# RIGAL manual page

### NAME

**rc** - checks and compiles RIGAL file<br>
**ic** - interprets intermediate RIGAL code<br>
**v** - viewer for RIGAL data saved in file<br>
**rig_crf** - RIGAL cross -referencer<br>
**rig_lint** - RIGAL Lint (data type checker)

### SYNOPSIS

**rc** *filename* **[-c] [ -D] [ -P** parameter]...<br>
**ic** *codefilename* **[ -p** *printfilename*] **[ -d][** parameter...]<br>
**v** *filename*<br>
**rig_crf** *filename*<br>
**rig_lint** *filename executable* **[ -p** *printfilename*] [ parameter...]<br>
**(!)** the options and parameters must be separated by one or more spaces.

### COMPILATION TARGET (Pascal and C)

There are two variants of RIGAL implementation: RIGAL/Pascal in UNIX Pascal and RIGAL/C in UNIX C. The differences between them are mentioned below.

### DESCRIPTION OF rc

**rc** *filename* **[ -c] [ -D] [ -P** parameter]...<br>
**rc** - checks source file written in RIGAL language. The source code file must have suffix **.rig**, but in the

command *filename* is specified without the suffix. **rc** reports error messages if any appear; file name, row, column and message text are reported to standard output. **rc** creates intermediate code *filename***.rsc** which can be used for interpretation.

### OPTIONS

**-c**
creates an executable file *filename* (without suffix). In RIGAL/Pascal variant an intermediate Pascal code is generated and then Pascal compiler (**pc**) and linker are invoked. In RIGAL/C variant an intermediate C code is generated and then C compiler (**cc**) and linker are invoked. Some intermediate files, among them script file **xd**, also appear.
**-D** does not remove intermediate files after the compilation.
**-P***parameter*
passes option to Pascal or C compiler. You can use **-P** several times or use **-P** *'param1 param2'* syntax. Typically you may be need to compile with statically linked standard libraries, e.g. **$rig/rc rigcrf -c -P -Bstatic**

### DESCRIPTION OF *executable* file (compilation result)

*executable* **[ -p** *printfilename*] [parameter...]
The parameters are passed to RIGAL program and can be received through #PARM built -in rule which returns list of parameters. The execution is identical to the interpreted program.

### OPTION

**-p** *printfilename* - name of file for statement PRINT output. By default output is written to standard output.

### DESCRIPTION OF ic

**ic** *codefilename* [ **-p** *printfilename*] [ **-d**] [parameter...]
**ic** - interprets the intermediate code, created by **rc**. The intermediate code must have suffix **.rsc**, but in the command it is specified without the suffix. Other parameters are passed to RIGAL program and can be received through #PARM built -in rule which returns list of parameters.The execution is identical to the compiled program.

### OPTIONS

**-p***printfilename*
name of file for statement PRINT output. By default output is written to standard output.
**-d** tracing mode is turned on. Trace of rules and branches, rule arguments and success/unsuccess messages is written to file specified by **-p** option and is mixed with output from PRINT statements. If **-p** option is absent trace is written to standard output.

### DESCRIPTION of v

**v** *filename*
**v** writes the contents of RIGAL code saved in file to standard output. The structure of code is written with indentation, in well -readable form. If the depth of some element of the structure is more than 15 levels, then characters '..' appear instead of the element. The *filename* must contain data saved by RIGAL statement SAVE $*var* '*filename*'

### DESCRIPTION of rig_crf

**rig_crf** *filename*

**rig_crf** creates cross reference table for given RIGAL program *filename*. The program file must have suffix **.rig**, but in the command *filename* is specified without the suffix. The result is written to file *filename***.crf**. For every rule the following is reported: rules calling this one, rules called by this one, variables, use of global variables.

## DESCRIPTION of rig_lint

**rig_lint** *filename*
**rig_lint** creates warning table for given RIGAL program *filename*. The program file must have suffix **.rig**, but in the command *filename* is specified without the suffix. The result is written to file *filename***.lnt**. **rig_lint** checks type compatibility and cases when values of different types can be assigned to the same variable.

## ERROR MESSAGES

**Error in file** *filename* **Line=***nn* **Message=***text* **Column=***mm* - checker found syntax error in your RIGAL file. Intermediate code is not created, compilation failed.
***** ERROR** *nn xxx* - runtime warnings about dangerous manipulations with RIGAL data structures or runtime input/output errors.
rig:undefined variable
**pc:command not found** - see ENVIRONMENT section of this manual
ERROR:COMPILATION
**ERROR IN RIGAL, rule #***name* - internal compilation message, normally does not appear; try to simplify this rule.
**ld() errors or warnings** - try command **ranlib $rig/lib/riglib.a** to remove errors.

## ENVIRONMENT

For running RIGAL/Pascal you must have Sun Pascal release 2.1 (or higher). Command **pc** (Pascal compiler) must be available.
For running RIGAL/C you must have a C compiler (command **cc**) available. You must have environment variable **rig** set to the name of directory where RIGAL system binary files are located. You can set it by command **setenv rig** *full_path***/bin** in your **.login** file It is recommended to include this directory at the start of path list by command **set path=( $rig ... )** in your **.login** file. Then you can use RIGAL commands **rc**, **ic** etc. from any directory.

## INSTALLATION of RIGAL from sources

All files are packed and compressed in file **rigs***variant.system_version.***tar.Z** (e.g. rigsp.444.tar.Z). Here **rigsp** denotes Pascal sources, **rigsc** - C sources. **uncompress** file (without suffix **.Z**) **tar -xvf** file execute instructions in ENVIRONMENT section.
Go to RIGAL source directory (**/src**) and execute **make** ; it takes 3-5 minutes to compile all sources by Sun Pascal (or C) compiler and perform self-test. The result is in **/bin**

## FILES

**bin/rc, bin/ic, bin/rig_crf, bin/rig_lint, bin/v** basic RIGAL system executable files
**lib/riglib.a** RIGAL compiler runtime library. If you move it from one directory to another, execute command **ranlib** *new_directory***/riglib.a**
**include/***file***.inc**, **include/***file***.h** m include files for compilation
**bin/anrig, bin/genrig** parts of RIGAL compiler
**docs/rigal.1** this manual
**docs/RIGAL_language_description.html** RIGAL language description
**docs/RIGAL_programmers_guide.html** User's guide

*file***.rig** source programs in RIGAL
*file***.rsc** intermediate code for interpreter
**xcrg*.pas,** *file*.**RC2**, *file*.**RC4**, *file*.**TMP** intermediate files, appear and by default are removed during compilation.

## AUTHORS

Mikhail Auguston and Vadim Engelson, Inst. of Math. and Comp.Sci, University of Latvia, Rainis blvd. 29, Riga, LV-1459, Latvia (Lettland). Currently (1994-1995) Vadim Engelson is at IDA, S-58183, Linkoping University, Linkoping, Sweden. e -mail: vadim@mii.lu.lv , vaden@ida.liu.se.

Some info and pointers to papers are available from Rigal home page
Rigal and papers about it available by ftp from RIGAL FTP directory