Structural Completeness

User Tutorial for the Structural Completeness Test Bench and Tool

May 2, 2014

1.0 Purpose

The purpose of the Structural Completeness Test Bench and tool is to evaluate the design for completeness in terms of specification of the mechanisms that join two or more components together. Specifically, this tool evaluates the connection or join information between two or more components and provides a measure of how complete the join definition is for those joins.

The general design process is iterative and often begins by notionally placing components in a location or near the final location to gauge space requirements, fit, and/or overall system structure. These components may or may not be "connected" to other components within the design. This test bench can be used to identify the components that need to be connected at any stage of the design process, i.e. have no connections. It will also be used to evaluate the completeness of the design by analyzing the defined connections (joins) to ensure that a particular connection is fully specified. This test bench will not identify under-connected components, i.e. components that do not have a sufficient number of connections, only that there are either no connections or that the defined connections are defined sufficiently.

2.0 Procedures

The instructions in this manual assume that the user has installed the latest version of GME/CyPhy and the python libraries installed with those tools.

2.1 Installation

Initial installation of the Structural Completeness Tool will be provided with the installation of the CyPhy tool suite. Future editions of the tool may be packaged as a standalone or combined test bench installation package.

The test bench in GME that engages this tool is provided as a separate download from VehicleForge.

2.2 Tool

The Structural Completeness Test Bench is the test bench in GME that the designer uses to interface with the Structural Completeness Tool. This tool is a software application that accepts a design, analyzes the defined component connections and returns a percentage score for structural completeness.

The software uses the following equation to calculate the metric: (number of connections – connections with incomplete data – connections with missing data)/number of connections = structural completeness

3.0 Requirements Tested

- **Structural Completeness:** The percent of joins or structural connections that have been fully specified sufficient to fabricate the design.

4.0 Required Components

There must be at least one component in the system under test, however this component can be of any type.

5.0 Theory of Operation

The user specifies connections between components either explicitly using a join block in CyPhy or implicitly using the connectors and interfaces associated with AVM components. This testbench will evaluate the design to determine the percentage of those connections that have been specified to a sufficient level of detail to fabricate/manufacture the design.

6.0 Test Bench Structure

This test bench contains a system under test that is to be assembled and analyzed for its connection (join) completeness.
The Structural Completeness test bench is included in the Official Test Benches located on gamma.vehicleforge.org. Steps 1-6 below show how to instantiate the test bench if the designer would like to re-create the test bench. Steps 7-8 discuss running the test bench and viewing the output from the tool.

Step 1

In the GME Browser, within the "Testing" **Test Bench folder**, insert a new **Test Bench subfolder** ("Structural Completeness "). Then insert a new **Test Bench model** ("Structural_Completeness "). Figure 1 shows the new test bench and model.

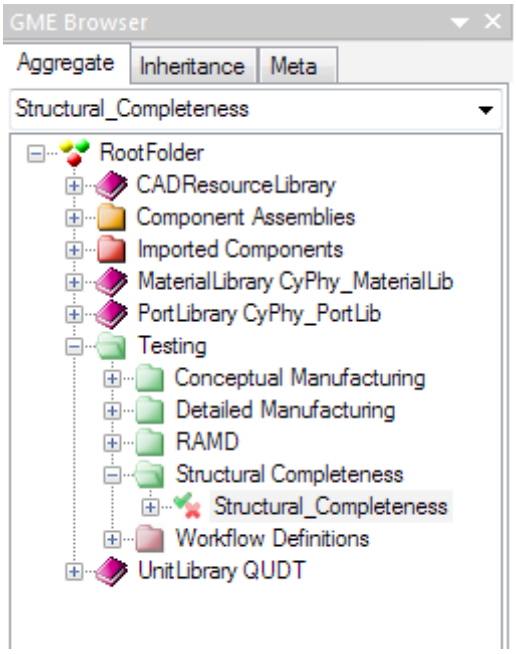

Figure 1: Structural Completeness Test Bench and Model.

