OpenStudio Package Installer Test Standards

This document describes the minimum set of tests that should be conducted to determine if an installation package for the OpenStudio software suite is acceptable for release.

# Windows XP/7

* We provide a self-extracting installer for MS Windows systems.
* Double-click the installer application to start the installation process.
* Walk through the standard NSIS installer screens.
* Install to the default location, C:\Program Files\<OpenStudio Version>
  + If the user does not have admin rights, the location is C:\Documents and Settings\<username>\My Documents\<OpenStudio Version>
* The following executables are installed in the <OpenStudio Version>\bin directory.
  + PolicyAnalysisTool
  + ResultsViewer
  + RunManager
  + SystemOutliner
* The following components are installed in the <OpenStudio Version>\Ruby directory
  + The OpenStudio SketchUp Plugin
  + The OpenStudio Ruby bindings
* The following component is installed in the <OpenStudio Version>\CSharp directory
  + The OpenStudio CSharp bindings
* The installer also attempts to locate any SketchUp 7 or 8 installation and installs the OpenStudio Plugin to SketchUp’s Plugins directory
  + If the SketchUp directory cannot be found the user will be informed at install time and the Plugin will not be copied to any location outside of the <OpenStudio Version>\Ruby directory.
* Shortcuts to PolicyAnalysisTool, ResultsViewer, RunManager, SystemOutliner, and the Uninstaller are added to the Start menu.
  + Click on each of the applications in the start menu to launch the individual applications.
* Install Graphviz/MiKTeX, ensure that it is in the system path before any Qt or OpenStudio paths, and then attempt to launch SketchUp and OpenStudio. If the plugin and app launches successfully, it is using the correct dlls.

# Mac OS X 10.5 or greater

* We provide a Disk Image file (.dmg) containing a PackageMaker package (.pkg) file for Mac OS X systems.
* The package installer places the following applications in the /Applications/<OpenStudio Version> directory
  + PolicyAnalysisTool.app
  + ResultsViewer.app
  + RunManager.app
  + SystemOutliner.app
* In addition, a /Applications/<OpenStudio Version>/Ruby/ directory is created.
  + The Ruby bindings are placed in this directory.
  + The OpenStudio SketchUp Plugin is located in this directory
  + A symbolic link to the Ruby bindings is placed in the /Library/Ruby/Site directory
* The OpenStudio Sketchup Plugin is installed to the /Users/<username>/Library/Application Support/Google/Google Sketchup 8/Plugins directory if it exists.

# Ubuntu 10.04

* Ubuntu 10.04 is required because of Qt and Boost version dependencies.
* We provide a Debian software package file (.deb)
  + In addition a self-extracting tarball (.sh) is available.
* The Debian installer will place the following applications in the /usr/bin directory
  + PolicyAnalysisTool
  + ResultsViewer
  + RunManager
  + SystemOutliner
* Additionally all of the shared libraries are installed at /usr/lib and the release notes are copied to /usr/.
* Example files and perturbation files are installed at /usr/share/openstudio/.
* The ruby bindings are installed to /usr/local/lib/site\_ruby/1.8/.

