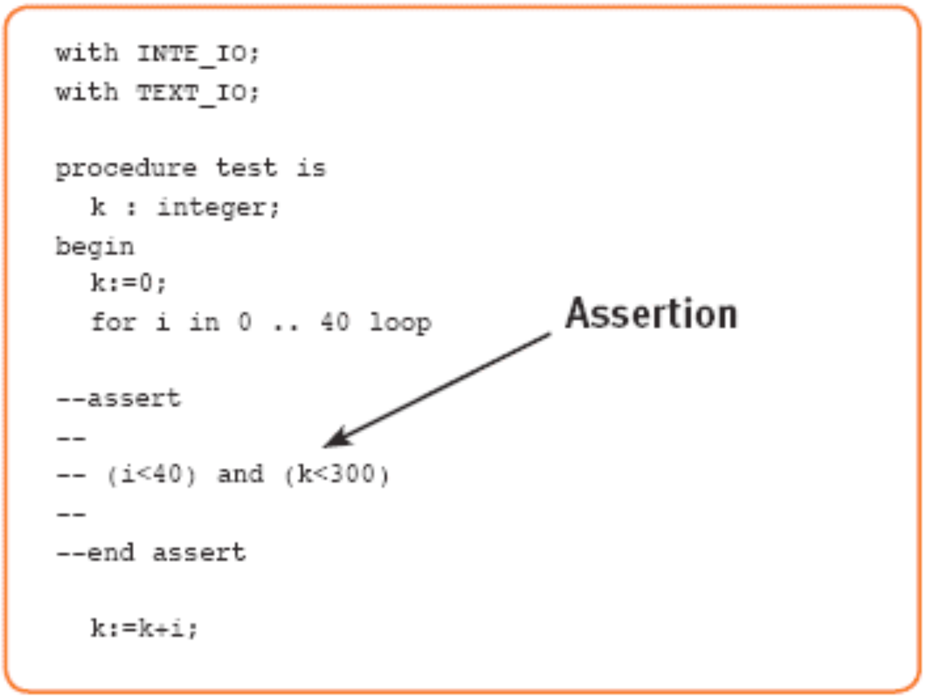
TBsafe® - Exact Semantic Analysis

LDRA Testbed is able to check assertions for conformance at run-time. This involves comparing variable values with a particular Boolean condition. The assertions are written as special comments in the source code in the style of many formal specification notations.

LDRA Testbed scans a source file for particular strings. The exact string and some of the matching rules are described in a parameter file. Each assertion must start and finish with one of these special strings. Every line of an assertion (including the start and finish special strings) must be part of a normal comment in the source language. This means that if the original source was submitted to the compiler it would compile as normal with the assertions treated as comments. When the start of an assertion string is detected the subsequent lines (up to the finishing special comment) are treated as an assertion expression. The expression is parsed to check that it is a valid Boolean expression. The explicit syntax can be customised, the standard form corresponding to that of languages such as Ada, C and Pascal.



