

# Package ‘ami’

June 16, 2024

**Title** Checks for Various Computing Environments

**Version** 0.1.1

**Description** A collection of lightweight functions that can be used to determine the computing environment in which your code is running. This includes operating systems, continuous integration (CI) environments, containers, and more.

**License** MIT + file LICENSE

**Imports** curl, glue, lifecycle, rlang, rstudioapi

**Suggests** config, covr, testthat (>= 3.0.0), withr

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**URL** <https://github.com/briandconnelly/ami>,  
<https://briandconnelly.github.io/ami/>

**BugReports** <https://github.com/briandconnelly/ami/issues>

**NeedsCompilation** no

**Author** Brian Connelly [aut, cre, cph]  
(<<https://orcid.org/0000-0002-9948-0379>>),  
Mark Padgham [ctb] (<<https://orcid.org/0000-0003-2172-5265>>),  
Lluís Revilla Sancho [ctb] (<<https://orcid.org/0000-0001-9747-2570>>)

**Maintainer** Brian Connelly <bdc@bconnelly.net>

**Repository** CRAN

**Date/Publication** 2024-06-16 21:10:02 UTC

## Contents

online . . . . .	2
on_bioconductor . . . . .	3

on_cran . . . . .	3
using_account . . . . .	4
using_ci . . . . .	4
using_conda . . . . .	5
using_config . . . . .	6
using_container . . . . .	7
using_covr . . . . .	7
using_cpu . . . . .	8
using_databricks . . . . .	8
using_envvar . . . . .	9
using_nix_shell . . . . .	9
using_option . . . . .	10
using_os . . . . .	11
using_rstudio . . . . .	11
using_r_version . . . . .	12
using_testthat . . . . .	13
using_vscode . . . . .	13
<b>Index</b>	<b>14</b>

---

online	<i>Checks related to networking</i>
--------	-------------------------------------

---

## Description

online() uses [curl::has\\_internet](#) to check whether the machine is connected to the internet

## Usage

```
online()
```

```
using_host(hostname)
```

## Arguments

hostname      String containing a hostname or hostnames to check

## Value

A logical value

## Examples

```
online()
using_host("somehost.fakedomain.com")
```

---

on_bioconductor	<i>Detect Bioconductor</i>
-----------------	----------------------------

---

**Description**

Detect Bioconductor

**Usage**

```
on_bioconductor()
```

**Value**

A logical value

**References**

Check the Bioconductor Build System: <https://github.com/Bioconductor/BBS/>

**Examples**

```
on_bioconductor()
```

---

on_cran	<i>Detect CRAN</i>
---------	--------------------

---

**Description**

This function detects whether the current R environment is a CRAN machine or not. It returns FALSE if the NOT\_CRAN environment variable used in "github/r-lib" packages like **devtools** and **testthat** is set to "true". If that variable is not set, the function examines other environment variables typically set on CRAN machines, as documented in the issue on this packages' GitHub repository at <https://github.com/briandconnelly/ami/issues/14>.

**Usage**

```
on_cran(cran_pattern = "_R_", n_cran_envvars = 5L)
```

**Arguments**

cran_pattern	String to match against environment variables.
n_cran_envvars	If at least this number of environment variables match the cran_pattern, on_cran() returns TRUE.

**Value**

A logical value

**Examples**

```

on_cran()
withr::with_envvar(
  list("NOT_CRAN" = "false", "_R_1" = 1, "_R_2" = 2),
  on_cran(n_cran_envvars = 2L)
)

```

---

using\_account

*Determine whether a given user account is being used*


---

**Description**

Determine whether a given user account is being used

**Usage**

```
using_account(username)
```

**Arguments**

username      Username to check for

**Value**

A logical value

**Examples**

```
using_account("root")
```

---

using\_ci

*Detect whether running in a CI environment*


---

**Description**

using\_ci() reports whether a continuous integration environment is being used.

using\_appveyor() reports whether AppVeyor is being used

using\_circle\_ci() reports whether CircleCI is being used

using\_codebuild() reports whether AWS CodeBuild is being used

using\_github\_actions() reports whether GitHub Actions is being used

using\_gitlab\_ci() reports whether GitLab CI/CD is being used

using\_jenkins() reports whether Jenkins is being used

using\_travis\_ci() reports whether Travis CI is being used

**Usage**

```
using_ci(service = NULL)

using_appveyor()

using_circle_ci()

using_codebuild()

using_github_actions()

using_gitlab_ci()

using_jenkins()

using_travis_ci()
```

