

# Running Tests

CSi.Testing

Exported on 06/24/2018

# Table of Contents

<b>1</b>	<b>Source Control Testing Directory Structure .....</b>	<b>4</b>
1.1	Models Database Directories.....	4
1.2	_Tests Directories.....	5
1.3	_Tests\Testing Directories.....	6
<b>2</b>	<b>Testing Quickstart.....</b>	<b>9</b>
2.1	Testing .....	9
2.2	Results .....	9
2.3	Archive .....	10
2.4	Deploying Testing Directories .....	10
2.4.1	Deployment Scripts .....	10
2.5	Running Tests.....	11
2.5.1	Generic Builds .....	11
2.5.2	Specific Builds .....	11
2.5.3	Example with Generic Builds.....	11
2.6	Viewing Test Results .....	12
2.6.1	Example with ETABS Analysis and Design Suites .....	12
2.7	Archiving Test Results.....	12

The child pages describe various details related to the running, reviewing, and archiving of test results.

**On this page:**

# 1 Source Control Testing Directory Structure

The following is the structure of the testing directories stored through source control. Directories that are not version controlled are created by running the deployment batch files after getting the latest files from source control.

- Names within curly braces { } are generic and there may be more than once occurrence.
- Names within brackets [] are optional.
- Names starting with \_ tend to be auxiliary directories or contain supporting files that are deployed after pulling from source control.

## On this page:

- [Models Database Directories](#) (see page 4)
- [\\_Tests Directories](#) (see page 5)
- [\\_Tests\Testing Directories](#) (see page 6)

## 1.1 Models Database Directories

The following is the structure of all directories stored under 'Models Database' in source control. These files all relate to the testing of CSi programs.

Path	Files Under Version Control?	Description
.\Models Database	Yes	Root node in source control.
.\Models Database\_Tests	Yes	<b>Contains all files and directories relevant to running and reviewing tests in an organized way.</b>
.\Models Database\_Tests\Archive	No	Contains log files and html summary of test runs.
.\Models Database\_Tests\Results	No	Contains complete results for any regTest runs.
.\Models Database\_Tests\Testing	Yes	Contains batch files for running all suites of a given program (or several), in parallel or series.
.\Models Database\Examples Development	No	<b>Example models under development, which are not yet checked in to source control.</b>
.\Models Database\Examples Development\{Program Name}	No	Contains example models under development, organized by program.

<b>.\Models Database\{Program Name}</b>	<b>Yes</b>	<b>All model files checked in to source control that are used for running tests.</b>
<b>.\Models Database\{Program Name}\{Suite Name}</b>	<b>Yes</b>	Contains all model files used for testing, organized by program and suite.

## 1.2 \_Tests Directories

The following is the structure of all directories stored under 'Models Database\\_Tests', which contains all files and directories relevant to running, reviewing, and archiving tests in an organized way.

<b>Path</b>	<b>Files Under Version Control?</b>	<b>Description</b>
<b>.\Models Database\_Tests</b>	<b>Yes</b>	<b>Contains all files and directories relevant to running, reviewing, and archiving tests in an organized way.</b>
<b>.\Models Database\_Tests\Archive</b>	<b>No</b>	<b>Contains log files and html summary of test runs.</b>
<b>.\Models Database\_Tests\Archive\{Program Name} v{version} Build {build}</b>	No	Contains results organized by program name, version, and build.
<b>.\Models Database\_Tests\Archive\{Program Name} v{version} Build {build}\{bit}-bit</b>	No	Contains results organized by 32-bit or 64-bit runs.
<b>.\Models Database\_Tests\Archive\{Program Name} v{version} Build {build}\{bit}-bit\{Suite Name}</b>	No	Contains all log files of the run, and the html run summary results for 32-bit runs of the suite using the program of a particular version and build.
<b>.\Models Database\_Tests\Results</b>	<b>No</b>	<b>Contains complete results for any regTest runs.</b>
<b>.\Models Database\_Tests\Results\Examples Development</b>	No	
<b>.\Models Database\_Tests\Results\Examples Development\{Program Name}</b>	No	Single run suite results of examples under development, organized by program name.
<b>.\Models Database\_Tests\Results\{Program Name}</b>	No	

