# Structural Completeness Test Bench and Tool

Table of Contents



## 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 metric of structural completeness for the design.

The 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 testbench can be used to identify the components that need to be connected at any stage of the design process. It will also be used to evaluate the completeness of the design by analyzing the defined connections (joins) to ensure that the connection is fully specified.

## 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 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.

### 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, undefined, and under defined component connections and returns essentially 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

# Test Bench: Structural Completeness

## Requirements Tested

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

## 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 the design.

## Test Bench Structure

This test bench contains a system under test that is to be assembled and analyzed for its connection (join) completeness.

## Step 1

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

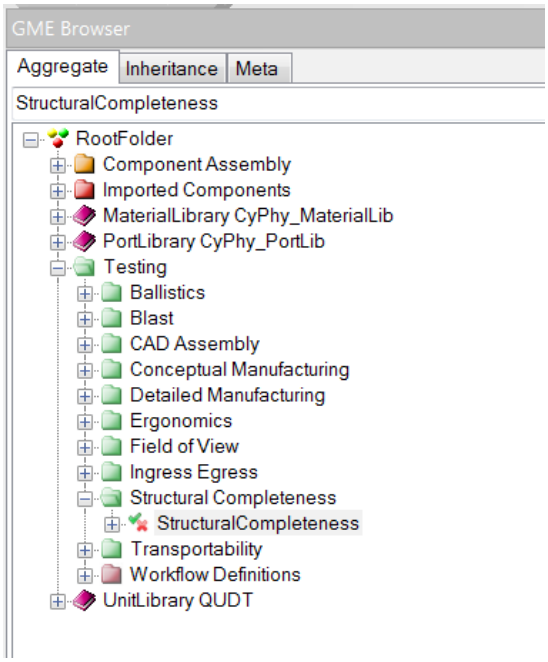


*Figure 1: StructuralCompleteness Test Bench and Model (highlighted).*

## Step 2

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

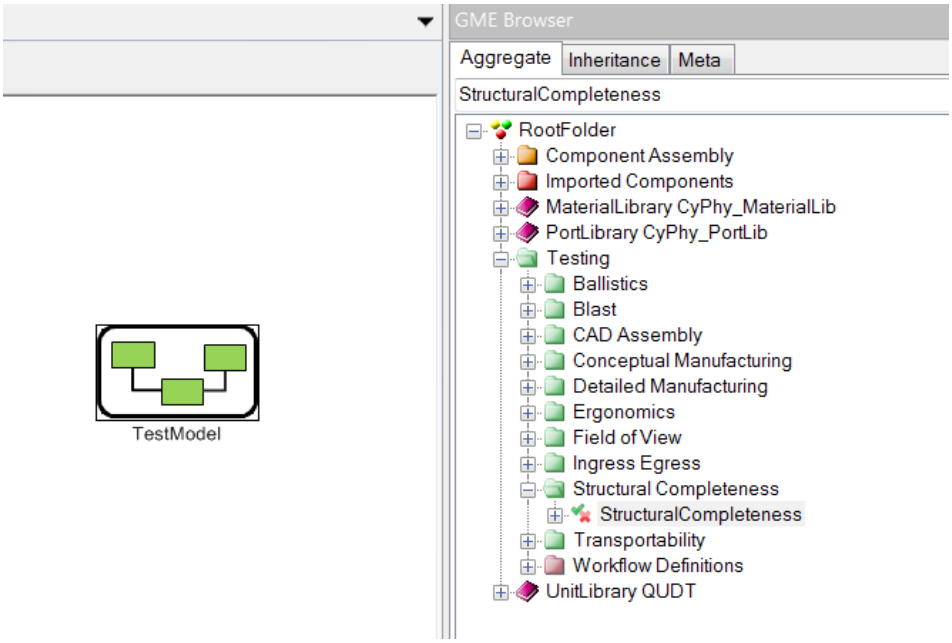


*Figure 2: Copy/Paste…as Reference the Design Assembly.*