Step 2

Now that the test bench container has been created, the design can be added. In the "Structural_Completeness" testbench, Copy/Paste…As Reference your assembly. Figure 2 shows the test model in the test bench.

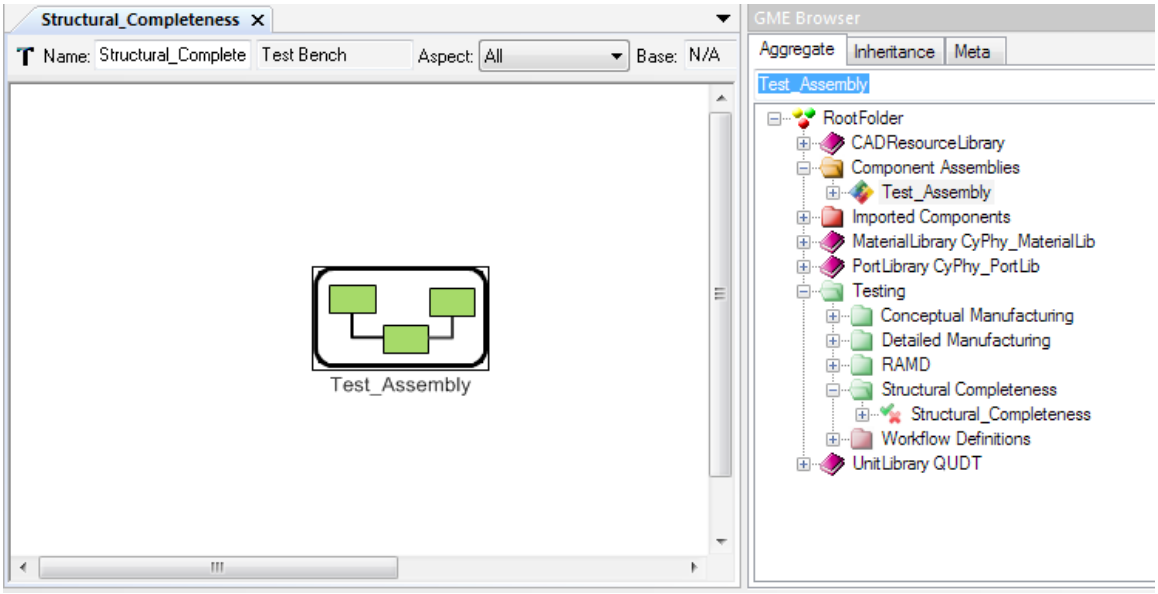

Figure 2: Copy/Paste…as Reference the Design Assembly.

Step 3

In the GME Browser, in the "Workflow Definition" subfolder create a new **Workflow Model** ("Completeness") as shown in Figure 3.

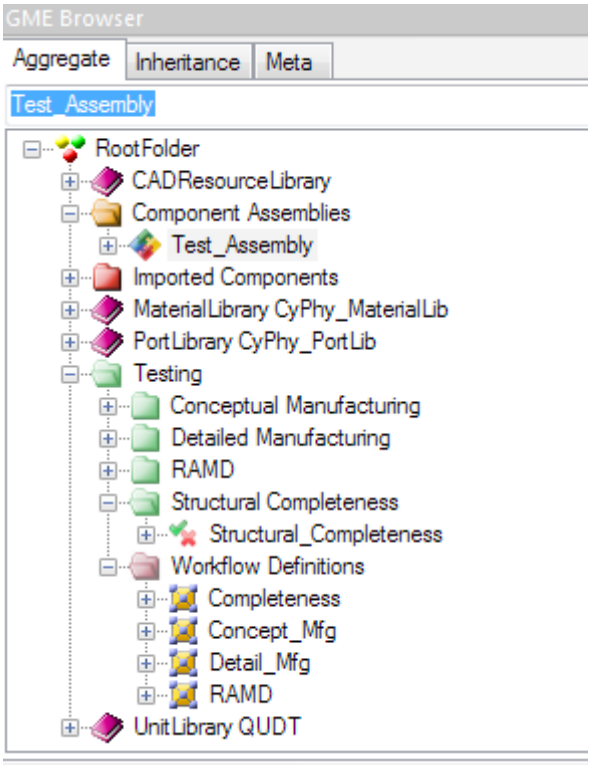

Figure 3: Workflow Definition (Completeness).

