



FORCHECK

for

PC/Linux

A Fortran Verifier and Programming Aid

version 14

Installation Guide

July 2, 2010

The information in this document is subject to change without previous notice and should not be taken as a commitment by Forcheck b.v. Forcheck b.v. can not assume responsibility for any errors which may appear in this document.

The software described in this document is furnished under a license and may be used, copied or disclosed only when in accordance with the terms of this license.

Copyright ©Forcheck b.v. 1984 through 2010. All rights reserved.

FORCHECK has been developed by Erik W. Kruyt.

FORCHECK is currently available for PC/Windows, PC/Linux and HP-UX.

FORCHECK is a registered trademark of Forcheck b.v.

Absoft is a trademark of Absoft Corporation.

DEC, PDP, VAX, AXP, Alpha, RSX, VMS, OpenVMS, Ultrix and Tru64 UNIX are trademarks of Hewlett Packard Company.

DR Fortran-77 is a trademark of Digital Research, Inc.

FTN77 and FTN95 are trademarks of Salford Software Ltd.

FTN90 is a joint trademark of Salford Software Ltd and the Numerical Algorithms Group Ltd.

Hewlett-Packard, UX, Fortran/9000 are trademarks of Hewlett-Packard Company.

IBM, MVS, VS Fortran, Professional Fortran, RS/6000 and AIX are trademarks of International Business Machines Corporation.

Intel is a trademark of Intel Corporation.

Cray, Unicos, CF77 and CF90 are trademarks of Silicon Graphics, Inc.

Silicon Graphics, IRIX and MIPSpro are trademarks of Silicon Graphics, Inc.

Lahey, F77L, LF90 and LF95 are trademarks of Lahey Computer Systems, Inc.

Linux is a registered trademark of Linus Torvalds.

Microsoft, MS-DOS, MS-Fortran, Microsoft Fortran PowerStation, Windows 95, and Windows NT are trademarks of Microsoft Corporation.

MicroWay and NDP Fortran-386 are trademarks of MicroWay, Inc.

NAG and NagWare are trademarks of The Numerical Algorithms Group Limited.

Prospero Fortran and Pro Fortran-77 are trademarks of Prospero Software.

Ryan-McFarland and RM/Fortran are trademarks of Ryan-McFarland Corporation.

Sun and Solaris are trademarks of Sun Microsystems, Inc.

WATCOM is a trademark of Sybase, Inc.

All other trademarks and registered trademarks are the property of their respective holders.

Website: <http://www.forcheck.nl>

Email: info@forcheck.nl

Contents

1	Introduction	5
2	Distribution	7
3	Installation	9
3.1	Copying and retrieving the distribution files	9
3.2	Installation on a private account	9
3.3	System wide installation	10
3.4	Checkout	12
4	Environment	13
4.1	Operating system	13
4.2	Copyright message	13
4.3	Configuration file	13
4.4	Exit status	14
4.5	Help	14
4.6	User's guide	14
5	Customizing	15
5.1	Generation	15
5.2	Compiler emulation	15
5.3	Messages	16
5.4	Remarks	16
5.5	Summary of global variables	16

Chapter 1

Introduction

FORCHECK is a Fortran program development, conversion and maintenance tool. It parses Fortran programs, verifies the syntax and composes documentation. It analyzes both separate program units and the program as a whole. FORCHECK verifies the syntax by parsing the source program. This is done as precisely as possible at compile time. The full Fortran 2003 syntax (which includes the Fortran 95, Fortran 90 and FORTRAN 77 syntax) is supported. Moreover most language extensions of many compilers can be handled. As an option the syntax can be checked for strict conformance to the Fortran 2003, Fortran 95, Fortran 90, or FORTRAN 77 standard.

Cross-reference tables of all objects within program units are composed. Information and warnings concerning the usage of these objects are provided.

The reference structure (call tree) of the program can be analyzed and presented. Recursive references are traced.

The consistency of the entire program is verified by checking the type of the procedures and the argument lists of all procedure references. Length, data type and structure of the common blocks specified in the various subprograms are compared. Cross-reference tables of all procedures, common blocks, common-block objects, modules, external I/O and include files over the program are composed.

FORCHECK can emulate the language extensions of a specific compiler by reading a configuration file in which all data types and extensions to be supported are enumerated.

The global information of each program unit can be stored in library files which can be referenced and updated in subsequent FORCHECK runs to test program units in the context of the entire program. A librarian utility is supplied to maintain the FORCHECK library files.

