

VectorCAST ATG Baseline Option

Version 1.0

2024-05-06

Application Note AN-ACT-1-016

Author	Jeffrey Fortin
Restrictions	Public Document
Abstract	Using the VectorCAST ATG Baseline Option

Table of Contents

1	Overview	2
2	Use Cases	2
	2.1 Basic Use Case.....	2
	2.2 Advanced Features	4
3	Additional Resources	5
4	Trademarks	5
5	Contacts	5

1 Overview

With the release of VectorCAST 2023 SP5, the VectorCAST ATG (Automatic Test case Generation) feature now supports a baselining functionality, which allows ATG to precisely encapsulate the behavior of existing code by capturing expected values from test case execution. This is achieved by integrating ATG with the ability of VectorCAST to specify and record certain values during test runs (i.e., using TEST.ATTRIBUTE lines). These recorded values are then automatically transformed into expected values through test execution and recorded as TEST.EXPECTED lines within the generated test-cases.

2 Use Cases

Baselining Process

ATG's baselining process is as follows:

1. **Backup and Remove Existing Tests:** Before performing baselining on your environment, ATG backs-up and then removes any tests currently imported.
2. **Generate and Import ATG Test Cases:** New test-cases are then generated using ATG and imported into the environment.
3. **Execute Test Cases:** The newly generated test-cases are executed and have their values captured.
4. **Convert Values into Expected Values:** The captured values are transformed into expected values.
5. **Re-execute Test Cases:** To ensure consistency, the test-cases are executed again.

Repeat steps 3 to 5 if any values fail after re-execution (e.g., due to uninitialized test inputs, random values, "unstubable" functions, etc.). ATG's baselining process will automatically remove any failing expected values during these iterations.

2.1 Basic Use Case

To baseline an existing VectorCAST/C++ environment, you use ATG from the command line:

```
atg --baselining <output_file>
```

Replace <output_file> with the actual file name where the generated values and expected lines for the environment will be stored. If no file name is provided, the output defaults to `atg.tst`.

An example of this use case is shown in Figure 1.

```
>atg --baselining
Archiving existing tests
Processing unit: baselining_example
Subprogram: Object::main(float*,int*,char*,int)char* 1..... 2..
Baselining iteration 1 (generating expected values)
    Deleting current tests
    Importing tests from the previous iteration
    Executing tests
    Setting actuals to expecteds
    Exporting tests for the next iteration
Baselining iteration 2 (verifying generated expected values)
    Deleting current tests
    Importing tests from the previous iteration
    Executing tests
    Exporting tests for the next iteration
    Total mismatched expecteds: 3
Baselining iteration 3 (test differences found; re-verifying)
    Deleting current tests
    Importing tests from the previous iteration
    Executing tests
    Exporting tests for the next iteration
    Tests are the same between iterations; baselining complete
Exporting baselining tests
Restoring archived tests
Baselining completed

>clicast -lc -e ATG_BASELINING TEST Script run atg.tst
```

Figure 1 – ATG Baseline Example

Figure 2 shows the tests and test report for an example test environment with tests generated using the atg --baselining command.

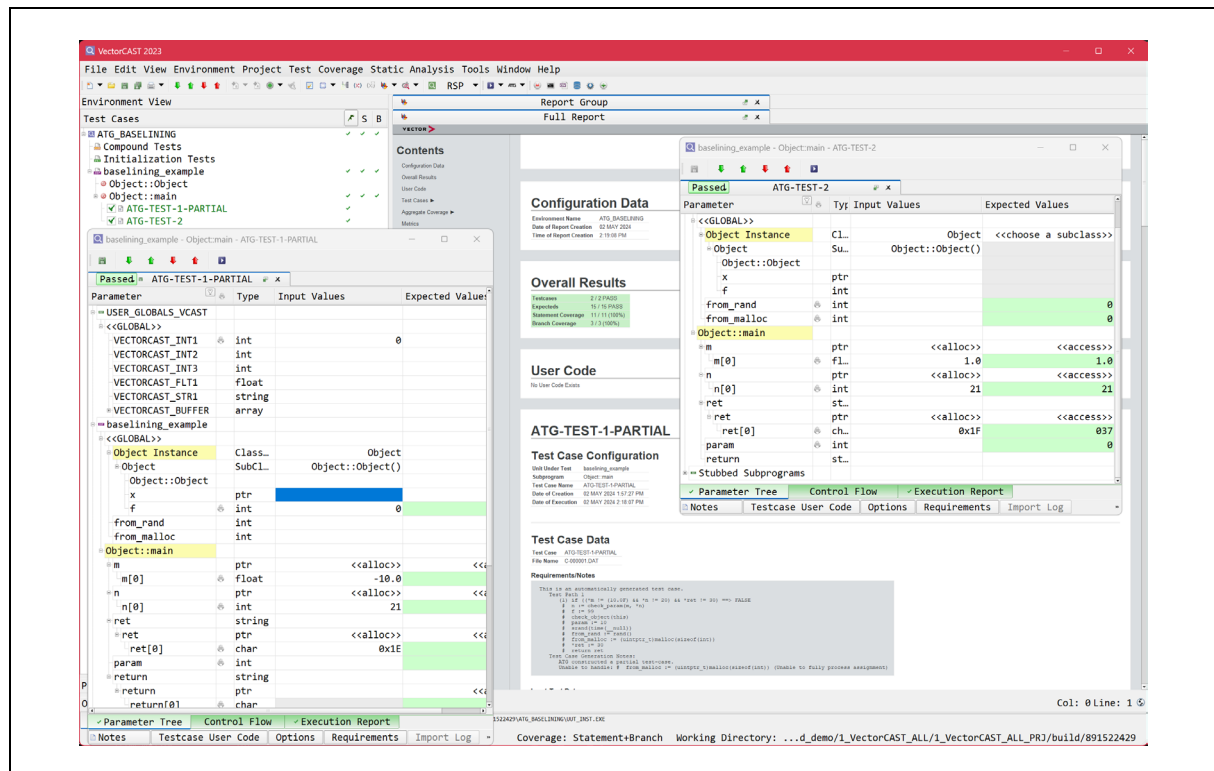


Figure 2 – Example Test Results

2.2 Advanced Features

Several other command line options are available for enhancing the baselining functionality:

- `--baseline-on-tst <path to test script>`: Use this option to baseline using a pre-existing file, without generating additional value lines.
- `--baselining-allow-atg-tests-in-env`: This checks if any ATG tests are loaded in the current environment before proceeding with baselining.
- `--no-baselining-delete-tests`: Select this option to retain currently imported tests when generating expected values.

3 Additional Resources

Some additional resources you may find helpful:

> [VectorCAST Product Page](#)

4 Trademarks

All mentioned names are either registered or unregistered trademarks of their respective owners.

5 Contacts

For a full list with all Vector locations and addresses worldwide, please visit <http://vector.com/contact/>.