Step 4

Open the “Completeness” **Workflow Model**, drag a "Task" element into the workspace window, and select "CyPhyCADAnalysis" as the interpreter from the pop-up window. The Completeness Workflow Model is shown in Figure 4.

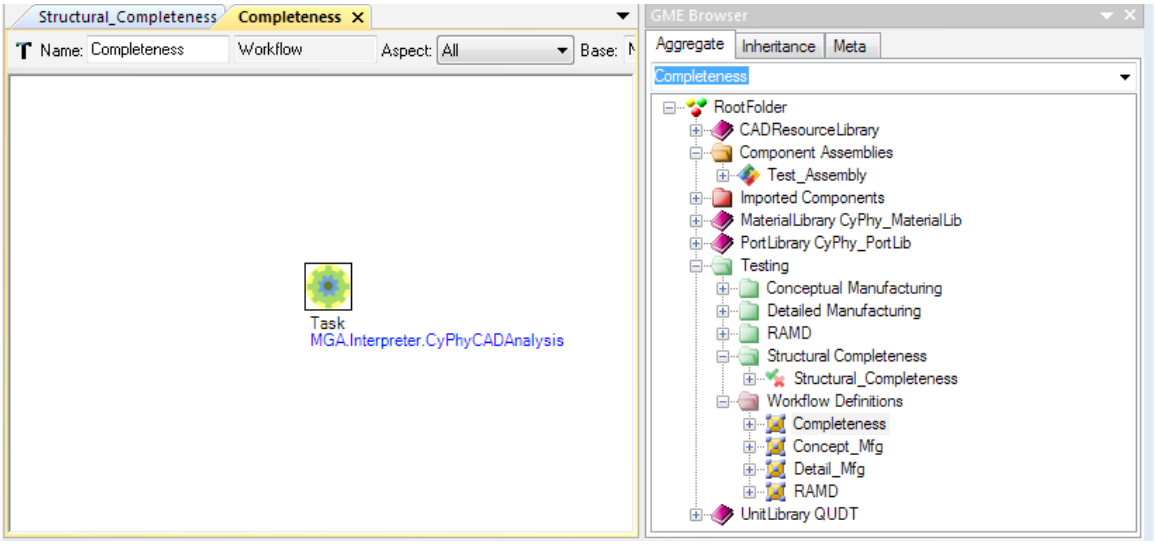

Figure 4: Completeness Workflow Model with Task.

Step 5

Setup the workflow parameters by Double-clicking on the newly created task, select "StructuralCompleteness" as the analysis tool. After exiting object inspector parameter should mimic Figure 5.

