New VFCS testing tool XVRTE

Mária Lucká*　　Martin Paul †

Overview

This document describes the method and manner of testing of the Vienna Fortran Compilation System (VFCS) [2] by means of the a testing tool "XVRTE". XVRTE uses regression method as method of testing and is supported by an X11-MOTIF based user interface, which facilitates testing of new versions of VFCS. The XVRTE testing environment contains two test suites with 36 test cases at present, and in comparison with the previous testing tools has the following advantages:

- supported by the X11-MOTIF based user interface
- easy handled
- contains new test suite *vfsuite*
- allows the choice of test programs for the actual test run
- programs in *vfsuite* are executable on hypercube iPSC/860

1 VFCS Regression Test Environment XVRTE

The XVRTE is a new VFCS test environment that provides the VFCS developers the possibility of automatic testing of the new versions of VFCS with an arbitrarily selected subset of test cases. The process of testing is supported by a hierarchically structured menu system based on the X11-MOTIF user interface. XVRTE has two important components:

- the XVRTE user interface

*Institute for Control Theory and Robotics, Slovak Academy of Sciences, Dúbravská cesta 9, 842 37 Bratislava Slovakia ,*utrrluck@savba.savba.sk*, *lucka@.par.univie.ac.at*

†Institute for Software Technology and Parallel Systems, University of Vienna, A-1090 Vienna, Liechtensteinstrasse 22, *martin@par.univie.ac.at*

- a set of test suites

The XVRTE testing environment is based on the *regression method* of testing. *Regression testing* is done by compiling and running a selected set of the test programs, followed by comparing the intermediate output files with the reference output files. If the modifications have been made correctly, the file comparison shows differences only where they are expected.
For the purposes of testing VFCS a new suite of test programs *vfsuite*, was created. Because the XVRTE testing tool supports the testing of VFCS in batch mode a special batch file has been created for every test program. XVRTE does not run the selected subset of tests automatically - the test programs are only compiled. The user has a possibility to copy the selected program into his own directory and to run it there. The reference output files from execution on the iPSC/860 are placed in the same directory as the test program.
We now describe the more precisely the structure of test suite *vfdir*.

2 Test suite

The XVRTE testing environment contains a set of test suites. The test suite *testsuite* is identical with the test suite involved in *vrte* [1].
The *vfsuite* is a new one and has the structure is described bellow. The programs there essentially belong to one of the three groups:

- synthetic programs - **syn**
- kernel programs - **ker**
- application programs - **app**

The current test suite *vfsuite* contains no application code, all present codes belong to the former two groups, likewise they have different names.
For example, the programs with the name " dist " are constructed to exercise the distribution of arrays.
Every test case "xx.vf" has the separate subdirectory with the same name "xx". This directory contains the following:

1. xx.vf - source text of a test case
2. xx.desc - description file.
   This file contains:

   DESCRIPTION=

   AUTHOR=

   BATCHFILE=

   FORTRANFILES=

OTHERINPUT=

OUTPUTFILES=

For example, the description file can look like:

```
DESCRIPTION=matrix multiplication
AUTHOR=adapted program from /global/superb/x_system/vftest/
testsrc/f12.f
BATCHFILE=ker_3.bat
FORTRANFILE=ker_3.vf
OTHERINPUT=ker_3.in
OUTPUTFILES=host.f init.nsb init.sb nodesys.f hostsys.f
init.ob init.tov init.nb init.ovp init.ws node.f ker_3.out
```

3. xx_77 - directory; it contains program xx_77.f (written in Fortran 77) and the execution output file.

4. xx.bat - batch file This file is written in the batch language and in general has the form:

```
delete_old_units
scalar_fwd
start_par
opt_comm
opt_mask
compute_overlap
set_byte_swap on
set_svm off
set_comm_prot off
set_lin_pointers off
set_target_arch ipsc
set_comm_lib nx
gen_code
save_final_code
```

   This file also contains the transformations to be applied to a given program.
   Remark.
   Some transformations are valid only in interactive mode and inefficient when used in batch mode.

