env2
by *David C Black*

---

env2

Table of Contents
env2

NAME

env2 - Converts environment variables to various script languages.

SYNOPSIS

env2 -from *SHELL* -to *SHELL* [-o *DESTINATION_FILE*] *SOURCE_FILE*
env2 —save [FILE]
env2 # displays help

DESCRIPTION

Do you prefer running bash while the rest of the team runs tcsh? Or perhaps you learned csh and the system administrators only know ksh. What happens when they (or you) supply an initialization script to source? If any of these situations sounds familiar, then this may be the script for you.
env2 takes shell scripts of one flavor in and spits out scripts effectively equivalent in another dialect. We say "effectively equivalent" because it does not translate syntax such as **if**/**else** statements. Instead, the original (source) script is evaluated to determine what environment variables it modifies, and the effective values of those variables are simply expressed in the syntax of the destination script‘s dialect. Typically, this is all you really need for scripts that modify the environment.
NOTE: If you need to the conditionals and for-loops to be used for different situations (e.g. different host architectures), then simply use this script repeatedly as needed.
Future extensions may include **aliases** or <functions>. Supported languages currently include: **bash**, **csh**, **ksh**, **modulecmd**, **perl**, **plist**, **sh**, **tclsh**, **tcsh**, **vim**, **yaml**, and **zsh**.
NOTE: The file version is identified by an internally computed SHA1 hash similar to the way **git** does versioning. If you get a warning message about inconsistent hash, it means that somebody modified the file without updating the $SHA variable.

OPTIONS

-all

Include all variables rather than just those that changed. See **-diff**.

-clear

Resets the ignore list to almost completely empty except for the generally dangerous environment list.
See **-ignore**, **-reset**, and **-unsafe_clear**.

-diff

Only include those variables whose values are different as a result of sourcing the specified input script. This is the default action.

-from DIALECT

The dialect to translate from. If not present, then the input file is examined for a #! line. If that fails, the **SHELL** environment variable is examined. If that fails, we default to **bash** (Hey, I had to choose something, and **bash** is the default Linux shell of choice).

-help

This built-in documentation. Written in POD so that you can also have it in HTML, PDF, RTF, plain text or as a man page.

-ignore VARLIST

Comma separated list of variables to ignore. By default, **env2** starts with the list set to
`_,ENV,ENV2,OLDPWD,SHLVL`

See also **-clear**, **-reset** and **-unsafe_clear**.

-ignored

Lists variables that will be ignored. Useful if you are uncertain and want to clarify things before proceeding.

-o [*FILE*]

Specifies a *FILE* to save the results in. By default results are sent to STDOUT. If you leave off the *FILE*, the filename will be `env2.$SHELL`.

-profile [*FILE*]

Specifies a file that contains the starting environment. Defaults to `.env2profile`. Searches for the file in the current directory or $HOME if path not specified.

-reset

Resets the ignore list to a minimum set.
See also **-clear**, **-ignore** and **-unsafe_clear**.

-save [FILE]
Saves the environment in a perl format. By default saves to env2.pl. Internally, this option is invoked as part of the conversion process and saves an intermediate file to /tmp/env2.$$.pl.

-sha1
Displays the SHA1 version identifier.

-to DIALECT
The dialect to translate to. If not supplied, the **SHELL** environment variable is examined. If that fails, we default to **bash** (Hey, I had to choose something, and **bash** is the default Linux shell of choice).

-uniq *[PATHLIST]*
Ensure that each path variable specified in *PATHLIST* contains a unique colon separated list. Default is to apply this to common path variables:

```
MANPATH
PATH
LD_LIBRARY_PATH
DYLD_LIBRARY_PATH
PERL5LIB
```

-unsafe_clear
Resets the ignore list to completely empty. This is potentially dangerous because allows inclusion of dangerous environment variables that the user should not touch. These dangerous variables can have an adverse affect on operation of UNIX.
See also **-clear**, **-ignore** and **-reset**.

ENVIRONMENT

ENV2 environment variable contains default command-line arguments if defined.

DEPENDENCIES

To support YAML, this script uses the CPAN YAML module.

EXAMPLES

```
# Simple conversion
env2 -from ksh -to csh -o setup.csh setup.ksh

# Create a modules cmd for a new xyz tool version 1.0
TOOL_SETUP=$TOOLS/vendor_dir/xyz_tool/xyz-1.0/setup.script
MDLDIR=$TOOLS/modules/tools/xyz
env2 -from sh -to modulecmd -o $MDLDIR/1.0 $TOOL_SETUP

# Dynamically use a script for another shell to set environment
eval ‘env2 -from ksh -to $SHELL -o setup.csh setup.ksh‘
```

COPYRIGHT/LICENSE

AUTHOR

David C. Black <dcblack@hldwizard.com>