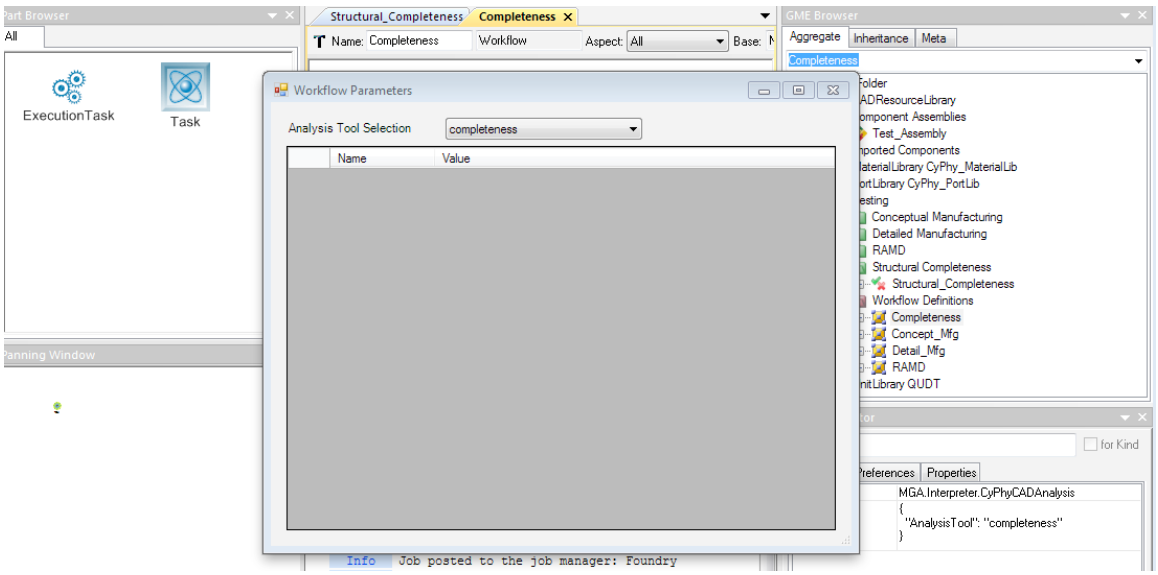

Figure 5: Setting up the Metrics.

Step 6

Now open the “Structural_Completeness” test bench and drag/drop the “Completeness” workflow definition and one (1) metric (from Part Browser) into test bench. Metric must be named “Structural_Completeness” as shown in Figure 6 (this is what is searched for in the testbench_manifest.json file).

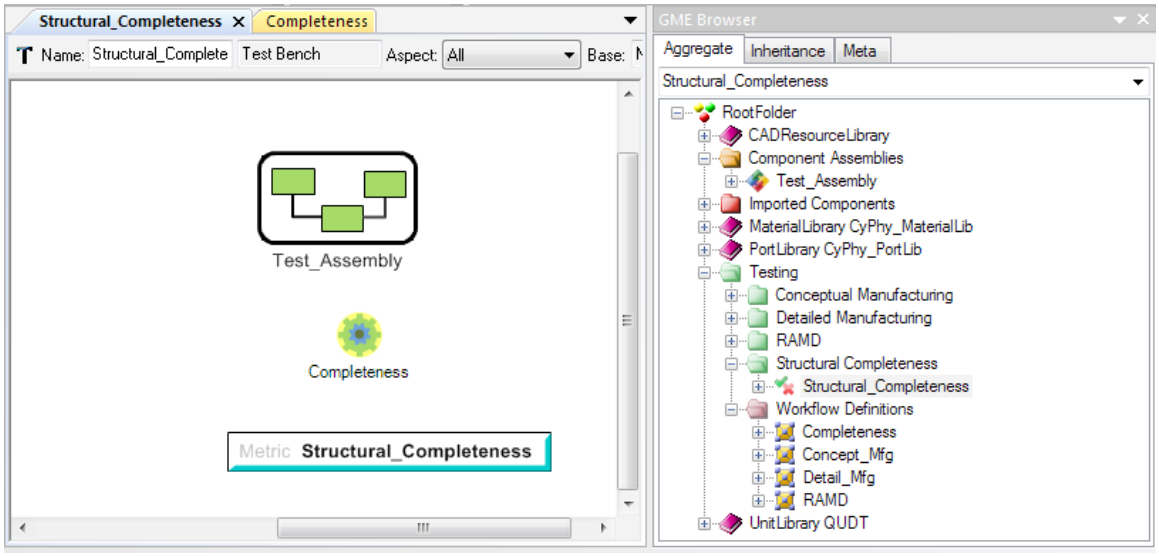

Figure 6: “Structural_Completeness” Test Bench.

Step 7

To exercise the test bench, run the Master Interpreter (highlighted in task bar) as shown in Figure 7.
Check the "Post to META Job Manager" in the pop-up shown in Figure 7.