5. REF - directory with reference output files for the computer iPSC/860 produced by the batch mode compilation. This directory contains the files: host.f, hostsys.f, node.f, nodesys.f, init.nb,init.nsb, init.ob, init.ovp, init.sb, init.tov, init.ws, xx.out. The file xx.out is the reference output file from the execution of the program xx.vf on hypercube iPSC/860.

6. REFI - directory with reference output files gained from the interactive mode of compilation and with the corresponding reference output files.

7. xx.rem - file with a remark concerning this test case.
   For example:

```
dist_1 : assignment of distributed and nondistributed arrays
Program has correct results after compilation in the
interactive mode.
During run after batch mode compilation  -
the following error message arises:

Fortran runtime error on external file "init.nb" (-1):
     End of file reached

The size of init.nb in the interactive mode is 1201;
the size of init.nb in the batch mode is 196

The program uses no special transformations.
There are also some differences in any constants in file node.f
generated interactively and in the batch mode.
```

8. xx.in - file with input data (if any)

3 How to use XVRTE?

The testing tool XVRTE can be obtained by sending e-mail to martin@par.univie.ac.at. You will obtain the distribution file 'xvrte.tar.gz', which must be uncompressed and untarred. This creates a subdirectory 'xvrte', containing all the relevant files. The README file there gives more for information on installing the tool. The test tool interface can be then called by typing the command 'xvrte'. The testing tool interface consists actually of three different boxes. The first of them - after 'XVRTE' call - is illustrated in the Figure 1.
This shows the user a set of test suite directories and he can choose one of them; the test programs involved there can be seen in Figure 2. The left hand side box on Figure 2 shows the names of directories containing the Vienna Fortran source program and all programs and subdirectories described above. Clicking on the name of program followed by clicking on the box " Add " selects the program and its name is moved into the testing directory. The right hand side box on the same figure shows the names of all programs that will take part in the actual test run.
The run over the all test programs need clicking on the button " Add all " as well. Removing a test program or all programs by means of buttons " Remove" or "Remove all".

The version of VFCS to be used for compilation of the selected set of test programs must be entered in the box "VFCS command". If this box is empty, the global version of VFCS is used. Another version of the compiler can be determined also by either entering the correct command line in this box, or by setting the environment variable VRTEVFCS like this

setenv VRTEVFCS "/global/vfcs/src/vfcs -ALL /global/vfcs/src "

The testing process is started by clicking on the button "Run". After the test process has been started, the box on the Figure 3 shows the current state of the testing process. The last output of this window is the information about the overall course of testing process. This information has the form:

```
Testsuite completed -- Wed May 31 18:28:12 MET DST 1994
Result:
Tests run: 21
Tests passed: 15
Tests failed: 6
```

The course of testing is written also in the file *vrte.log* in the user local testing directory. All differences found during the testing process are saved into this file. All Vienna Fortran codes placed in the *vfsuite*(25 programs) were checked on the hypercube iPSC/860. The execution reference output files for every test code are in the subdirectories REF and REFI.

4 Present restrictions and possible improvements

The new testing tool "XVRTE" has the following restrictions:

- the only possible target architecture is iPSC/860
- the execution of compiled programs is not done automatically
- XVRTE does not allow hierarchical view of test suite subdirectories and files

Remark.
If it is a necessary to view or to run a test program, it can be copied from the suite directory to the user directory.

