Testing Framework

What:

This is a framework to run multiple tests which can be used for verification purposes.

Why:

Doing a resolution study involves changing the resolution of the grid and running tests for each resolution configuration. The testing framework simplifies this by automating certain tasks such as creating *ups* files, uda directories etc.

HowTo:

The testing framework is a perl script that reads in a XML-based configuration file. The configuration file has all the parameters that are necessary to run the various tests.

The script can be found in the following path.

src/Packages/Uintah/StandAlone/inputs/ICE/Scripts/verificationScripts/tester1.pl

An example config file can also be found under

/home/csafe/ramanuja/builds/110206SCIRun/src/Packages/Uintah/StandAlone/inputs/ICE/Scripts/verificationScripts/res_study.xml

Running the tester script

$ perl tester1.pl res_study.xml

Explanation:

This section will explain how to create a config file and explains how the tester script works.

Config File:

All the XML tags need to be matched.

The Config file always has a <*start*> and a <*/start*> tag at the beginning and the end respectively.

<Test> </Test>

Each Test is enclosed between <*Test*> and <*/Test*>

<Meta> </Meta>

For each test the <*Meta*> and <*/Meta*> tags will define the testing variables, so each test MUST have the following parameters enclosed inside the <Meta> tag.

<Title> </Title>

This tag has the title of the test. This is very important, because the title name will be used in generating "*.ups*", "*.uda*", and if necessary "*.pbs*" files. So please avoid using spaces in the title name, always use underscore(_) for the delimiters.

<upsFile> </upsFile>

The name of the ups file needs to be specified in here. If the *ups* file is not in the same directory as the tester script please use the full path to the *ups* file.

This *ups* file will not be destroyed, it is only used as a base *ups* file from which is used to create multiple variants.

<pbsFile> </pbsFile>

If you have to run it in a batch queue, then specify the *pbs* file (defaults to current directory, if full path is not specified). Again this *pbs* file will be used as a read-only base file from which appropriate *pbs* files will be created using the title name.

<interactive> </interactive>

This tag must be used when the jobs are run interactively rather than via a batch queue. The command that will be used to run the experiment must be enclosed inside the tags.

For Example:

<interactive>./sus -ice </interactive>

Do not specify the ups file in the interactive tag. It will be automatically filled by the script.

Interactive tag is mutually exclusive to the pbsFile option. One test can't have both the options.

<content> </content>

This tag is the most important one. The part of the ups file that needs to be replaced should be specified inside the content tags. The way the script works is, it looks for the starting XML tag that appears after the <content> tag and searches for the exact tag in the base ups file provided. Then it looks for the corresponding closing tag in the ups file and replaces the whole section with the section given between the <content> tag.

It is imperative that the next line after the <content> tag be not a blank line or a complicated tag like <box label = "solid">. The tag must be a simple tag for example : <Level>

Also the content MUST have more than one line. A single line replacement is not possible.

Script Description:

The script tester1.pl will read this config file and it will generate multiple *ups* files (one for each test) in the same directory. It might also create multiple *pbs* files if you are using the batch queue option. It will automatically start the experiments for each specified test.

The interactive tests will be run in a serial order, meaning, the script will run one test at a time. This is to make sure not to overwhelm the machine with multiple simultaneous tests.

If multiple tests with *pbs* files are specified then all the tests will be added to the batch queue in a parallel fashion, meaning, the script will not wait for each test to complete before it starts the other.

Warnings:

It is not possible to specify comments in the config file.

All tags are case-sensitive, so please follow the capitalizations specified.