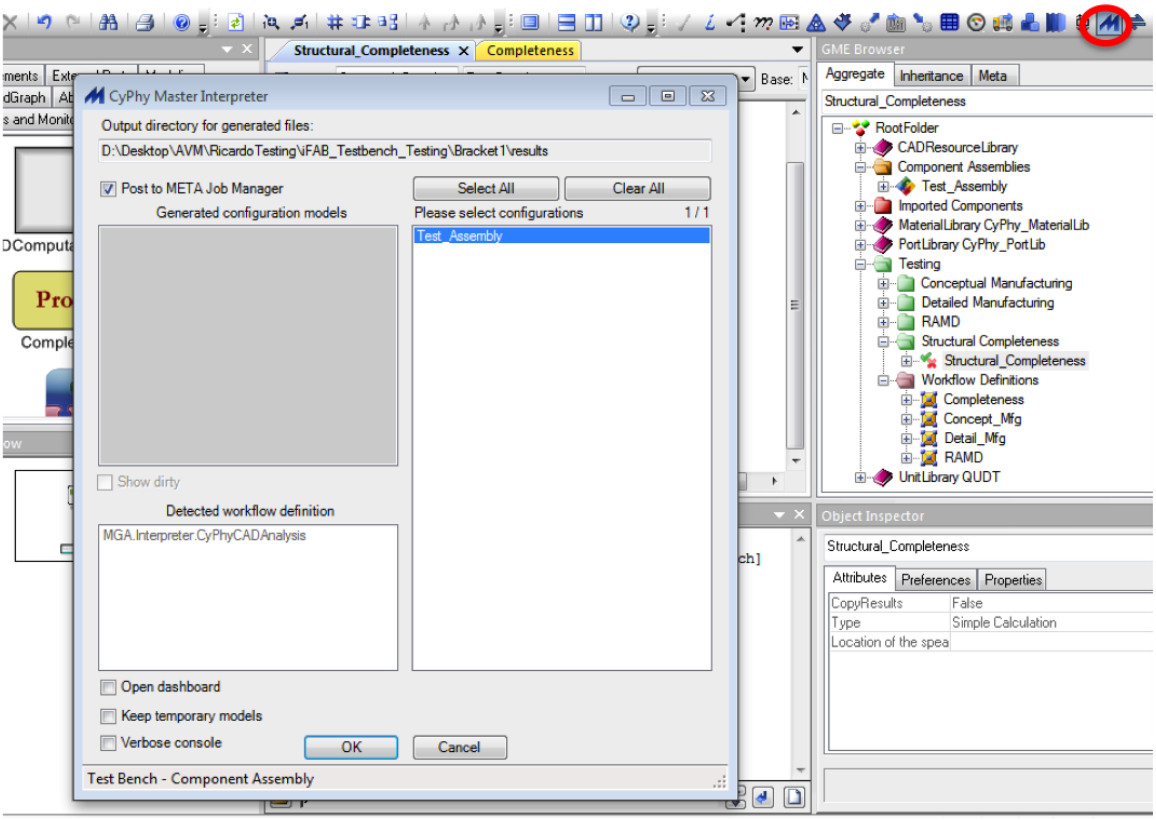

Figure 7: Running the Test Bench.

After selecting “OK” another pop-up will appear to set up the test bench run. Select the “Use Project Manifest” as well as the “AP203_E2_Single_File” and “AP203_E2_Separate_Part_Files” boxes as shown in Figure 8. Also, for any files created by the user that are not imported into GME, point to the Creo file location for the “CAD Auxiliary Directory:” field. After entering this information, your GME screen should mimic Figure 8.