OpenStudio Sanity Tests

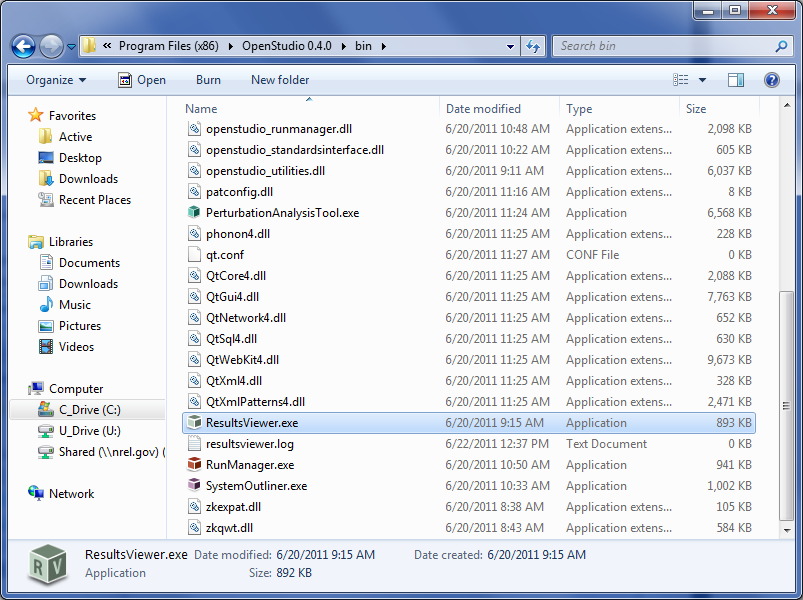
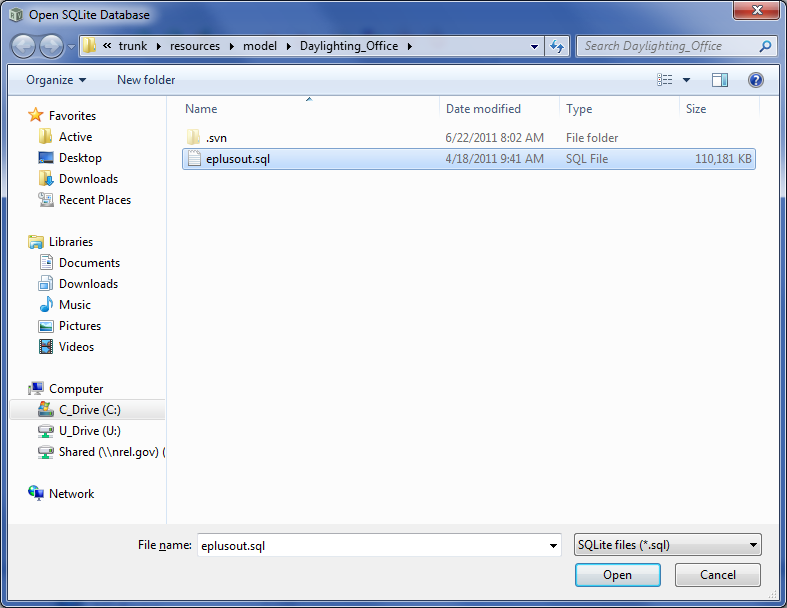
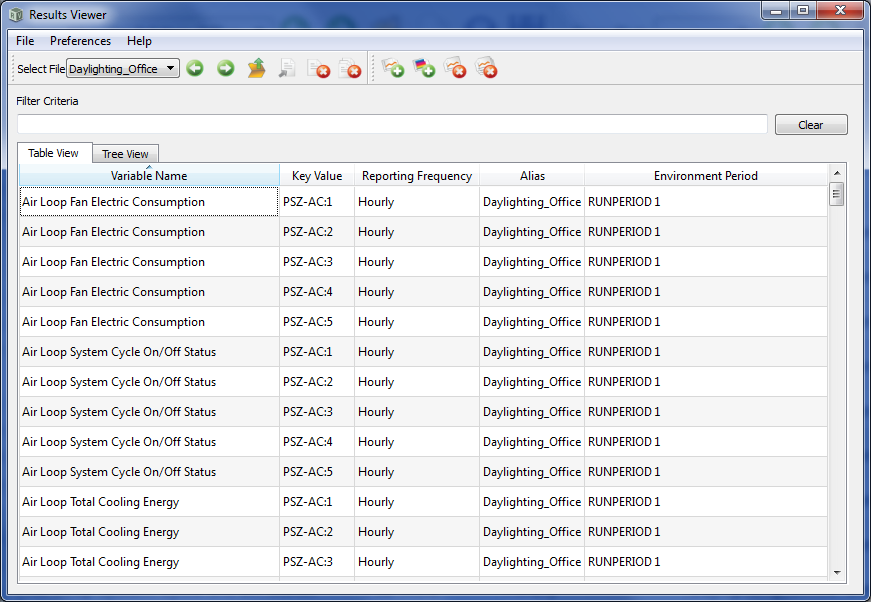
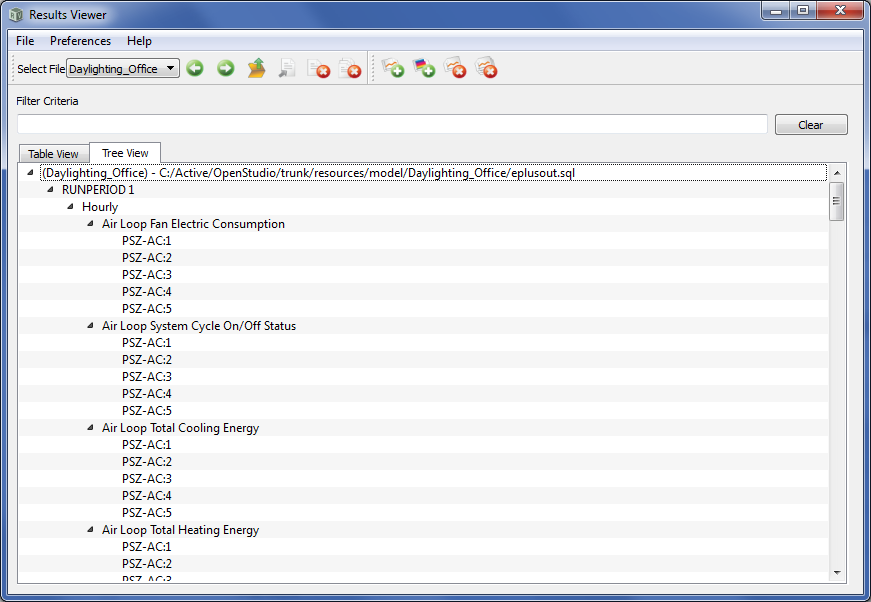
# Application Sanity Tests

