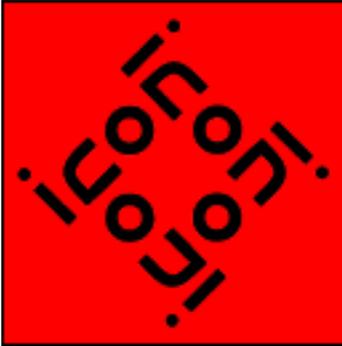


Icon 9 Variant Translator UNIX Manual Page

Ralph E. Griswold



Department of Computer Science
The University of Arizona
Tucson, Arizona

IPD246c
November 29, 1996
<http://www.cs.arizona.edu/icon/docs/ipd246.html>

NAME

`icon_vt` -- construct Icon variant translator system

SYNOPSIS

`icon_vt`

DESCRIPTION

An Icon variant translator system is a facility for constructing preprocessors for variants of Icon.

To set up a variant translator system, a new directory should be created solely for its use; otherwise files may be accidentally destroyed by the set-up process. For the purpose of example, suppose this directory is named `vt`. The set-up process consists of

```
mkdir vt
cd vt
icon_vt
```

Note that `icon_vt` must be run in the area in which the variant translator system is to be built.

The shell script `icon_vt` constructs two subdirectories: `h` and `itran`. The subdirectory `h` contains header files that are needed in C routines. The subdirectory `itran` contains the translator proper.

The main directory in which `icon_vt` is run contains a `Makefile` and auxiliary files needed to construct a variant translator, which is named `vitran`. The two files `variant.defs` and `variant.c` initially are placeholders. It is in these files that the variant is described by macro specifications and any needed C support routines. A *make* in this directory produces a translator, `vitran`. If `variant.def` and `variant.c` are left empty, the result is an identity translator.

If the input syntax of the variant translator is different from that of Icon, it is necessary to modify the Icon grammar, which is in `itran/icon_g.c`.

OPTIONS

`-m` Preprocess each `.icn` source file with the *m4(1)* macro processor.

`-s` Suppress informative messages. Normally, both informative messages and error messages are sent to standard error output.

-P Suppress `#line` directives in the output. This option may be necessary to prevent `#line` directives from occurring in places that result in syntactically incorrect output.

FILES

`v9/vtran` code for building personalized interpreters

SEE ALSO

`icont(1)`

Variant Translators for Version 9 of Icon , Ralph E. Griswold, [IPD245](#), Department of Computer Science, The University of Arizona, 1995.

Building Source-Code Processors for Icon , Ralph E. Griswold, [IPD263](#), Department of Computer Science, The University of Arizona, 1995.

The Icon Programming Language , Ralph E. Griswold and Madge T. Griswold, Peer-to-Peer Communications, Inc., third edition, San Jose, California, 1996.

[Icon home page](#)