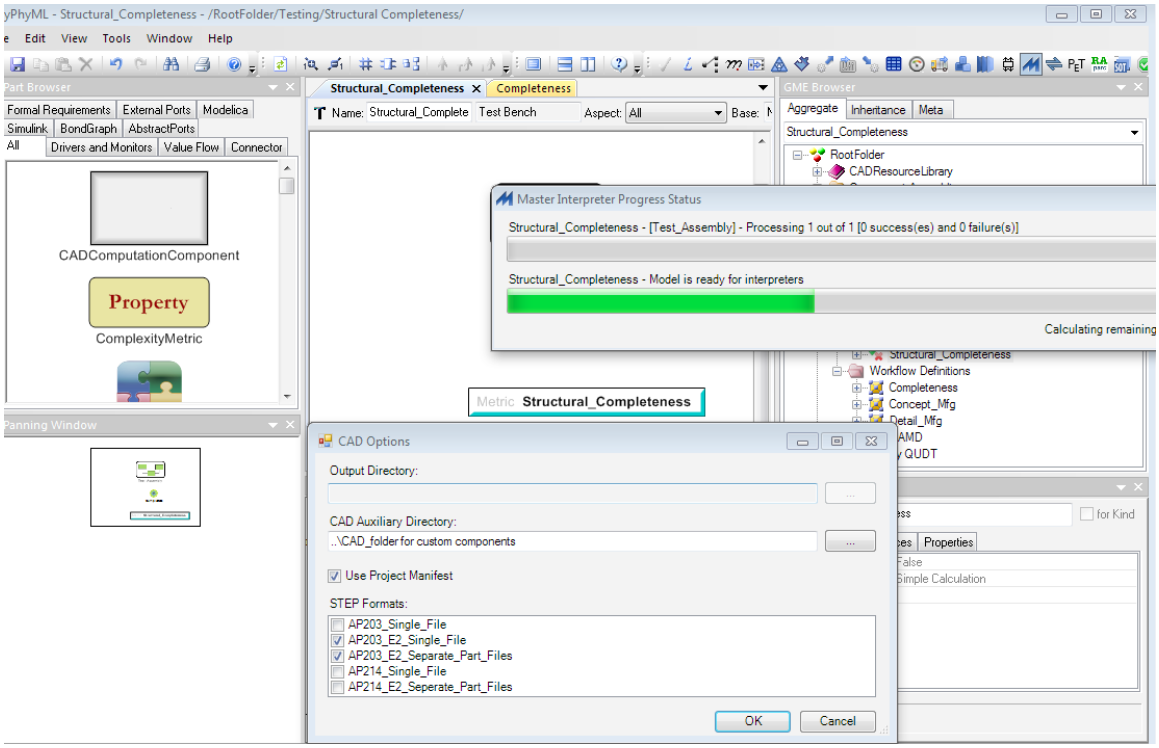

Figure 8: Setting up the Testbench Run.

Select “OK”.

The JobManager Configuraion pop-up window will appear. This test bench can be run either locally (if Creo is installed) or remotely. If running locally, uncheck the "Remote Execution" box and click "Save." If running on server, check "Remote Execution" box, and enter your Vehicle Forge credentials (i.e., URL, user name, and password) shown in Figure 9. Finally, Click "Save".

