



| PLACE_ORDER | PLACE_ORDER_001 | 6 FEB 2001 |
|-----------------|-----------------|-----------------|
| PLACE_ORDER_002 | 6 FEB 2001 | |
| PLACE_ORDER_003 | 6 FEB 2001 | |
| PLACE_ORDER_004 | 6 FEB 2001 | |
| Test Cases | 4 | |
| Complexity | 6 | Search Coverage |



Integrated with the Following Aonix Compilers:

- > AdaWorld™
- > ObjectAda™
- > ObjectAda/Raven™
- > ActivAda/Real-Time™

```
function Force_Upper_Case (S : String) return
string is
  Temp : String(1..Str'Length) := Str;
begin
  for I in 1..Str'Length loop
    -- if this is a lower case letter
    if (Temp(I) in 'a'..'z') then
      -- force it to upper case
      Temp (I) := Character'Pos(Temp(I)) - 32;
    end if;
  end loop;
  -- return converted value
  return Temp;
end Force_Upper_Case;
```

Highlights

- > Supports Ada83 and Ada95
- > Eliminates Need to Build Test Drivers and Stubs
- > Integrated Code Coverage Capabilities including MC/DC
- > Supports Host, Simulator, or Embedded Target Testing
- > Automates Regression Testing

< What is VectorCAST/Ada ? >

VectorCAST/Ada is a world-class integrated software test solution that significantly reduces the time, effort, and cost associated with testing Ada software components necessary for validating safety and mission critical embedded systems.

Automation includes:

- > complete test harness construction
- > test case definition
- > test execution from GUI or scripts
- > code coverage analysis
- > regression testing
- > code complexity calculation
- > computation of test paths for test case building

< Why VectorCAST/Ada ? >

With conservative estimates, software component testing requires the generation of one line of test code (in the form of stubs, drivers, and test data) for each line of application code to be tested. The necessity to create this "disposable" test software is the main reason why manual component testing is so expensive and inefficient. Test software not only has to be written but debugged to ensure that it performs as expected. With VectorCAST/Ada, component testing can be performed without writing a single line of test code.

< Simple To Use GUI >

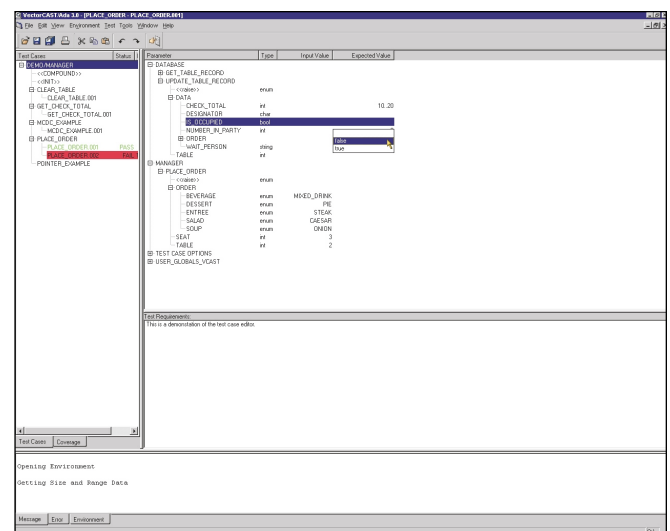
Test Case Building without writing test code. Parameters (of unit under test and stubs) and global data values are defined through the GUI.

Test Execution is performed with a simple mouse click.

Pass/Fail results of test cases are displayed on the GUI after test execution with color coded pass/fail indicators.

Code Coverage is shown in a color coded browser. Coverage levels are displayed for statement, branch, and MC/DC levels of coverage.

Execution can be performed on host platform, target instruction set simulator, or directly on embedded target. Execution platform is controlled from the GUI.



< How it Works >

VectorCAST/Ada parses your source code and invokes code generators to automatically create the test code required to construct a complete executable test harness. Once the test harness is constructed, utilities are provided to build and execute test cases, show code covered, and report static measurements. As you use VectorCAST/Ada, all data input during an interactive session is captured for future automated regression testing.

< Integrated Code Coverage >

Without a code coverage tool it is difficult to determine which portions of the source code have been exercised during testing. VectorCAST/Ada provides an integrated code coverage utility that allows you to gauge the effectiveness of your component testing by reporting on the source code statements or decision points exercised during individual or multiple test runs.

< Testing is Repeatable >

Once test cases have been developed, VectorCAST/Ada can be used to automatically run these test cases against successive versions of the software as the code changes. The management of test execution and cataloguing of test results is automatic. Comparing results of the same test cases against new software versions, prior to system integration, will result in fewer surprises caused by "one small change" to a software component.

< Compiler Integration >

VectorCAST/Ada is integrated with leading Ada compilers allowing for seamless test activities. All of the VectorCAST/Ada generated test harness components are automatically compiled and linked using your compiler. An interface to your compiler's debugger is also provided so you can run test cases under control of the debugger.

< Embedded Target Testing >

A version of VectorCAST/Ada is available to allow testing directly on your embedded target development system. *VectorCAST/Target* is integrated with your cross compiler and RTOS making it the perfect tool for testing real-time Ada applications. Tests may be developed in a host environment and then re-executed on an embedded target to verify the target and cross-compiler performance.

Product Features

- > Access data and subprograms defined in the package body
- > Test protected types and task types
- > Test exception raising and handling
- > Test generics
- > Test nested packages
- > Flexible test harness creation: stubs can be created anywhere in calling hierarchy
- > Graphical and script-based test case editing
- > Test all data types:
 - access types
 - tagged types
 - variant records
 - unconstrained arrays
 - system.address types

```
with DATABASE;
with TYPES, use TYPES;

package body MANAGER is

  procedure PLACE_ORDER (
    TABLE : in TYPES.TABLE_INDEX_TYPE,
    SEAT : in TYPES.SEAT_INDEX_TYPE,
    ORDER : in TYPES.ORDER_TYPE) is
    TABLE_DATA : TYPES.TABLE_DATA_TYPE;

  begin
    11 * DATABASE.GET_TABLE_RECORD (
      TABLE => TABLE,
      DATA => TABLE_DATA);
    12 *
    13 * TABLE_DATA.IS_OCCUPIED := true;
    14 * TABLE_DATA.NUMBER_IN_PARTY := TABLE_DATA.NUMBER_IN_PARTY + 1;
    15 * TABLE_DATA.ORDER (SEAT) := ORDER;
    16 *
    17 * case ORDER.ENTREE is
    18 * when TYPES.NO_ORDER =>
    19 *   when TYPES.STEAK =>
    20 *     TABLE_DATA.CHECK_TOTAL := TABLE_DATA.CHECK_TOTAL + 14;
    21 *   when TYPES.CHICKEN =>
    22 *     TABLE_DATA.CHECK_TOTAL := TABLE_DATA.CHECK_TOTAL + 10;
    23 *   when TYPES.LOBSTER =>
    24 *     TABLE_DATA.CHECK_TOTAL := TABLE_DATA.CHECK_TOTAL + 18;
    25 *   when TYPES.PASTA =>
    26 *     TABLE_DATA.CHECK_TOTAL := TABLE_DATA.CHECK_TOTAL + 12;
    27 * end case;
    28 *
    29 * DATABASE.UPDATE_TABLE_RECORD (
      TABLE => TABLE,
      DATA => TABLE_DATA);
  end PLACE_ORDER;
```

Aonix

5040 Shoreham Place, San Diego, CA 92122

tel: 858.457.2700 www.aonix.com

sales offices: www.aonix.com/content/company/sales.html

VECTOR
software

www.vectorcast.com