The following tests are performed to prove that the applications have been installed correctly, can be launched, and run minimally without crashing.

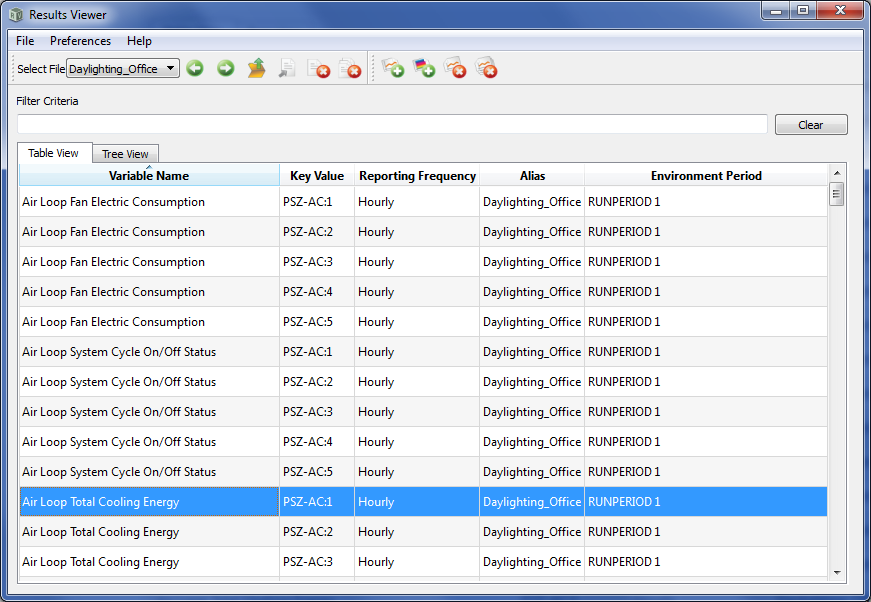
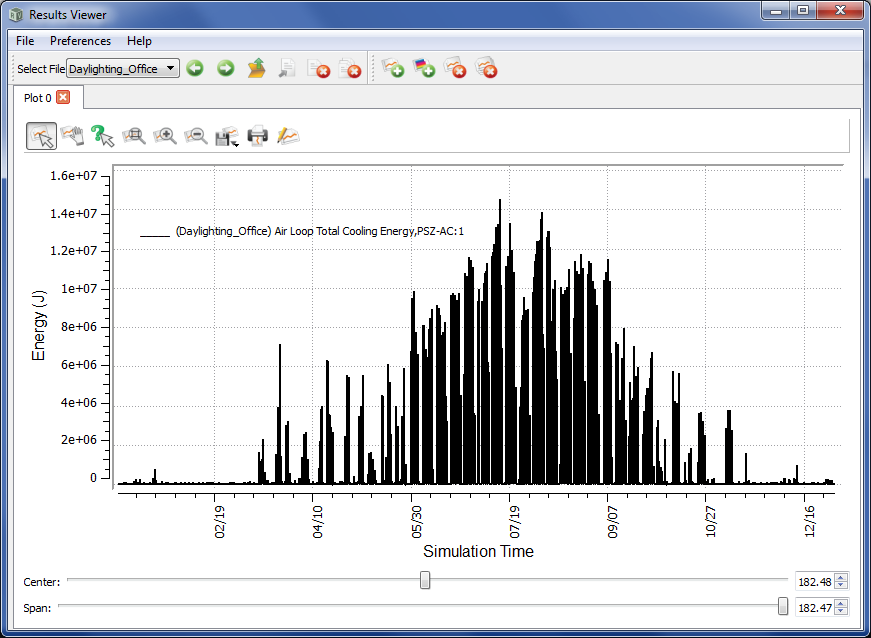
# OpenStudio ResultsViewer sanity tests