5 List of involved programs

Here is the a list with short descriptions of the single test cases taken from the *vfsuite*.

```
 dist_1
DESCRIPTION= assignment of some elements of distributed array to
           a nondistributed array
AUTHOR=adapted from /global/superb/x_system/vftest/testsrc/
f10.f
****************************************************************
 dist_2
DESCRIPTION= assignment of some elements of distributed array to
           a nondistributed array
AUTHOR=adapted from /global/superb/x_system/vftest/testsrc/
f101.f
****************************************************************
 dist_3
DESCRIPTION= assignment of some elements of distributed array to
           a nondistributed array
AUTHOR=adapted from /global/superb/x_system/vftest/testsrc/
f101.f
****************************************************************
 dist_4
DESCRIPTION= handling with distributed arrays
AUTHOR=adapted from /global/superb/x_system/vftest/testsrc/
f35.f
****************************************************************
 gauss
DESCRIPTION=Gauss elimination
AUTHOR=program f14.f from/global/superb/x_system/vftest/testsrc
****************************************************************
 ind_var_sub_1
DESCRIPTION=Induction variable substitution
AUTHOR=Robert Mayer, according to program ind204 from
regression test suite
****************************************************************
  ker_1
DESCRIPTION= assignment of some elements of distributed array
            to a nondistributed array
AUTHOR=T.Haupt-adapted from hpf_test_suite/sy02.hpf by M.Lucka;
****************************************************************
 ker_2
DESCRIPTION=four-color-scheme
AUTHOR=adapted from red-black: program from /global/superb/
x_system/vftest/testsrc/f11.f
****************************************************************
 ker_3
DESCRIPTION=matrix multiplication
```

```
AUTHOR=adapted program from /global/superb/x_system/vftest/
testsrc/f12.f
****************************************************************
 ker_4
DESCRIPTION=matrix multiplication
AUTHOR=adapted program from /global/superb/x_system/vftest/
testsrc/f29.f
****************************************************************
 ker_5
DESCRIPTION=integration with the rectangular method
AUTHOR=adapted program from /global/superb/x_system/vftest/
testsrc/f71.f
****************************************************************
 ker_6
DESCRIPTION=kernel from Jacobi
AUTHOR=adapted program from /global/superb/x_system/vftest/
testsrc/f34.f
****************************************************************
 sin_trans
DESCRIPTION=one-dimensional fast sine transform
AUTHOR=Maria Lucka, lucka@par.univie.ac.at
****************************************************************
 syn_1
DESCRIPTION=writing of three values of one array
AUTHOR=taken from  /global/superb/x_system/vftest/testsrc/f13.f
****************************************************************
 syn_2
DESCRIPTION=total overlap
AUTHOR=taken from  /global/superb/x_system/vftest/testsrc/f15.f
****************************************************************
 syn_3
DESCRIPTION= assignment of distribution array to replicated one
AUTHOR=taken from  /global/superb/x_system/vftest/testsrc/f28.f
****************************************************************
 syn_4
DESCRIPTION=mark reduction transformation example
AUTHOR= fragment from /global/superb/x_system/vftest/testsrc/
f71.f
****************************************************************
 syn_5
DESCRIPTION= array distribution
AUTHOR= from /global/superb/x_system/vftest/testsrc/f7.f
****************************************************************
 syn_6
```

```
DESCRIPTION=passing of distributed array as subprogram argument
AUTHOR=adapted from /global/superb/x_system/vftest/testsrc/f1.f
****************************************************************
 syn_6
DESCRIPTION=passing of distributed array as subprogram argument
AUTHOR=adapted from /global/superb/x_system/vftest/testsrc/f1.f
****************************************************************
 syn_7
DESCRIPTION= distribution of three-dimensional array
AUTHOR= from /global/superb/x_system/vftest/testsrc/f20.f
****************************************************************
 syn_8
DESCRIPTION= distribution of one-dimensional array
AUTHOR= from /global/superb/x_system/vftest/testsrc/f22.f
****************************************************************
 syn_9
DESCRIPTION=  passing of distributed and non-distributed
arrays
AUTHOR= from /global/superb/x_system/vftest/testsrc/f27.f
****************************************************************
 syn_10
DESCRIPTION= transformation reduction
AUTHOR=according T.Haupt prob4.hpf;sy20.hpf from hpf_test_suite
- M.Lucka
****************************************************************
 syn_11
DESCRIPTION= transformation reduction
AUTHOR=according T.Haupt prob06.hpf;sy21.hpf - hpf_test_suite
- M.Lucka
****************************************************************
 syn_12
DESCRIPTION=assignment of distributed and non-distributed array
AUTHOR=according T.Haupt array02.hpf;sy26.hpf -  hpf_test_suite
- M.Lucka
****************************************************************
```