**Arguments**

service	If provided, a particular CI service is checked. If not, the commonly-used CI environment variable is checked.
---------	--

**Value**

A logical value

**Examples**

```
using_ci()
using_appveyor()
using_circle_ci()
using_codebuild()
using_github_actions()
using_gitlab_ci()
using_jenkins()
using_travis_ci()
```

---

using\_conda

*Determine whether Conda environment is being used*

---

**Description**

Determine whether Conda environment is being used

**Usage**

```
using_conda(env = NULL)
```

**Arguments**

env                    Optional environment name to compare against

**Value**

A logical value

**Examples**

```
# Check if Conda is being used (regardless of environment name)
using_conda()

# Check if the 'dev' Conda environment is being used
using_conda(env = "dev")
```

---

using_config	<i>Detect whether a configuration is currently active</i>
--------------	---

---

**Description**

Environment-specific configuration values can be used to alter code's behavior in different environments. The `config` package uses the `R_CONFIG_ACTIVE` environment variable to specify the active environment. If `R_CONFIG_ACTIVE` is not set, the "default" configuration is used.

**Usage**

```
using_config(config)
```

**Arguments**

config                Configuration name

**Value**

A logical value

**Examples**

```
# See whether the default configuration is being used
using_config("default")

# See whether the "production" configuration is being used
using_config("production")
```

---

using_container	<i>Detect container environments</i>
-----------------	--------------------------------------

---

**Description**

Detect container environments

**Usage**

```
using_container()
```

```
using_docker_container()
```

```
using_podman_container()
```

```
using_kubernetes()
```

**Value**

A logical value

**Examples**

```
using_container()  
using_docker_container()  
using_podman_container()  
using_kubernetes()
```

---

using_covr	<i>Detect covr</i>
------------	--------------------

---

**Description**

Detect covr

**Usage**

```
using_covr()
```

**Value**

A logical value

**Examples**

```
using_covr()
```

---

`using_cpu`*Processor Checks*

---

**Description**

`using_cpu()` checks whether the machine uses the given type of processor or not.

`using_x86_cpu()` checks whether the machine uses an x86 processor

`using_arm_cpu()` checks whether the machine uses an ARM-based processor

**Usage**

```
using_cpu(arch = c("arm", "x86"))
```

```
using_x86_cpu()
```

```
using_arm_cpu()
```

**Arguments**

`arch` CPU architecture name. Either "arm" or "x86".

**Value**

A logical value

**Examples**

```
using_arm_cpu()
```

```
using_x86_cpu()
```

```
using_arm_cpu()
```

---

`using_databricks`*Detect Databricks Runtime Environment*

---

**Description**

Detect Databricks Runtime Environment

**Usage**

```
using_databricks()
```

**Value**

A logical value



**Examples**

```
using_databricks()
```

---

using_envvar	<i>Determine whether an environment variable is being used</i>
--------------	--

---

**Description**

Determine whether an environment variable is being used

**Usage**

```
using_envvar(x, value = NULL)
```

**Arguments**

x	Environment variable
value	Optional value to compare against

**Value**

A logical value

**Examples**

```
using_envvar("NOT_CRAN")  
using_envvar("CI", "true")
```

---

using_nix_shell	<i>Detect Nix Shell</i>
-----------------	-------------------------

---

**Description**

using\_nix\_shell() checks whether code is running within an environment defined by a **Nix expression**.