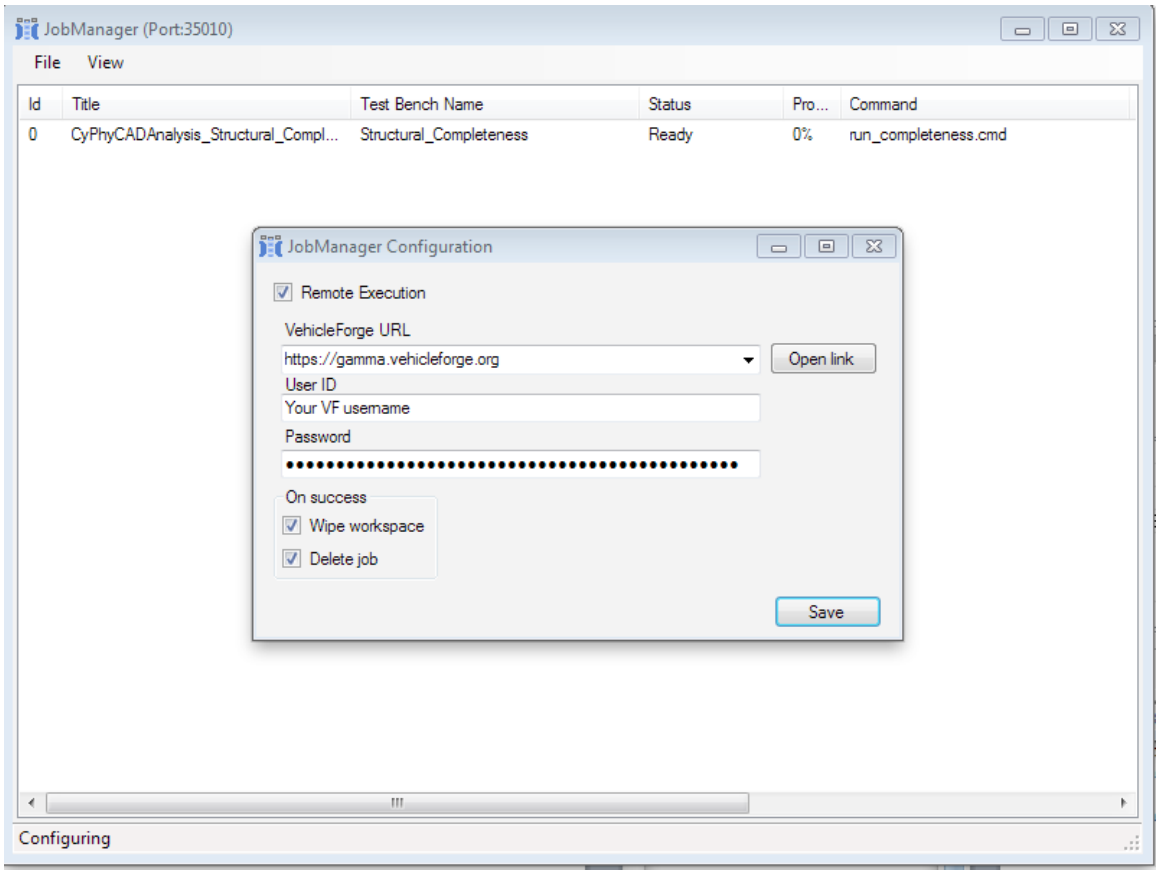

Figure 9: Setting up Remote Execution.

Step 8

The test bench will create a results folder with the structural completeness information about the design. The Job Manager will show the specific results folder which will contain the testbench_manifest.json output file. The testbench_manifest.json file contains cost and structural completeness information that can be used to evaluate the design. To get to the results folder, right-click on the analysis in the Job Manager and select "Show in Explorer".

Typical analysis times are between 5 and 10 min and increase with the number of components in the design.
Figure 10 shows an example completeness output visualizer. There are two ways to use this visualization: on the user's local machine and through VehicleForge.