## Step 3

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

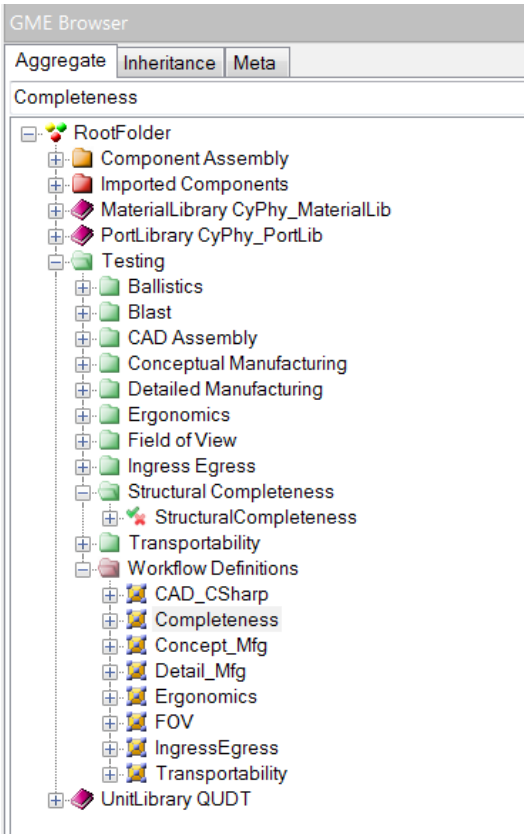


*Figure 3: Workflow Definition (highlighted).*

## 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.

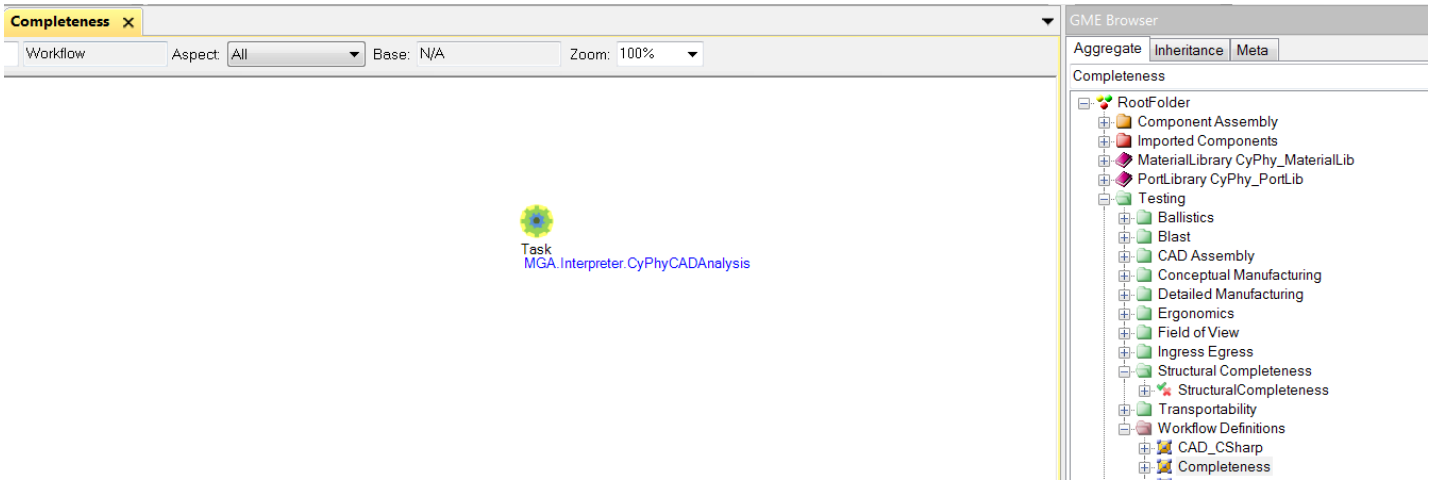


*Figure 4: Completeness Workflow Model with Task.*

## Step 5

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

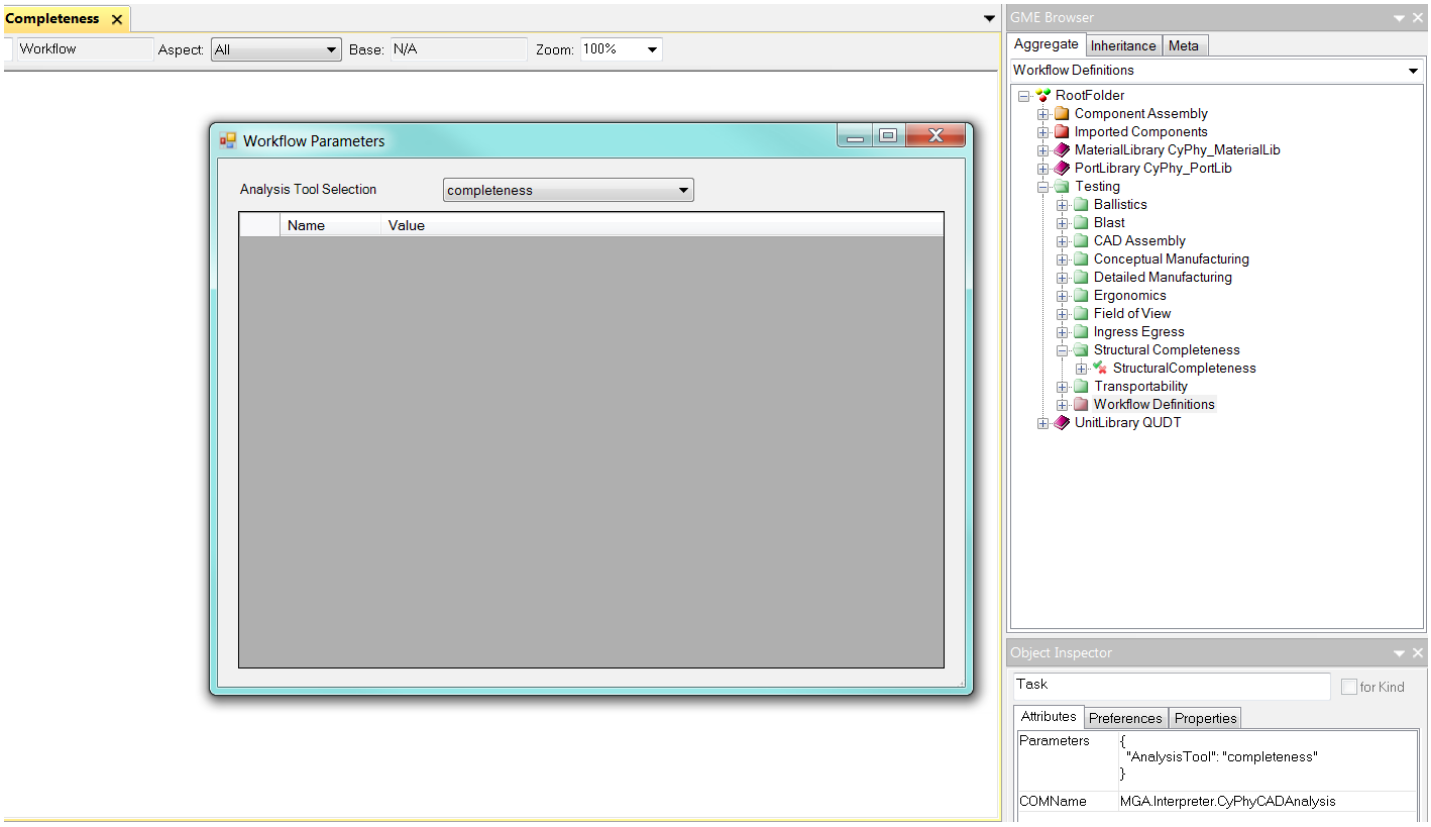


*Figure 5: Setting up the Metrics.*

## Step 6

Now open the “StructuralCompleteness” test bench and drag/drop the “StructuralCompleteness” workflow definition and 2 metrics (from Part Browser) into test bench. Metrics must be named “StructuralCompleteness” as shown in Figure 6 (this is what is searched for in the summary.testresults.json file).