Chapter 2

Distribution

The distribution kit contains the following files:

documentation files:

readme	last minute information
linux_i.pdf	The Installation Guide in PDF format
unix_u.pdf	The User's Guide in PDF format

production files:

fckinstall	script to install FORCHECK
forchk	task image
forchk.1	unformatted on-line documentation of FORCHECK
fcklib.1	unformatted on-line documentation of fcklib
fckerr.msg	direct access file with error messages
_fck_tree.xml	Style sheet for xml output
*.cnf	configuration files for compiler emulations
MPI.flb	interface library for MPI
fckdem.f	demonstration program
fckdem.inc	include file for demonstration program
fckdem.lst	output of analysis of demonstration program

maintenance files:

link	script to link forchk, fcklib and interf
*.obj	objects
fckerrsq.msg	sequential file with messages

utilities:

fcklib	FORCHECK library utility
interf	FORCHECK interface builder

maintenance utilities:

cnverm	convert sequential error message file to direct access file
--------	-------------------------------------------------------------

Chapter 3

Installation

3.1 Copying and retrieving the distribution files

Download the distribution kit from the ftp server and place the compressed tar file on the tmp directory.

Now you can uncompress and retrieve the files to an installation directory:

- Create a sub-directory, for example, forcheck:

```
mkdir forcheck
```

- Set the default directory:

```
cd forcheck
```

- Uncompress and retrieve the files:

```
tar -xzf /tmp/PCLinux_distr.tar.gz
```

The files in the distr1.tar archive are needed for maintenance purposes only. When needed they can be copied in the same way. They may not be supplied in an evaluation kit.

3.2 Installation on a private account

To use FORCHECK for one user only or during an evaluation period you may want to install FORCHECK not system-wide but only on a private account, e.g. ~/forcheck.

- Retrieve the password file from the email. Mind that the password file attached to the email is in DOS format and must be transformed to Unix format by replacing the cr/lf's by linefeed's only.
- FORCHECK tries to open the direct access error message file and password file using the environmental variable FCKDIR. Define an environmental variable with the pathname of these files. For the C-shell:

```
setenv FCKDIR ~/forcheck
```

or for the Bourne or Korn shell:

```
FCKDIR=~ /forcheck
export FCKDIR
```

3.3 System wide installation

- Make sure you have root privileges.
- Retrieve the password file from the email. Mind that the password file attached to the email is in DOS format and must be transformed to Unix format by replacing the cr/lf's by linefeed's only. Make sure this file is on the installation directory.
- Apply the installation script `fckinstall` to move the files to the correct path, set the protection codes and generate the man pages:

```
chmod +x fckinstall
./fckinstall
```

or, if you are not logged in with root privileges:

```
sudo ./fckinstall
```

This script will try to perform the actions listed hereafter, which can also be carried out manually:

- Place the executables `forchk` `fcklib` and `interf` on the directory `/usr/local/bin`:

```
cp forchk /usr/local/bin/
cp fcklib /usr/local/bin/
cp interf /usr/local/bin/
```

- Set the correct protection mode for these executables:

```
chmod 711 /usr/local/bin/forchk
chmod 711 /usr/local/bin/fcklib
chmod 711 /usr/local/bin/interf
```

- Create a subdirectory to store the ancillary files :

```
mkdir /usr/local/lib/forcheck
```

If you place these files in a different directory, you have to set the environmental variable `FCKDIR` to define this directory.

- Place the direct access `FORCHECK` error message file `fckerr.msg`, the password file `fckpwd.pwd` and the xml stylesheet on the directory `/usr/local/lib/forcheck`:

```
cp fckerr.msg /usr/local/lib/forcheck/
cp fckpwd.pwd /usr/local/lib/forcheck/
```

```
cp _fck_tree.xsl /usr/local/lib/forcheck/
```

- Set the correct protection mode for these files:

```
chmod 644 /usr/local/lib/forcheck/fckerr.msg  
chmod 644 /usr/local/lib/forcheck/fckpwd.pwd  
chmod 644 /usr/local/lib/forcheck/_fck_tree.xsl
```

- Make the configuration files *.cnf available to the users by copying them to /usr/local/lib/forcheck:

```
cp *.cnf /usr/local/lib/forcheck
```

- Set the correct protection mode for these files:

```
chmod 644 /usr/local/lib/forcheck/*.cnf
```

- Make the library interface files *.flb available to the users:

```
unzip *.zip  
cp *.flb /usr/local/lib/forcheck
```

- Set the correct protection mode for these files:

```
chmod 644 /usr/local/lib/forcheck/*.flb
```

- Copy the FORCHECK help files forchk.1 and fcklib.1 to the directory /usr/man/man1:

```
cp forchk.1 /usr/man/man1/forchk.1  
cp fcklib.1 /usr/man/man1/fcklib.1
```

- Set the correct protection mode for these files:

```
chmod 444 /usr/man/man1/forchk.1  
chmod 444 /usr/man/man1/fcklib.1
```

- Format the help files:

```
nroff -man /usr/man/man1/forchk.1 > /usr/man/cat1/forchk.1  
nroff -man /usr/man/man1/fcklib.1 > /usr/man/cat1/fcklib.1
```

- Update index- and keyword file:

```
makewhatis
```

- Check the manual pages:

```
man forchk  
man fcklib
```

3.4 Checkout

- Save the supplied list file:

```
mv fckdem.lst fckdem.ref
```

- Run forchk and analyze the demonstration program:

```
forchk -l fckdem.lst -f77 fckdem.f
```

- You can now compare the output file with the supplied output file:

```
diff fckdem.lst fckdem.ref
```

Because date and time will be different, the page headers will be shown as differences.