## RV Test 1

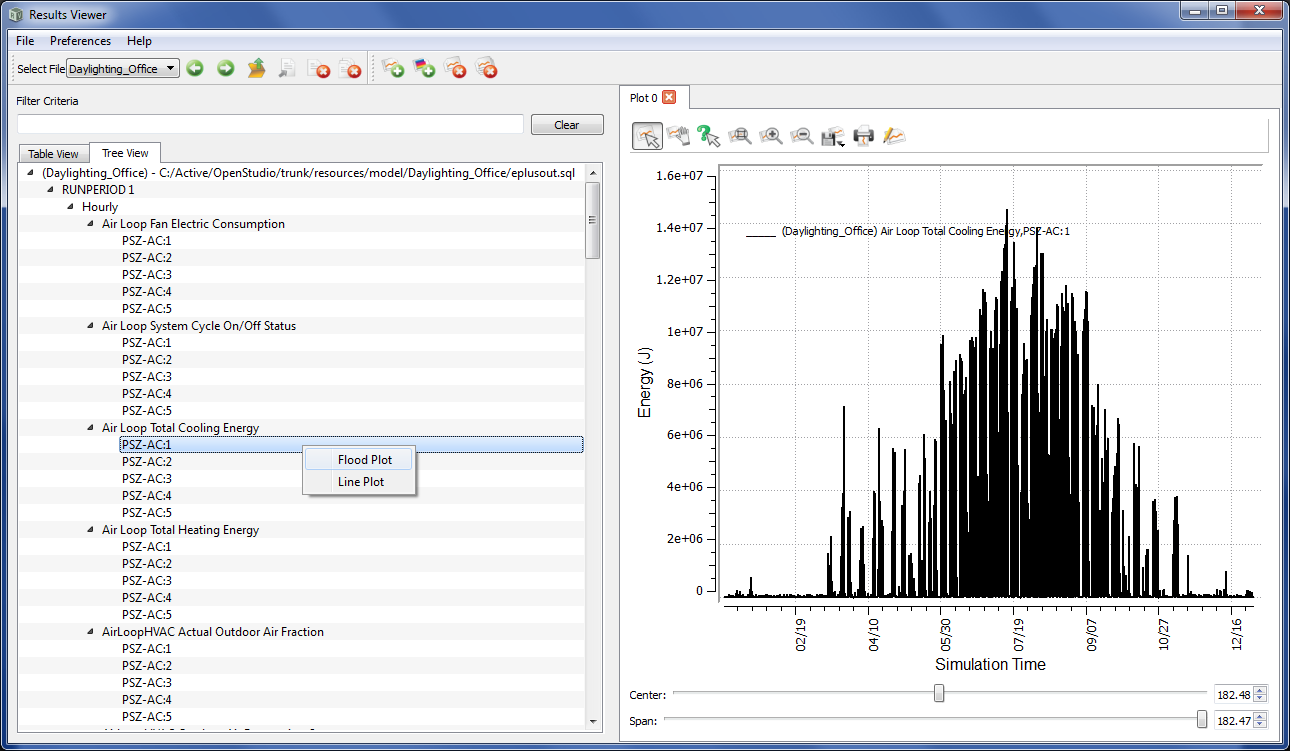
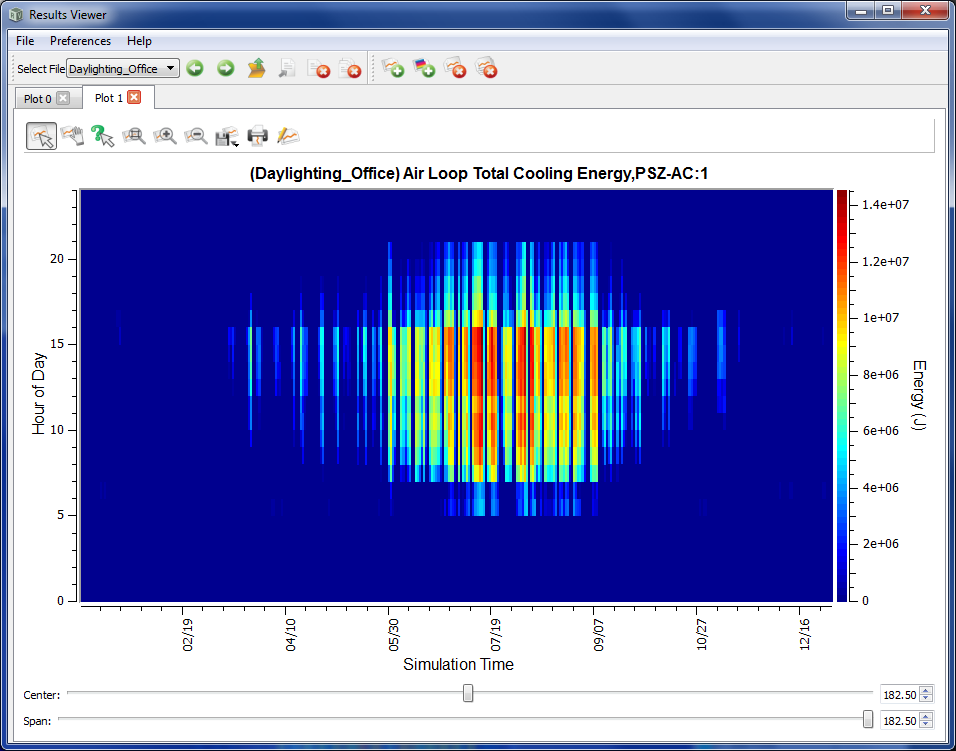
(Assumes that resources/model/Daylighting\_Office/eplusout.sql file exists. If not, generate by building PACKAGE and open the resources/model/Daylighting\_Office/in.osm in ModelEditor, export it as an in.idf, then run the simulation in RunManager)