.\Models Database\_Tests\Results\ {Program Name}\{Suite Name}	No	
.\Models Database\_Tests\Results\ {Program Name}\{Suite Name}\{Single}	No	Single run results of the complete suite.
.\Models Database\_Tests\Results\ {Program Name}\{Suite Name}\{PSB}	No	PSB run results of the suite.
<b>.\Models Database\_Tests\Testing</b>	<b>Yes</b>	<b>Contains builds, XML files, and batch files for running all suites of a given program (or several), in parallel or series.</b>

### 1.3 \_Tests\Testing Directories

The following is the structure of all directories stored under 'Models Database\\_Tests\Testing', which contains all program and regTest builds, all instruction XML files, and all batch files for running all suites of a given (or several) program(s), in parallel or series.

Path	Files Under Version Control?	Description
<b>.\Models Database\_Tests</b>	<b>Yes</b>	<b>Contains all files and directories relevant to running and reviewing tests in an organized way.</b>
<b>.\Models Database\_Tests\Testing</b>	<b>Yes</b>	<b>Contains builds, XML files, and batch files for running all suites of a given program (or several), in parallel or series.</b>
<b>.\Models Database\_Tests\Testing\_builds</b>	<b>No</b>	<b>Contains all builds to be used for testing.</b>
.\Models Database\_Tests\Testing\_builds\ <b>{Program Name}</b>	No	Contains all builds to be used for testing for a given program. Also contains auxiliary files for testing builds, such as level, licence, and native imaging files for a given program.
.\Models Database\_Tests\Testing\_builds\ {Program Name}\ <b>{Program Name} Build</b>	No	Generic build to be use for testing. Clear this directory and copy all files from the desired build into this directory.

.\Models Database\_Tests\Testing\_builds\ {Program Name}\{ <b>Program Name</b> <b>v{version} [Build {build}] {bit}-bit</b>	No	Specific build to be used for testing. These are copied down from where the builds are stored on the network.
<b>.\Models Database\_Tests\Testing\_regTest</b>	<b>Yes</b>	<b>regTest files to use for testing.</b>
.\Models Database\_Tests\Testing\_regTest\previ ous_test_results	<b>Yes</b>	
.\Models Database\_Tests\Testing\_regTest\previo us_test_results\{ <b>Program Name</b>	<b>Yes</b>	regTest previous test results, organized by program, to use for testing. All builds of regTest deployed will be updated with these results.
.\Models Database\_Tests\Testing\_regTest\regTe st	<b>Yes</b>	regTest program to use for testing. Clear this directory and copy all files from the desired build into this directory.
<b>.\Models Database\_Tests\Testing\_src</b>	<b>Yes</b>	<b>Contains source files for deployment.</b>
.\Models Database\_Tests\Testing\_src\_localBatc hFiles	<b>Yes</b>	Batch files used to run regTest that are copied in deployment to the local setup. These are re-used among the different setups.
.\Models Database\_Tests\Testing\_src\Examples Development	<b>Yes</b>	Source files unique to 'Example Development' runs.
.\Models Database\_Tests\Testing\_src\Examples Development\regTest-Development	<b>Yes</b>	Destination directory for local regTest copy and all XML files used for development model test runs.
.\Models Database\_Tests\Testing\_src\PSB	<b>Yes</b>	Source files unique to 'PSB' runs.
.\Models Database\_Tests\Testing\_src\PSB\_src	<b>Yes</b>	Batch files used to run regTest for each PSB setting that are copied in deployment to the local setup. These are re-used among the different PSB setups for each suite.
.\Models Database\_Tests\Testing\_src\PSB\ { <b>Suite Name</b>	<b>Yes</b>	
.\Models Database\_Tests\Testing\_src\PSB\{Suite Name}\_regTestXml	<b>Yes</b>	All XML files used for PSB runs of the corresponding suite.
.\Models Database\_Tests\Testing\_src\Single	<b>Yes</b>	Source files unique to 'Single' runs.