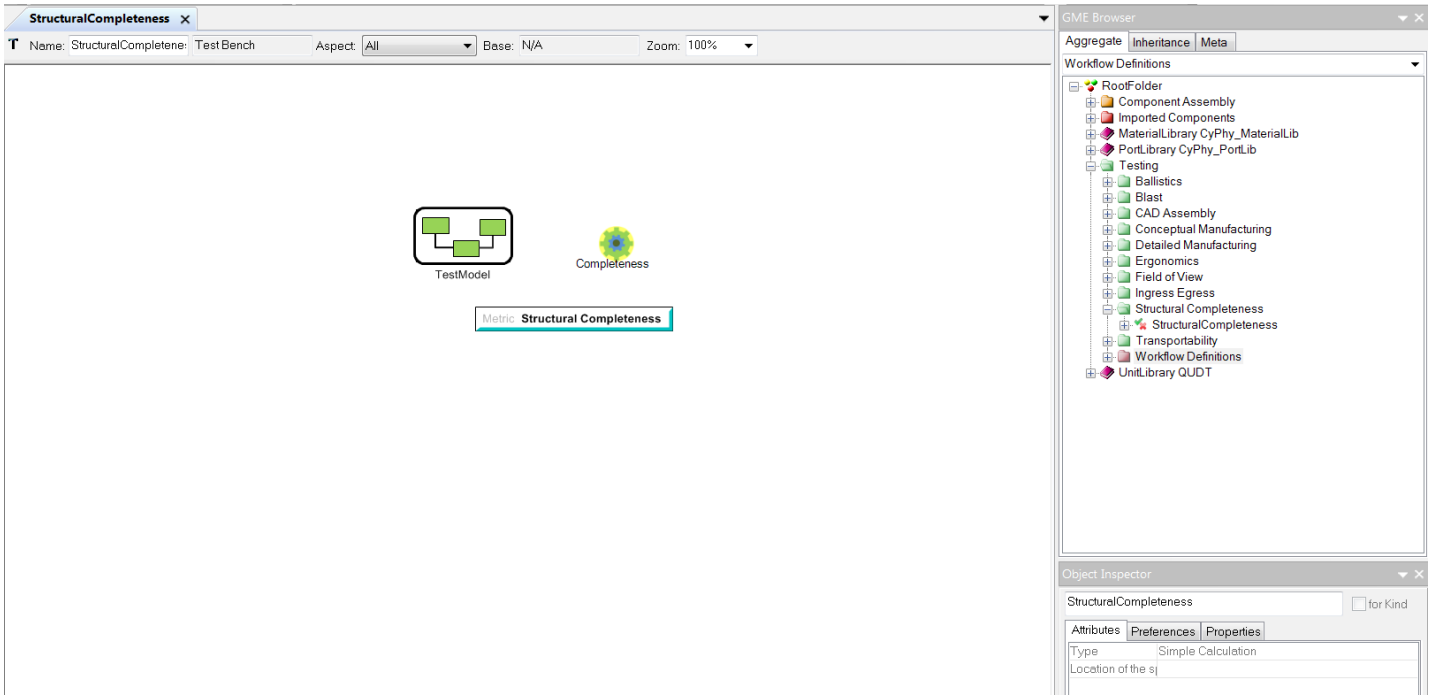


*Figure 6: “StructuralCompleteness” Test Bench.*

## Step 7

To exercise the test bench, run the Master Interpreter (highlighted in task bar). Check the “Post to META Job Manager” and “Use Project Manifest” boxes as well as the “AP203_E2_Single_File” and “AP203_E2_Separate_Part_Files” as shown in Figure 7. Also point to the Creo file location for the “CAD Auxiliary Directory:” field.