1. Run ResultsViewer from explorer (double click from explorer)
2. File->Open and browse to resources/model/Daylighting\_Office folder and open eplusout.sql
3. Check the “Table View” tab for the following:
4. Check the “Tree View” tab for the following:

## RV Test 2

1. Open resources/model/Daylighting\_Office/eplusout.sql as described in RV Test 1
2. Double click on “Air Loop Total Cooling Energy” “PSZ-AC:1” (row 11 in Table View) 
3. Check that a line plot is generated as follows:

## RV Test 3

1. Open resources/model/Daylighting\_Office/eplusout.sql as described in RV Test 1
2. Right mouse click on the Hourly->Air Loop Total Cooling Energy->PSZ-AC:1 node in the Tree View and select “Flood Plot”
3. Check that the following plot is generated: 

# OpenStudio RunManager sanity tests

## RM Test 1: Verify Configuration

1. Launch application
2. Click File->Settings File
3. Choose “Restore Defaults”
4. Hit “OK”
5. Choose Settings -> Preferences
6. Verify default EnergyPlus version and location for shipping EnergyPlus release

## RM Test 2: Queue Add

1. Verify default list of examples is displayed
2. Click on 1ZoneEvapCooler.idf
3. Shift-click on 1ZoneUncontrolled\_win\_2.idf
4. Right-click, choose "Check Selected"
5. Right-click, choose "Add Checked To Queue"
6. Accept EPW warning dialog
7. Verify 10 Jobs added dialog
8. Switch to "Job Queue" tab
9. Verify 10 jobs in queue
10. Verify queue paused

## RM Test 3: Process Queue

1. Click "Start Queue Processing" button to un-pause queue
2. Wait while queue processes
3. Click on "EnergyPlus (1ZoneEvalCooler.idf)"
4. Right-click and choose "Show Job Warnings And Errors"
5. Observe warning dialog, expect to see some warnings and errors
6. Click "OK", closing the dialog
7. Go to "Job Output" tab
8. Verify EnergyPlus generated output

