

CONTENTS

Export command in Linux without any arguments

Viewing all exported variables on current shell

Using export with functions

RELATED

What is .bashrc file in Linux?

[View](#) 

How To Execute a Command with a Shell Script in Linux

[View](#) 

// Tutorial //

Export Command in Linux

Published on August 3, 2022

UNIX/Linux



By Pankaj





While we believe that this content benefits our community, we have not yet thoroughly reviewed it. If you have any suggestions for improvements, please let us know by clicking the “report an issue” button at the bottom of the tutorial.

In this guide, we will look at the export command in Linux. Export is a built-in command of the Bash shell. It is used to mark variables and functions to be passed to child processes. Basically, a variable will be included in child process environments without affecting other environments. To get a clearer picture of what we are talking about, let's dive in and have a look at the export command examples.

Export command in Linux without any arguments

Without any arguments, the command will generate or display all exported variables. Below is an example of the expected output.

```
$ export
```

Sample output

```
root@linux:~# export
declare -x HOME="/root"
declare -x LANG="C.UTF-8"
declare -x LESSCLOSE="/usr/bin/lesspipe %s %s"
declare -x LESSOPEN="| /usr/bin/lesspipe %s"
declare -x LOGNAME="root"
```



Viewing all exported variables on current shell

[COOKIE PREFERENCES](#)

If you wish to view all exported variables on the current shell, use the `-p` flag as shown in the example

```
$ export -p
```

Sample output

```
root@linux:/# export -p
declare -x HOME="/root"
declare -x LANG="C.UTF-8"
declare -x LESSCLOSE="/usr/bin/lesspipe %s %s"
declare -x LESSOPEN="| /usr/bin/lesspipe %s"
declare -x LOGNAME="root"
```

Using export with functions

Suppose you have a function and you wish to export it, how do you go about it? In this case, the `-f` flag is used. In this example, we are exporting the function name `()`. First, call the function

```
$ name () { echo "Hello world"; }
```

Then export it using the `-f` flag

```
$ export -f name
```

Next, invoke bash shell

New! Premium CPU-Optimized Droplets are now available.
Learn more →

We're
hiring

Blog

Docs

Get
Support

Sales



Tutorials Questions Learning Paths For Businesses For Builders Social Impact

```
Hello World
```



```
root@linux:/# name () { echo "Hello world"; }
root@linux:/#
root@linux:/# export -f name
root@linux:/#
root@linux:/# bash
root@linux:/#
root@linux:/# name
Hello world
root@linux:/#
```

You can also assign a value before exporting a function as shown

```
$ export name[=value]
```

For example, you can define a variable before exporting it as shown

```
$ student=Divya
```

In the above example, the variable 'student' has been assigned the value 'Divya' To export the variable run

```
$ export students
```

You can use the `printenv` command to verify the contents of the variable as shown

```
$ printenv students
```

Check the output below of the commands we have just executed **Output**

```
root@linux:~# student=Divya
root@linux:~#
root@linux:~# export student
root@linux:~#
root@linux:~# printenv student
Divya
```

The above can be achieved in 2 simple steps by declaring and exporting the variable in one line as shown

```
$ export student=Divya
```

To display the variable run

```
$ printenv student
```

Output

```
root@linux:~# export student=Divya
root@linux:~#
root@linux:~# printenv student
Divya
```

This concludes our tutorial about export command. Go ahead and give it a try and see the magic! Your feedback is most welcome.

Thanks for learning with the DigitalOcean Community. Check out our offerings for compute, storage, networking, and managed databases.

[Learn more about us →](#)

Want to learn more? Join the DigitalOcean Community!

Join our DigitalOcean community of over a million developers for free! Get help and share knowledge in our Questions & Answers section, find tutorials and tools that will help you grow as a developer and scale your project or business, and subscribe to topics of interest.

[Sign up now →](#)

About the authors



Pankaj Author



Still looking for an answer?

Ask a question

Search for more help

Was this helpful?

Yes

No



Comments

[JournalDev](#)  • March 14, 2021 

Thank you so much Pankaj, it will help a lot.

- Rahul Bhichher

[JournalDev](#)  • August 27, 2020 

Good info. I have a command where it says "export MY_KERNEL =/root/xyz/Linux-v4.12/" and then compiling a make file by command "make KDIR=\$MY_KERNEL -j8". what does it mean. I am able to compile the make file using just "make" command only. what is the difference in compiling make file with normal "make" and "make KDIR=\$MY_KERNEL -j8". Please help me

- GSK





This work is licensed under a Creative Commons Attribution-NonCommercial- ShareAlike 4.0 International License.

