export Cheatsheet

By Dejan Panovski • Updated on Mar 16, 2026 • Download PDF

Quick reference for exporting shell variables and functions to child processes with export in Bash

The `export` builtin marks shell variables and functions so child processes can inherit them. This cheatsheet covers variable export, function export, removing the export property, and common shell-session patterns.

Basic Syntax

Core `export` command forms in Bash.

`export VAR=value`	Create and export a variable
`export VAR`	Export an existing shell variable
`export -p`	List all exported variables
`export -n VAR`	Remove the export property
`help export`	Show Bash help for `export`

Export Variables

Use `export` to pass variables to child processes.

`export APP_ENV=production`	Export one variable
`PORT=8080; export PORT`	Define first, export later
`export PATH="$HOME/bin:$PATH"`	Extend `PATH`
`export EDITOR=nano`	Set a default editor
`export LANG=en_US.UTF-8`	Set a locale variable

Export for One Shell Session

These changes last only in the current shell session unless you save them in a startup file.

`export DEBUG=1`	Enable a debug variable for the current session
`export API_URL=https://api.example.com`	Set an application endpoint
`export PATH="$HOME/.local/bin:$PATH"`	Add a per-user bin directory
`echo "$DEBUG"`	Verify that the exported variable is set
`bash -c 'echo \"$DEBUG\"'`	Confirm the child shell inherits the variable

Export Functions

Bash can export functions to child Bash shells.

`greet() { echo "Hello"; }`	Define a shell function
`export -f greet`	Export a function
`bash -c 'greet'`	Run the exported function in a child shell
`export -nf greet`	Remove the export property from a function

Remove or Reset Exports

Use these commands when you no longer want a variable inherited.

`export -n VAR`	Keep the variable, but stop exporting it
`unset VAR`	Remove the variable completely
`unset -f greet`	Remove a shell function
`export -n PATH`	Stop exporting PATH in the current shell
`` `env ``	grep '^VAR='

Make Variables Persistent

Add `export` lines to shell startup files when you want them loaded automatically.

`~/.bashrc`	Interactive non-login Bash shells
`~/.bash_profile`	Login Bash shells
`/etc/environment`	System-wide environment variables
`/etc/profile`	System-wide shell startup logic
`source ~/.bashrc`	Reload the current shell after editing

Troubleshooting

Quick checks for common `export` problems.

A child process cannot see the variable	Confirm you used `export VAR` or `export VAR=value`
The variable disappears in a new terminal	Add the export line to `~/.bashrc` or another startup file
A function is missing in a child shell	Export it with `export -f name` and use Bash as the child shell
The shell still sees the variable after `export -n`	`export -n` removes inheritance only; use `unset` to remove the variable completely
The variable is set but a command ignores it	Check whether the program reads that variable or expects a config file instead

Related Guides

Use these guides for broader shell and environment-variable workflows.

export Command in Linux	Full guide to `export` options and examples
How to Set and List Environment Variables in Linux	Broader environment-variable overview
env cheatsheet	Inspect and override environment variables
Bash cheatsheet	Quick reference for Bash syntax and shell behavior