References

[1] Paul M., VFCS Regression Test Guide Previous, Vienna, October 1994

[2] Vienna Fortran Compilation System, Version 1.0,User's Guide, Jannuary 1993

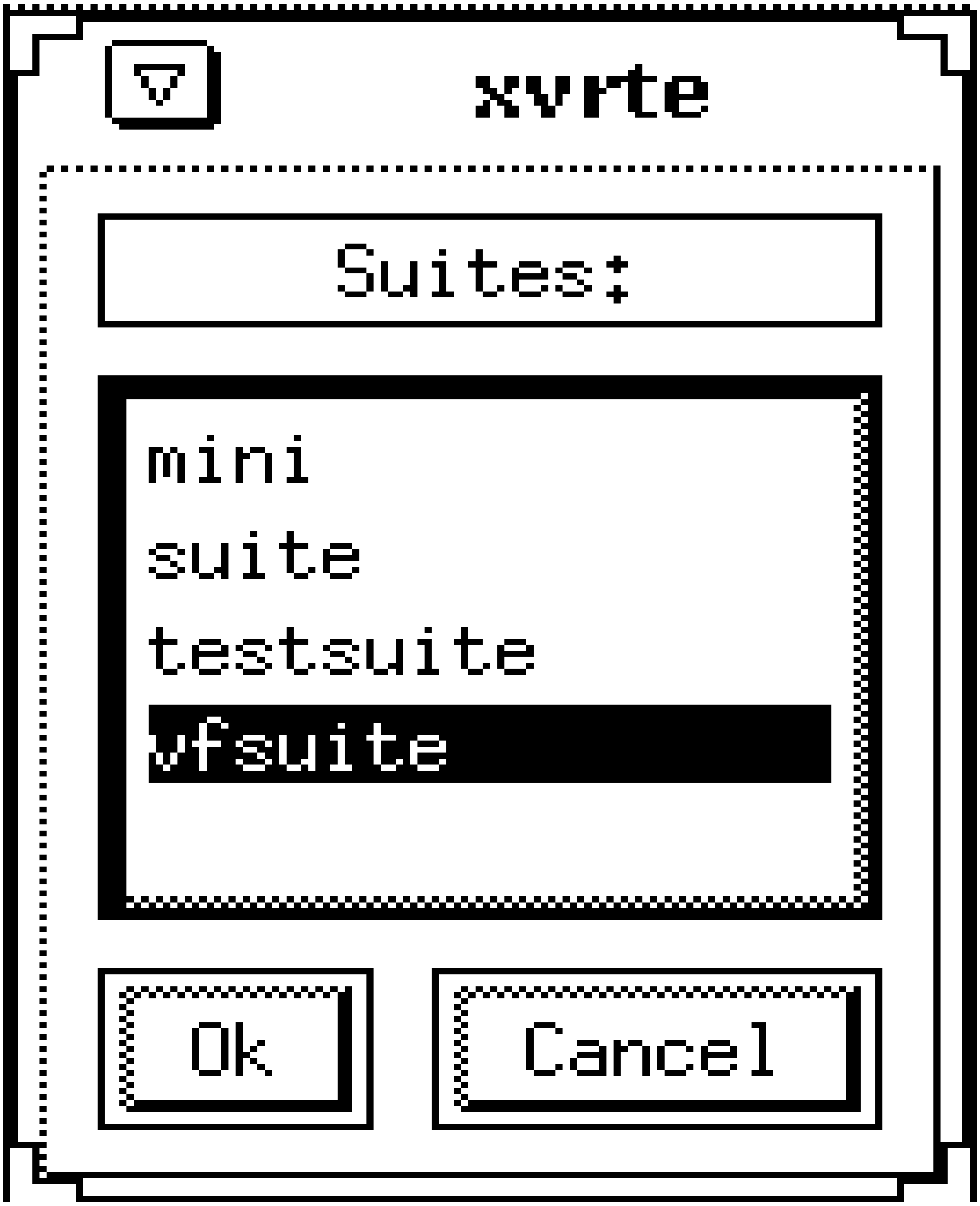

Figure 1: Starting test box