When `fcklib` compresses a library file, it creates a temporary file `.#fcklib.tmp`, which is deleted after successful compression. If, however, `fcklib` ends abnormally, the user will find this file on his current directory.

Chapter 4

Environment

4.1 Operating system

FORCHECK runs under the Linux operating system.

4.2 Copyright message

You can suppress the copyright message by setting the environmental variable FCKCPR to the keyword QUIET. For the C-shell:

```
setenv FCKCPR QUIET
```

or for the Bourne or Korn shell:

```
FCKCPR=QUIET  
export FCKCPR
```

4.3 Configuration file

To emulate a specific compiler and to tune the usage of FORCHECK the user can apply a different configuration file than the default. To use a specific configuration file the user must set the environmental variable FCKCNF with the pathname of the configuration file to be used. For example, to emulate the VAX Fortran compiler type, using the C-shell:

```
setenv FCKCNF /usr/local/lib/forcheck/vax.cnf
```

or for the Bourne or Korn shell:

```
FCKCNF=/usr/local/lib/forcheck/vax.cnf  
export FCKCNF
```

4.4 Exit status

When `forchk` exits, it generates a specified exit status which can be used in for example a script file.

exit status:

- 0 no informative, warning, overflow or error messages presented
- 2 informative, but no warning, overflow or error messages presented
- 4 warning, but no overflow or error messages presented
- 6 table overflow, but no error messages presented
- 8 error messages presented
- 16 fatal error occurred

4.5 Help

Man pages are supplied. Users can access the man pages through the `man` command.

4.6 User's guide

The user's guide is supplied as a pdf file.

Chapter 5

Customizing

5.1 Generation

Generation is normally not required: the executables are supplied. Some unix systems support dynamic load libraries which can be shared between tasks. This feature saves disk space and load time. To deliver FORCHECK independent of the version of the operating system we have linked the executables 'statically', with all run time code included. To take advantage of the dynamic load library feature of your operating system you have to relink using the script link.

5.2 Compiler emulation

FORCHECK can be used as a tool to convert Fortran programs from one system to another. FORCHECK can support most of the compiler syntax extensions of many compilers. During its initialization FORCHECK reads a configuration file in which relevant data for the emulation is stored. If the environmental variable FCKCNF has been specified, FORCHECK uses this path to read the configuration file. If FCKCNF has not been specified, FORCHECK reads the default configuration file:

```
/usr/local/lib/forcheck/g95.cnf
```

If a user wants FORCHECK to emulate a different compiler, he must set the environmental variable FCKCNF to the path of the configuration file to be used. For example for the C-shell:

```
setenv FCKCNF /usr/local/lib/forcheck/gfortran.cnf
```

or for the Bourne or Korn shell:

```
FCKCNF=/usr/local/lib/forcheck/gfortran.cnf
```

```
export FCKCNF
```

to emulate gnu fortran.

If a user wants to enable different Fortran extensions than the default the user must have his own copy of the appropriate configuration file.

5.3 Messages

We distinguish three kinds of messages, namely operational messages, analysis messages and system messages. All analysis and system messages are stored in a direct access file `fckerr.msg`. If you want to change analysis or system messages for all users, for example, change a message from 'E' (error) to 'W' (warning), you can edit `fckerrsq.msg` and convert this file to a direct access file `fckerr.msg` using the supplied utility `cnverm`.

If a user wants to adapt certain analysis messages to his needs, he must place the numbers and new level flags of these analysis messages in an option file which can be concatenated to the configuration file to be used (for the C-shell):

```
setenv FCKCNF /usr/local/lib/forcheck/sun.cnf;options_file
```

or for the Bourne or Korn shell:

```
FCKCNF=/usr/local/lib/forcheck/sun.cnf;options_file
export FCKCNF
```

In the option file the user can also specify the line or statement `count_mode` options of his choice. See the section on messages in the user's guide.

5.4 Remarks

- Input files and options are placed in a direct access scratch file `.#f1xxxx.tmp` (in which `xxxx` is a random number) , which will be deleted automatically on exit.
- The annotated source lines of each program unit are temporary stored in a sequential scratch file `.#f3xxxx.tmp` (in which `xxxx` is a random number) , which will be deleted automatically on exit.
- When `forchk` encounters an `INCLUDE` line, or an include preprocessor directive, it tries to open the include file specified. See the user guide for the search strategy of include files.
- A scratch library file `.#f2xxxx.tmp` (in which `xxxx` is a random number) is created, which be deleted automatically on exit.

5.5 Summary of global variables

<code>FCKCNF</code>	pathname of configuration file to be used
<code>FCKCPR</code>	suppression of copyright message
<code>FCKDIR</code>	directory of FORCHECK system files
<code>FCKPWD</code>	pathname of password file
<code>FCKOPT</code>	default options
<code>TMPDIR</code>	directory for scratch files