.\Models Database\_Tests\Testing\_src\Single\reg Test-{Suite Name}	Yes	Destination directory for local regTest copy and all XML files used for suite model test runs.
.\Models Database\_Tests\Testing\Examples Development	No	<b>Directory for examples under development. This is created by deployment scripts.</b>
.\Models Database\_Tests\Testing\Examples Development\regTest-Development	No	Contains run batch file, regTest & local instructions for examples under development.
.\Models Database\_Tests\Testing\PSB	No	<b>Directory for examples run under the 9 PSB analysis cases. This is created by deployment scripts.</b>
.\Models Database\_Tests\Testing\PSB\ {Suite Name}	No	
.\Models Database\_Tests\Testing\PSB\ {Suite Name}\regTest-P{n}S{n}B{n}[ - Standard]	No	Contains run batch file, regTest & local instructions for running all 9 analysis settings variations in parallel or series. The only irregular directory naming is regTest-P2S2B1 - Standard. All others are named as regTest-P{n}S{n}B{n}.
.\Models Database\_Tests\Testing\Single	No	<b>Directory for examples run under the single default analysis case. This is created by deployment scripts.</b>
.\Models Database\_Tests\Testing\Single\regTest- {Suite Name}	No	Contains run batch file, regTest & local instructions for running all suites in parallel or series.



## 2 Testing Quickstart

This page describes the testing setup organization and procedures in a brief and simple way to quickly run tests of the most common and generic type.

### On this page:

- [Testing](#) (see page 9)
- [Results](#) (see page 9)
- [Archive](#) (see page 10)
- [Deploying Testing Directories](#) (see page 10)
  - [Deployment Scripts](#) (see page 10)
- [Running Tests](#) (see page 11)
  - [Generic Builds](#) (see page 11)
  - [Specific Builds](#) (see page 11)
  - [Example with Generic Builds](#) (see page 11)
- [Viewing Test Results](#) (see page 12)
  - [Example with ETABS Analysis and Design Suites](#) (see page 12)
- [Archiving Test Results](#) (see page 12)

For testing, everything you will work with or view is within:

`.\TFS\Models Database\_Tests`

You can see in general what the directory structures contain within [Source Control Testing Directory Structure](#) (see page 4).
















### 2.1 Testing

This directory is where you will place updated builds and initialize test runs.

### 2.2 Results

This directory contains all results of any test.

Within 'Testing' there are handy shortcuts that jump you to the correct sub-directory within this directory for viewing results of a test you have just run.

 Results-ETABS-Analysis	6/1/2018 12:10 PM	Shortcut
 Results-ETABS-Design	6/1/2018 12:10 PM	Shortcut
 Results-ETABS-Development	6/1/2018 12:10 PM	Shortcut
 Results-ETABS-Regression	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-PSB-Analysis	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-PSB-ANSYS	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-PSB-Design	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-PSB-NAFEMS	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-PSB-Regression	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-Analysis	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-ANSYS	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-Design	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-Development	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-NAFEMS	6/1/2018 12:10 PM	Shortcut
 Results-SAP2000-Regression	6/1/2018 12:10 PM	Shortcut

## 2.3 Archive

This likely won't be of concern in most cases, but this is where archives of test results are being added. The process is being automated to make it faster, more consistent, and more organized. In order to contribute, please see [ResultsArchiver<sup>1</sup>](#).

## 2.4 Deploying Testing Directories

For greater organization, automation, and ease of transferring to other testing machines, there is a testing directory under source control that contains the basic files, structure, and deployment scripts. In this way, on any machine where testing is to be done, all that is necessary is the following:

1. 'Get latest' from source control for 'Models Database\' of the desired program to pull the necessary model testing files down to the local machine.
2. 'Get latest' from source control for 'Models Database\\_Tests' to pull the necessary testing directory files down to the local machine.
3. Run the \_Deploy- script under '\.TFS\Models Database\\_Tests'. There are several to choose from, which are described below.

### 2.4.1 Deployment Scripts

The following are the main scripts used to deploy the testing directories. For finer control to create a more limited setup for specific testing, see the scripts contained deeper within the directories. The following are global scripts that call these more local scripts.

1. **\_Deploy-PSB-All.bat**
  - a. This deploys all files and folders related to testing all programs where PSB tests are applicable. This may take some time and require a large amount of available space.
2. **\_Deploy-Single-All.bat**
  - a. This deploys all files and folders related to single test runs of all programs, including examples development.
3. **\_Deploy-Testing-All.bat**

<sup>1</sup> <http://localhost:8090/display/CSIT/ResultsArchiver>

- a. This script runs all deployment scripts. Only run this one if you want to be able to run both single, development, and PSB tests as it may take a while and require a large amount of available space.