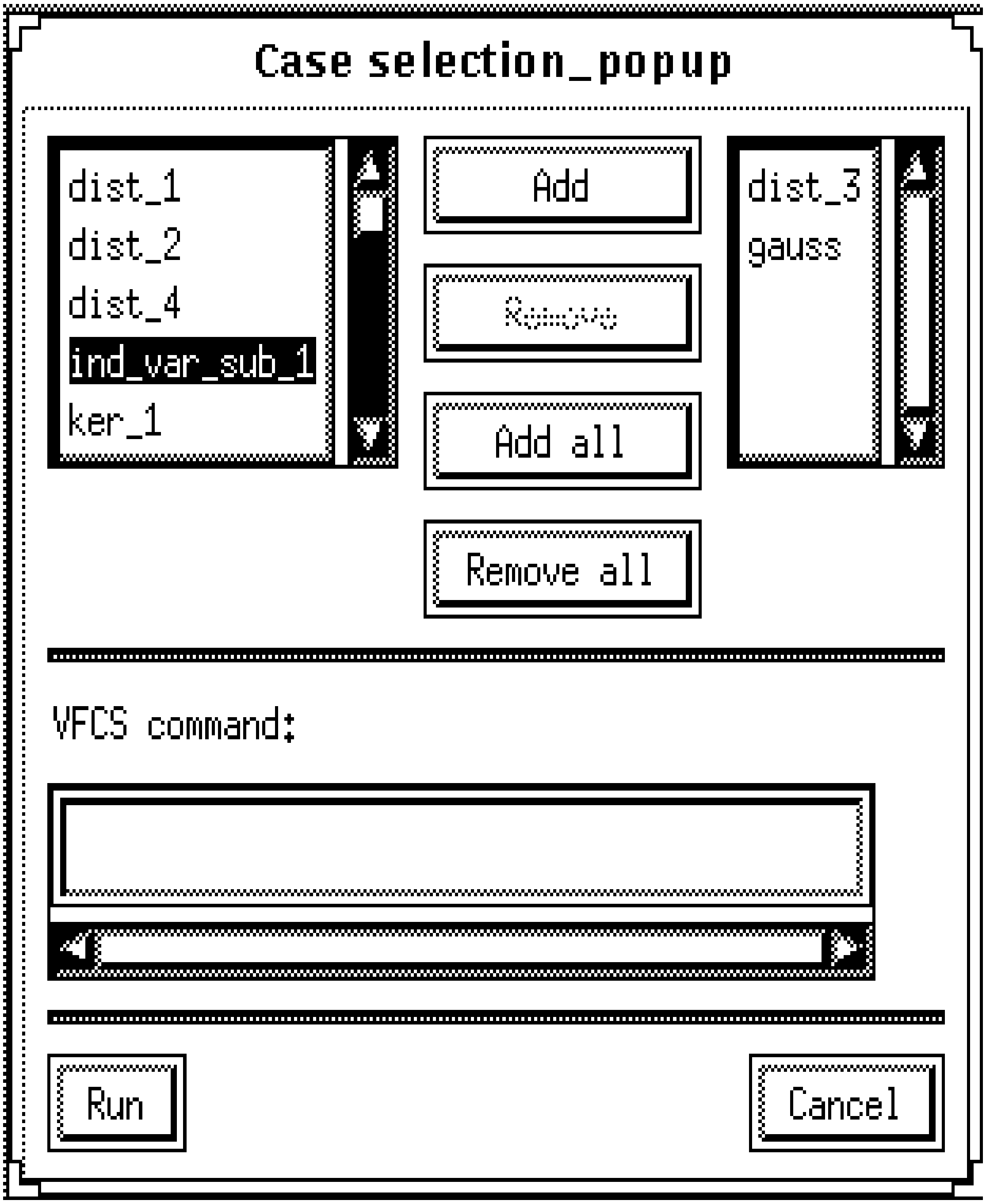

Figure 2: Case selection box

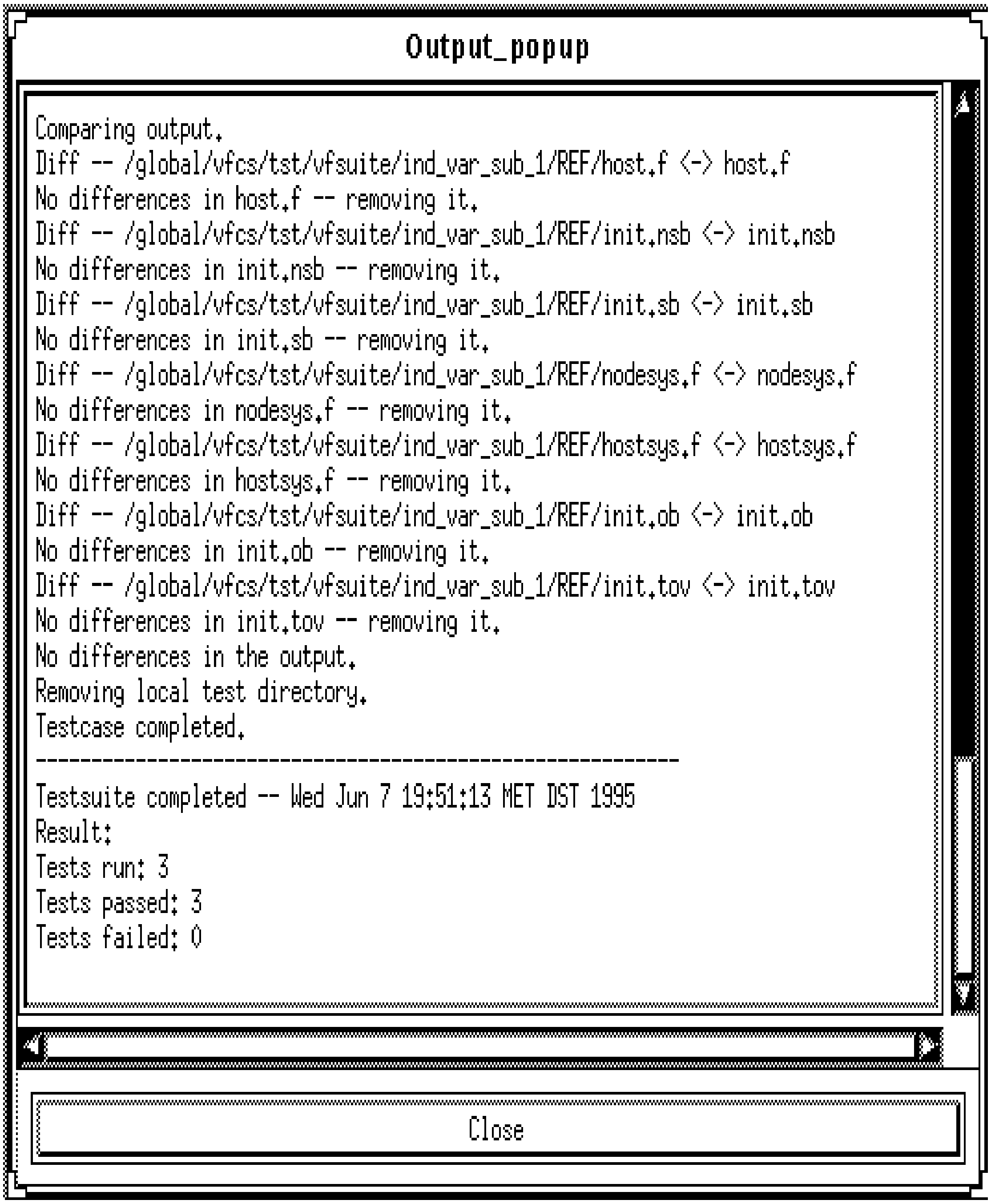

Figure 3: Case selection box