**Usage**

```
using_nix_shell(pure = NULL)
```

**Arguments**

pure	Whether or not the environment is pure, meaning most environment variables have been cleared before the shell started.
------	--

**Value**

A logical value

**Examples**

```
# Check for Nix
using_nix_shell()

# Check for Nix in a pure environment
using_nix_shell(pure = TRUE)
```

---

using\_option

*Determine whether a global option is being used*

---

**Description**

Determine whether a global option is being used

**Usage**

```
using_option(x, value = NULL)
```

**Arguments**

x	Option name
value	Optional value to compare against

**Value**

A logical value

**Examples**

```
using_option("width")

using_option("boot.parallel", value = "multicore")
```

---

using_os	<i>Tests for operating systems</i>
----------	------------------------------------

---

**Description**

Tests for operating systems

**Usage**

```
using_os(os = c("linux", "macos", "solaris", "windows"))  
  
using_linux()  
  
using_macos()  
  
using_solaris()  
  
using_windows()
```

**Arguments**

os                    Operating system name. One of "linux", "macos", "solaris", or "windows"

**Value**

A logical value

**Examples**

```
using_os(os = "linux")  
using_linux()  
using_macos()  
using_solaris()  
using_windows()
```

---

using_rstudio	<i>RStudio environments</i>
---------------	-----------------------------

---

**Description**

These functions enable you to determine whether code is being run in the presence of various features of the RStudio IDE and other Posit products.

using\_rstudio() determines whether code is being run in RStudio

using\_rstudio\_jobs() determines whether code is running as an **RStudio Job**

using\_rstudio\_dark\_theme() determines whether a dark theme is being used

using\_posit\_connect() checks whether **Posit Connect** is being used

**Usage**

```
using_rstudio()  
  
using_rstudio_jobs()  
  
using_rstudio_dark_theme()  
  
using_posit_connect()
```

**Value**

A logical value

**See Also**

<https://docs.posit.co/connect/user/content-settings/#content-vars>

**Examples**

```
using_rstudio()  
using_rstudio_jobs()  
using_rstudio_dark_theme()  
using_posit_connect()
```

---

using\_r\_version      *R session information*

---

**Description**

Get information about the R environment being used.

using\_r\_version() determines whether or not a particular version of R is being used.

using\_latest\_r\_version() determines whether or not the latest stable version of R is being used.

using\_interactive\_session() determines whether or not R is being run interactively.

**Usage**

```
using_r_version(ver)  
  
using_latest_r_version()  
  
using_interactive_session()
```

**Arguments**

ver                      Version string

**Value**

A logical value

**Examples**

```
using_r_version(ver = "4.3.0")  
using_latest_r_version()  
using_interactive_session()
```

---

using_testthat	<i>Detect testthat</i>
----------------	------------------------

---

**Description**

Detect testthat

**Usage**

```
using_testthat()
```

**Value**

A logical value

**Examples**

```
using_testthat()
```

---

using_vscode	<i>Detect whether code is running in Visual Studio Code</i>
--------------	---

---

**Description**

Detect whether code is running in Visual Studio Code

**Usage**

```
using_vscode()
```

**Value**

A logical value

**Examples**

```
using_vscode()
```

# Index

`curl::has_internet`, 2

`on_bioconductor`, 3

`on_cran`, 3

`online`, 2

`using_account`, 4

`using_appveyor` (`using_ci`), 4

`using_arm_cpu` (`using_cpu`), 8

`using_ci`, 4

`using_circle_ci` (`using_ci`), 4

`using_codebuild` (`using_ci`), 4

`using_conda`, 5

`using_config`, 6

`using_container`, 7

`using_covr`, 7

`using_cpu`, 8

`using_databricks`, 8

`using_docker_container`  
(`using_container`), 7

`using_envvar`, 9

`using_github_actions` (`using_ci`), 4

`using_gitlab_ci` (`using_ci`), 4

`using_host` (`online`), 2

`using_interactive_session`  
(`using_r_version`), 12

`using_jenkins` (`using_ci`), 4

`using_kubernetes` (`using_container`), 7

`using_latest_r_version`  
(`using_r_version`), 12

`using_linux` (`using_os`), 11

`using_macos` (`using_os`), 11

`using_nix_shell`, 9

`using_option`, 10

`using_os`, 11

`using_podman_container`  
(`using_container`), 7

`using_posit_connect` (`using_rstudio`), 11

`using_r_version`, 12

`using_rstudio`, 11

`using_rstudio_dark_theme`  
(`using_rstudio`), 11

`using_rstudio_jobs` (`using_rstudio`), 11

`using_solaris` (`using_os`), 11

`using_testthat`, 13

`using_travis_ci` (`using_ci`), 4

`using_vscode`, 13

`using_windows` (`using_os`), 11

`using_x86_cpu` (`using_cpu`), 8