## RM Test 4: Test “PreviewIES” Job

1. Verify that Radiance is installed.
2. Execute the following on the command line: <pathtorunmanager>/RunManager --radiance <pathtoradiancebinaries> --workflow previewies --outpath iesrun --input <pathtotrunk>/resources/runmanager/test.ies
3. Verify that iesrun/PreviewIES/out.bmp exists

# OpenStudio ModelEditor sanity tests

## ME Test 1

1. Start ModelEditor and open the example resources\model\Daylighting\_Office\in.osm file.
2. Navigate to the building node in the tree view.
3. Right-click->Expand All Nodes to expand all the tree nodes.
4. Again using the context menu, add 2 zone objects to the building node.
5. Then use the context menu to add 2 light nodes to one of the zone nodes.
6. Drag one light object to the other zone object.
7. Next, copy and paste one of the light objects to one of the zone objects.
8. Finally, delete one of the zones, remembering how many light objects were its children.
9. Toggle the GUID off and then on using the button in the toolbar.
10. Use InspectorGadget to modify the GUID and note its change in real-time in the tree view.
11. Toggle to the class view, and note if the number of expected zone and light objects are correct.
12. Click the class view column headers to sort the table.
13. Drag and drop an object to change its workspace index.
14. Use File->Export IDF to save the changes as an IDF file

## ME Test 2

1. Starting from the end of ME Test 2, restart ModelEditor and import the IDF file created at the end of the last test.
2. Use the ‘Search groups’ field to search for ‘HVAC’; note the changes in the two results boxes.
3. Use the ‘Object search’ field to narrow groups and objects using the word ‘Plant’
4. Drag one of the objects from the results into the class view, and verify that it has been added to the class view (workspace).
5. Export the IDF file and exit the app.

# OpenStudio Plug-in sanity tests

## OS Test 1

1. Install OpenStudio on a machine with SketchUp 8
2. Delete the OpenStudio registry keys to simulate a clean install; On Windows use regedit to delete ‘HKEY\_CURRENT\_USER\Software\Google\SketchUp8\OpenStudio’ and ‘HKEY\_CURRENT\_USER\Software\Google\SketchUp7\OpenStudio’, on Mac use Property List Editor to delete all SketchUp.OpenStudio keys in ~/Library/Preferences/com.google.sketchup8free.plist and ~/Library/Preferences/com.google.sketchup7free.plist
3. Launch SketchUp
4. You should be prompted to view getting started, press yes and verify you see the document
5. Restart SketchUp and select no at the prompt to see getting started help
6. Select Plugins->OpenStudio->About OpenStudio and verify that it shows the correct version information
7. Select Window->Ruby Console and type ‘OpenStudio::Plugin.dir’, verify that this is the correct installation directory
8. Click the toolbar button to make a new zone
9. Click somewhere in the workspace to create the zone
10. Double click the small tick mark to edit the zone
11. Draw a rectangle in the zone
12. Use the push pull tool to extrude the rectangle vertically
13. Draw a window on one surface
14. Hit the space bar and then escape key to exit out of the zone

## OS Test 2

1. Starting from the end of OS Test 1, save your model as an OSM file.
2. Save your SketchUp file
3. Click the RunManager icon and run a simulation (I’ll have to get you some settings for this)
4. When simulation is done switch render mode to render by data to confirm svn was loaded in.
5. Quit SketchUp
6. Re-launch SketchUp and open the saved OSM file.

# OpenStudio SystemOutliner sanity tests

## SO Test 1

1. Launch application.
2. Click on the plus button at the top of the screen to add a new Single Zone AirLoopHVAC.
3. Browse to a constant volume fan and drag it onto the supply side of the new air loop.
4. Continue by dragging a cooling coil, a heating coil, and an outdoor air system onto the air loop.
5. Use File->Save from the menu bar to save an OSM file to the hard drive.
6. Use File->Exit to quit the program.

## SO Test 2

1. Complete SystemOutliner Test 1.
2. Using File->Open, browse to the saved OSM file from Test 1.
3. Use File->Export IDF, to create an IDF file.
4. Verify that the idf file was created by opening it in a text editor and verify that it is not an empty file.