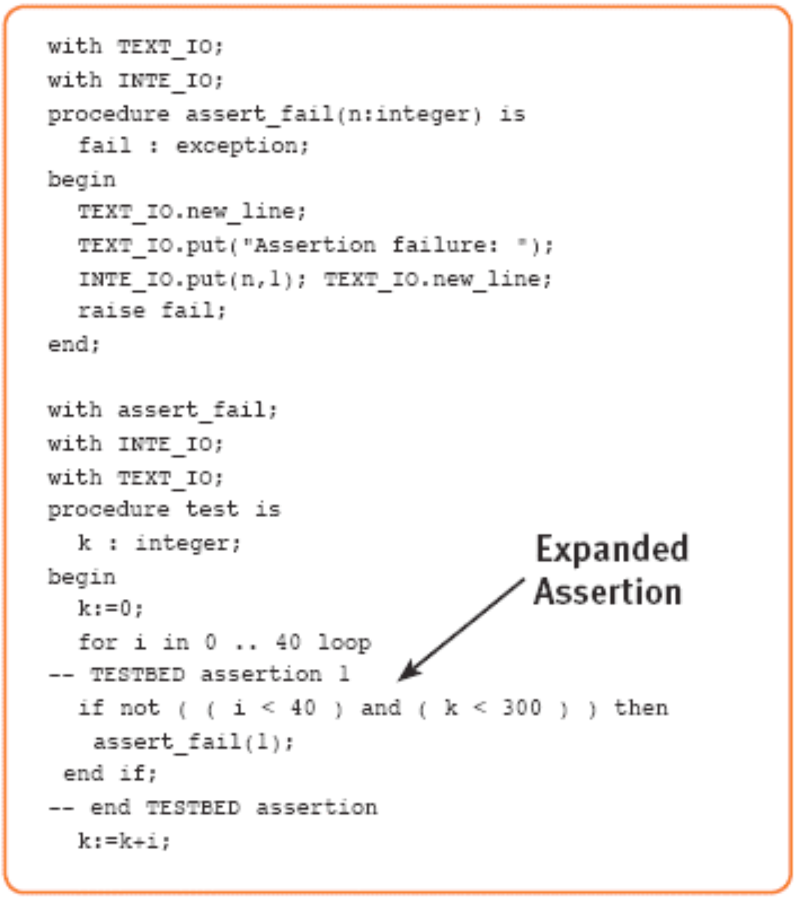
```
with INTE_IO;
with TEXT_IO;

procedure test is
  k : integer;
begin
  k:=0;
  for i in 0 .. 40 loop

--assert
--
-- (i<40) and (k<300)
--
--end assert

  k:=k+i;
```

After Dynamic Conformance Analysis the Assertion (above) becomes expanded as demonstrated below:



```
with TEXT_IO;
with INTE_IO;
procedure assert_fail(n:integer) is
  fail : exception;
begin
  TEXT_IO.new_line;
  TEXT_IO.put("Assertion failure: ");
  INTE_IO.put(n,1); TEXT_IO.new_line;
  raise fail;
end;

with assert_fail;
with INTE_IO;
with TEXT_IO;
procedure test is
  k : integer;
begin
  k:=0;
  for i in 0 .. 40 loop
-- TESTBED assertion 1
  if not ( ( i < 40 ) and ( k < 300 ) ) then
   assert_fail(1);
 end if;
-- end TESTBED assertion
  k:=k+i;
```

Instrumentation is added before and after the expression to sense its truth value. The instrumentation is such that an untrue assertion will cause a "failure" routine to be called which passes a reference number indicating this particular assertion. The assertions will typically have been derived from requirements and specification documentation. They can be used to specify pre- or post-conditions applying to a section of code. LDRA Testbed produces instrumented source (with assertions switched on) and a "failure" handling routine, which is customisable. The diagrams [right] demonstrate the effect of the source code when Exact Semantic Analysis has transformed assertions.

Exact Semantic Analysis is a combination of formal methods and pragmatic testing, checking assertions at run-time for compliance.

Exact Semantic Analysis Summary

- Achieve Exact Semantic Analysis of source code
- Produce diagnostics for the source code
- Specify pre- or post-conditions applying to a section of source code
- Check that inputs satisfy predetermined ranges
- Check that loop and array indices are within bounds

Obtaining Further Information

For further information on this particular feature of TBsafe and its availability please complete:
the LDRA reply form or email info@ldra.com.

Certificate Number FM 26376

www.ldra.com

LDRA UK & Worldwide
Portside, Monks Ferry,
Wirral, CH41 5LH
Tel: +44 (0)151 649 9300
e-mail: info@ldra.com

LDRA Technology Inc.
Lake Amir Office Park, 1250 Bayhill Drive Suite # 360
San Bruno CA 94066 Tel: (650) 583 8880
e-mail: info@ldra.com

LDRA Technology Pvt. Ltd
#2989/1B, 3rd Floor, 12th Main, 80 Feet Road,
HAL II Stage, Bangalore- 560008. Near BSNL Building
Tel: +91 80 4080 8707
e-mail: india@ldra.com