern int mains(int b);

Now you should be able to use it like:

int t =mains(3);