## 2.5 Running Tests

### 2.5.1 Generic Builds

The simplest way to test a build is to just do the following steps:

1. Copy the files of the desired build to:
  - a. **.\TFS\Models Database\Tests\Testing\\_builds\{Program Name}\{Program Name} Build**
2. Run the following batch file to run examples:
  - a. **.\TFS\Models Database\Tests\Testing\Run-{Program Name}-{Parallel | Series}-GenericBuild.bat**

Note that testing in this way is not compatible with storing results in the archive using [ResultsArchiver](#)<sup>2</sup>.

### 2.5.2 Specific Builds

With a bit more work you can test with specific builds such that results are easier to track and can be archived using [ResultsArchiver](#)<sup>3</sup>. The procedure is as follows:

1. Copy the desired build folder from the network to:
  - a. **.\TFS\Models Database\Tests\Testing\\_builds\{Program Name}**
2. For all tests to be run, update the XML elements in the regTest-{Program Name}.xml files for the latest build location, version, and build numbers. A build find/replace action in an editor such as Notepad++ makes this more reasonable.
  - a. Go to: **.\TFS\Models Database\Tests\Testing\{Single | PSB | Examples Development}\[{PSB Suite Name}]\regTest-{Suite Name}\regTest**
  - b. Open **regTest-{Program Name}.xml**
  - c. Update **regtest/testing/program/version** (~line 137)
  - d. Update **regtest/testing/program/build** (~line 138)
  - e. Update **regtest/testing/program/path** (~line 139)
3. Run the following batch file to run the examples:
  1. **.\TFS\Models Database\Tests\Testing\Run-{Program Name}-{Parallel | Series}.bat**

### 2.5.3 Example with Generic Builds



#### Example with Generic Builds

The simplest way to test a build is to just do the following steps (for programs other than ETABS, just change the name accordingly):

1. Copy the files of the desired build to:
  - a. **.\TFS\Models Database\Tests\Testing\\_builds\ETABS\ETABS Build**

<sup>2</sup> <http://localhost:8090/display/CSIT/ResultsArchiver>

<sup>3</sup> <http://localhost:8090/display/CSIT/ResultsArchiver>

2. Run the following batch file to run both the analysis and design verification examples concurrently:
  - a. **.\TFS\Models Database\\_Tests\Testing\Run-ETABS-Parallel-GenericBuild.bat**

## 2.6 Viewing Test Results

To review the results is to just do the following steps:

1. Double click the following shortcuts, located at the same place as where you initiated the test run:
  - a. **.\TFS\Models Database\\_Tests\Testing\Results-{Program Name}-{Suite Name}**
2. Or go to here:
  - a. **.\TFS\Models Database\\_Tests\Results\{Program Name}\{Suite Name}\{Single | PSB | Examples Development}**
3. Enter the most recently created directory (date/time are in the name), and once the test run is far enough along, you can find and open the following HTML file to view the run status and results:
  - a. **test\_results\_{YYYY-MM-DD(seconds) Program Name & Build} run\_as\_is.html**

### 2.6.1 Example with ETABS Analysis and Design Suites

#### ✓ Example with ETABS Analysis and Design Suites

The simplest way to review the results is to just do the following steps after a test run has been started:

1. Double click the following shortcuts, located at the same place as where you initiated the test run:
  - a. **.\TFS\Models Database\\_Tests\Testing\Results-ETABS-Analysis**
  - b. **.\TFS\Models Database\\_Tests\Testing\Results-ETABS-Design**
2. Or go to here:
  - a. **.\TFS\Models Database\\_Tests\Results\ETABS\Analysis\Single**
  - b. **.\TFS\Models Database\\_Tests\Results\ETABS\Design\Single**
3. Enter the most recently created directory (date/time are in the name), and once the test run is far enough along, you can find and open the following HTML file to view the run status and results:
  - a. **test\_results\_{YYYY-MM-DD(seconds) Program Name & Build} run\_as\_is.html**

## 2.7 Archiving Test Results

Test results may be archived for future storage in source control or on the network. See [Source Control Testing Directory Structure](#) (see page 4) for the preferred naming convention and directory organization. This may be automated using [ResultsArchiver](#)<sup>4</sup>.

<sup>4</sup> <http://localhost:8090/display/CSIT/ResultsArchiver>