Try DigitalOcean for free

Click below to sign up and get **\$200 of credit** to try our products over 60 days!

Sign up →

Popular Topics

Ubuntu

Linux Basics

JavaScript

Python

MySQL


Docker


Kubernetes

[All tutorials →](#)

[Free Managed Hosting →](#)

 Congratulations on unlocking the whale ambience easter egg! Click the whale button in the bottom left of your screen to toggle some ambient whale noises while you read.

 Thank you to the [Glacier Bay National Park & Preserve](#) and [Merrick079](#) for the  hands behind this easter egg.

 Interested in whales, protecting them, and their connection to helping prevent climate change? We recommend checking out the [Whale and Dolp](#)



[Reset easter egg to be discovered again](#) / [Permanently dismiss and hide easter egg](#)



Get our biweekly newsletter

Sign up for Infrastructure as a Newsletter.

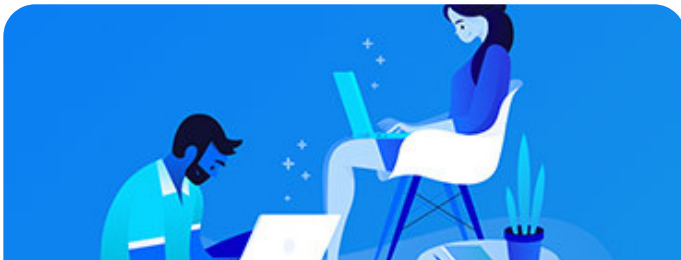
[Sign up →](#)



Hollie's Hub for Good

Working on improving health and education, reducing inequality, and spurring economic growth? We'd like to help.

[Learn more →](#)



Become a contributor

You get paid; we donate to tech nonprofits.

[Learn more →](#)



Featured on Community

[Kubernetes Course](#)

[Learn Python 3](#)

[Machine Learning in](#)



DigitalOcean Products

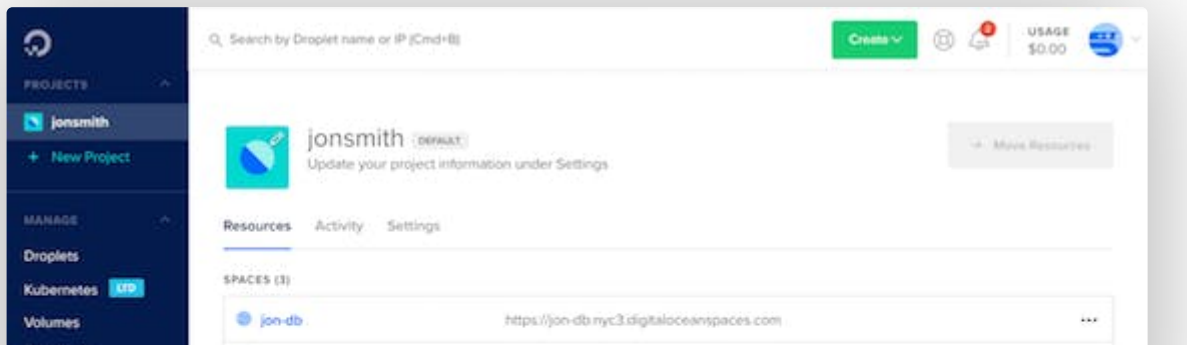
[Cloudways](#)[Virtual Machines](#)[Managed Databases](#)[Managed Kubernetes](#)

[Block Storage](#)[Object Storage](#)[Marketplace](#)[VPC](#)[Load Balancers](#)

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you’re running one virtual machine or ten thousand.

Learn more →



Company	Products	Community	Solutions	Contact
About	Products	Tutorials	Website Hosting	Support
Leadership	Overview	Q&A	VPS Hosting	Sales
Blog	Droplets	CSS-Tricks	Web & Mobile Apps	Report Abuse
Careers	Kubernetes	Write for	Game Development	System Status
Customers	App Platform	DOnations	Streaming	Share your ideas
Partners	Functions	Currents	VPN	
Channel Partners	Cloudways	Research	SaaS Platforms	
	Managed Databases	Hatch Startup Program		

Referral Program	Spaces	deploy by DigitalOcean	Cloud Hosting for Blockchain
Affiliate Program	Marketplace	Shop Swag	Startup Resources
Press	Load Balancers	Research Program	
Legal	Block Storage	Open Source	
Security	Tools & Integrations	Code of Conduct	
Investor Relations	API	Newsletter Signup	
DO Impact	Pricing	Meetups	
	Documentation		
	Release Notes		
	Uptime		

© 2023 DigitalOcean, LLC. All rights reserved.