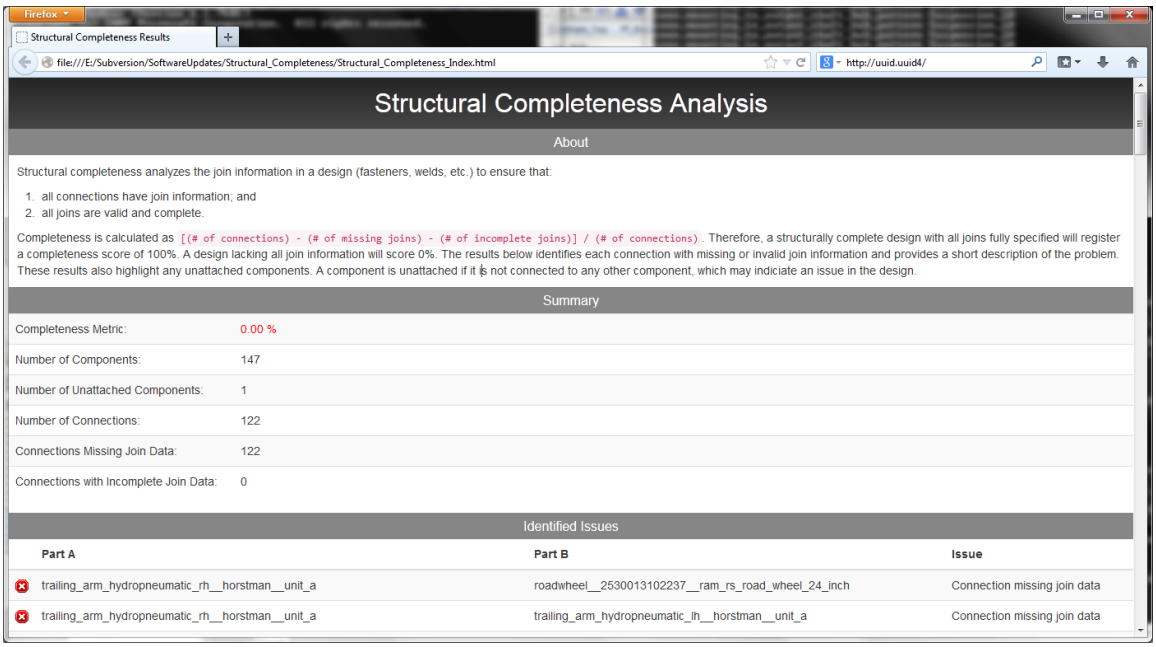

Figure 10: Structural Completeness Output Visualizer.

To view the results locally, navigate to the results folder of the structural completeness test bench output and double click on the Structural_Completeness_Index.html file. The html viewer works with Firefox and Internet Explorer browsers. Other browsers (Chrome) can be used by setting the "target" for the shortcut startup to: "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" –allow-file-access-from-files. You will need to change the file path of the installation location for Chrome is different on your machine.
To view the results through Vehicle Forge, check-in the results folder to your SVN or GIT repository on Vehicle Forge and click on the testbench_manifest.json.

7.0 Description

Structural Completeness is one (of several) test benches that can be used to evaluate the completeness of a design. Ideally, several test benches would be incorporated to measure the completeness of a design as it progresses to a fully defined and specified design that is ready to be manufactured. This test bench is one of the test benches in the "suite" that measures the completion of the specified connections between components.

Results are returned in the "testbench_manifest.json" file. This file contains the completeness metric in terms of a percentage complete. The summary results can be viewed in the dashboard.

8.0 Metrics

Structural Completeness (% Complete): a number between 0 and 100, where 100 is fully structural complete and 0 is not complete at all. In other words, percentage complete.

Test Bench #	Metric	Description
23	Structural_Completeness	The percentage of structural joins that are complete

9.0 Required Connections to System Under Test

NONE

10.0 Outputs

The output of this test bench is the "testbench_manifest.json" file. The summary results summarizes the structural completeness as a number between 0 and 100, where 100 is fully defined and 0 has no structural join completeness. A portion of the summary results is shown in Figure 11.


```
{
    "Status": "OK",
    "CopyTestResults": false,
    "VisualizationArtifacts": [],
    "Parameters": [],
    "TierLevel": 0,
    "Artifacts": [],
    "AnalysisStatus": "OK",
    "Created": "2014-01-13T17:15:02.7721124Z",
    "DesignName": "Meta_assembly",
    "Metrics": [
        {
            "GMEID": "id-0067-000018f4",
            "Description": "",
            "DisplayedName": null,
            "VisualizationArtifacts": [],
            "Value": "100.0",
            "ID": "7bb18524-df0d-4876-aef0-badc9cba552c",
            "Unit": "%",
            "Name": "Structural_Completeness"
        }
    ],
    "DesignID": "{90b2b46c-0183-431a-b239-7c67ee39a281}",
    "Steps": [
        {
            "ExecutionCompletionTimestamp": null,
            "Description": null,
            "Parameters": [],
            "ExecutionStartTimestamp": null,
            "Invocation": "run_completeness.cmd",
            "PostProcess": null,
            "PreProcess": null,
            "Type": null
        }
    ],
    "TestBench": "Completeness"
}
```


Figure 11: Portion of Summary Results.

The summary results can be seen and compared with other system metrics in the Dashboard. For further information on using the dashboard, please go to [the Dashboard page] in the Mass Spring Damper tutorial.