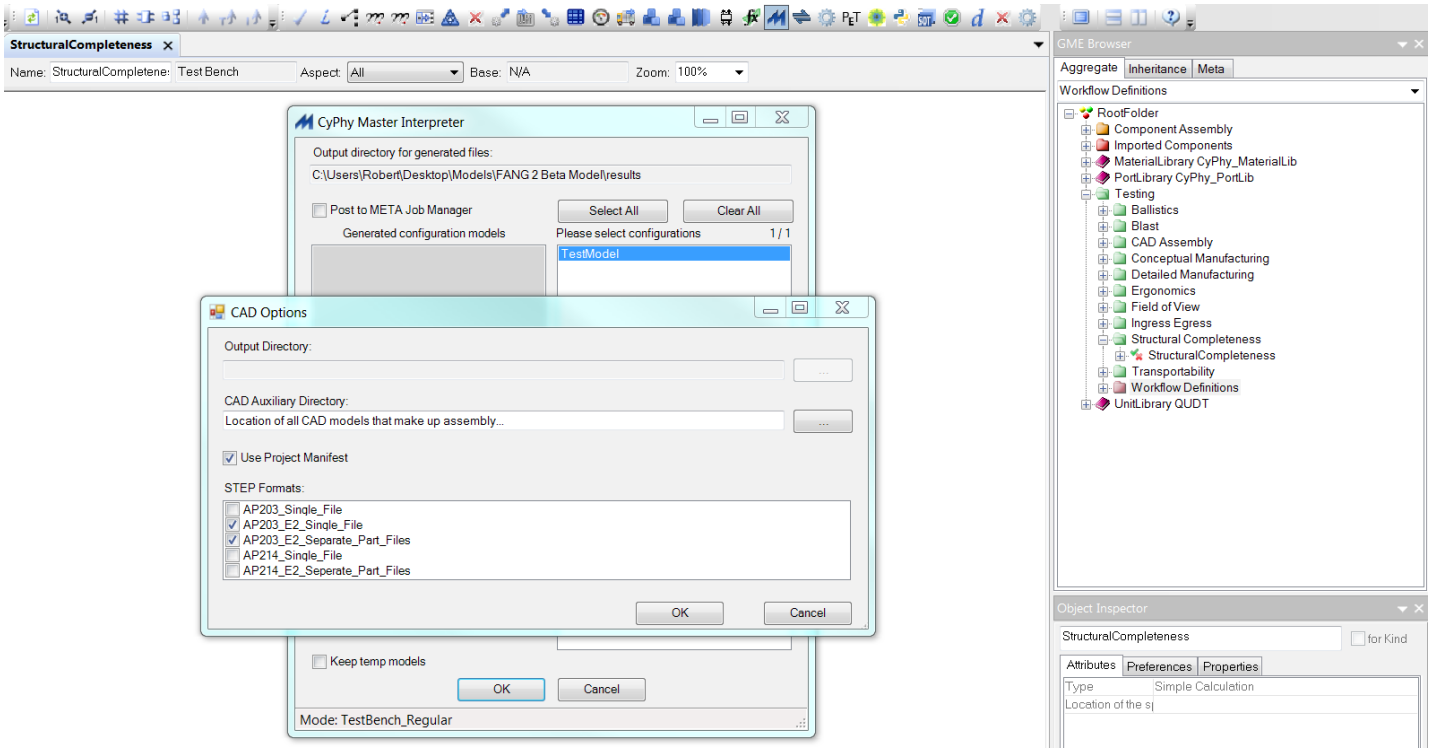


*Figure 7: Running the Test Bench.*

### Step 8

The test bench will create a results folder with the structural completeness information about the design.

# Description

Structural Completeness is one (of several) testbenches that can be used to evaluate the completeness of a design. Ideally, several testbenches 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 testbench is one of the testbenches in the “suite” that measures the completion of the specified connections between components.

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

## Metrics

- **Structural Completeness (N/A):** a number between 1 and 0, where 1 is fully structural complete and 0 is not complete. In other words, percentage complete.

## Required Connections to System Under Test

NONE

## Outputs

The output of this test bench is the “summary.testresults.json” file. The summary results summarizes the structural completeness as a number between 0 and 1, where 1 is fully defined and 0 has no structural join completeness. A portion of the summary results is shown in Figure 8.


```
"AnalysisStatus": "OK",
"TestBench": "StructuralCompleteness",
"DesignName": "StructuralCompleteness",
"Metrics": [
    {
        "Name": "Structural Completeness",
        "DisplayedName": null,
        "GMEID": "id-0067-00028009",
        "Value": "0.266666666667",
        "ID": "843e9197-1460-4bea-bcb9-32fd15c9d1cd",
        "Unit": ""
    }
],
"DesignID": "{def04a1e-dc80-4852-a2df-52bb7a66cdfd}",
"Design": "StructuralCompleteness.metadesign.json",
"Details": "",
"Time": "2013-10-07T20:42:27.0208817Z"
```

*Figure 8: Portion of Summary Results.*

The summary results can be seen and compared with other system metrics in the Dashboard.