## SO Test 3

1. Launch the Ruby Console from File->Show Ruby Console.
2. Check that it immediately reports the location of the Ruby libraries.
3. Click the Thermal Zones tab in SO. Then type the following in the console:
   1. t = OpenStudio::Model::ThermalZone.new($m)
4. Verify that a new thermal zone appeared.
5. Enter t.setName(“Testing”), and verify that the name has changed.
6. Enter t.remove() and verify that the thermal zone is gone.

# OpenStudio PolicyAnalysisTool sanity tests

1. Launch OpenStudio PolicyAnalysisTool.
2. Create a new database.
3. Select the Rules tab and verify that it is not displayed.
4. Select the Climate Zone tab, show the CEC map, and check the radio button for “Los Angeles, California”.
5. Select the Building Type tab and check the radio button for “Warehouse”.
6. Select the Perturbations tab, change the selected perturbations and note the number of selected combinations. Check that you cannot deselect all perturbations for each variable, check that you cannot select perturbations to create more than 16 combinations. Make sure that the Null perturbation is selected for all variables, and make sure that at least one alternate roof or wall construction is selected. Note the selected perturbations. The number of perturbations should be the product of selected perturbations for each variable.
7. Select the Rules tab, and click on at least one rule to make sure it displays properly.
8. Begin running the simulations.
9. Cancel running the simulations.
10. Run the simulations to completion.
11. On results page right click on all points and make sure that all expected combinations of perturbations are found. Check that all models fail the ruleset.
12. Open two rules engine html reports: baseline, and one containing a roof or wall construction perturbation. Verify that the u-factor reported for the perturbed construction is different from that for the baseline construction.
13. Select the Rules tab and change the metal building wall u-factor to 3.5, the metal building roof u-factor to 0.6, and the window u-factor to 7.0. Apply the ruleset again. PAT should automatically switch to the analysis tab when complete, and at least the baseline model should now pass the ruleset.
14. Close the application.
15. Launch the application.
16. Open the database used above.
17. Verify that “Los Angeles, California” is selected.
18. Verify that “Warehouse” is selected.
19. Verify that the perturbations selected above are still selected.
20. Verify that the rule changes from above still remain. Verify that re-running the ruleset results in the baseline model still passing.
21. Return to Climate Zone tab, select “DOE”, pick Miami.
22. Check that all results pass.
23. Go to Building tab, select “Small Office”.
24. Select the Perturbations tab, change the selected perturbations and note the number of selected combinations. Deselect the Null perturbations in all variables, note the selected perturbations. Check that you cannot deselect all perturbations for each variable, check that you cannot select perturbations to create more than 16 combinations. The number of perturbations should be the product of selected perturbations for each variable + 1 for the baseline.
25. Run the simulations.
26. On results page right click on all points and make sure that all expected combinations of perturbations are found. Check that all results pass.
27. Verify results as expected.
28. Open up one of the Los Angeles model directories created by PAT, and look at one of the RulesEngine reports. Make sure the result is Pass or Fail (not Incomplete or Error), and that wall, roof, and window rules were triggered.

# OpenStudio Ruby Bindings sanity tests

1. Open a command prompt or terminal and change directories to $INSTALL/ Ruby.
2. Type ‘ruby openstudio\_test.rb’
3. The test suite should run to completion with no failures except for the known test failures:
   1. ValidateIdfs\_Test.rb

# OpenStudio C# Bindings sanity tests (Windows only)

## RunManagerExample

1. Navigate to $INSTALL/CSharp/examples/RunManagerExample/RunManagerExample and open RunManagerExample.csproj using Visual Studio.
2. Open the Solution Explorer and navigate to References, there will be an OpenStudio item with a warning label because the reference is not found, delete this reference.
3. Right click on References and select Add Reference, choose browse and then navigate to and select $INSTALL/CSharp/openstudio.dll
4. Build the project with the x86 configuration, save a solution file when prompted.
5. Copy all dlls in $INSTALL/CSharp/ to the build directory for your solution, e.g. $INSTALL/CSharp/examples/RunManagerExample/RunManagerExample/bin/Debug
6. Press the Start Debugging green arrow to begin running the program. You should see a blank dialog with a Show Config button, push the button and verify it shows the config dialog.
7. Exit the program.

## Alpha1

1. Repeat steps 1-5 of RunManagerExample for $INSTALL/CSharp/examples/Alpha1/Alpha1/Alpha1.csproj.
2. This test is out of date, running does not work.