



Red Hat Developer Tools 1

Using Go 1.12.12 Toolset

Installing and using the Go 1.12.12 toolset

Red Hat Developer Tools 1 Using Go 1.12.12 Toolset

Installing and using the Go 1.12.12 toolset

Olga Tikhomirova
otikhomi@redhat.com>

Supriya Takkhi

Peter Macko

Kevin Owen

Vladimir Slavik

Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Go is a Red Hat offering for developers on the Red Hat Enterprise Linux platform. The Using Go provides an overview of this product, explains how to invoke and use the Go versions of the tools, and links to resources with more in-depth information.

Table of Contents

CHAPTER 1. GO	3
1.1. ABOUT THE GO TOOLSET	3
1.2. COMPATIBILITY	3
1.3. GETTING ACCESS TO THE GO TOOLSET ON RED HAT ENTERPRISE LINUX 7	3
Additional Resources	4
1.4. INSTALLING THE GO TOOLSET	4
Additional Resources	5
1.5. ADDITIONAL RESOURCES	5
Online Documentation	5
CHAPTER 2. GO	6
2.1. INSTALLING GO	6
2.2. WRITING GO 1.12.12 PROGRAMS	6
Additional Resources	7
2.3. USING THE GO COMPILER	7
2.4. RUNNING A GO PROGRAM	7
2.5. INSTALLING GO PROJECTS	8
2.6. DOWNLOADING GO PROJECTS	8
2.7. ADDITIONAL RESOURCES	9
Installed documentation	9
Online Documentation	9
See Also	9
CHAPTER 3. GOFMT	10
3.1. INSTALLING GOFMT	10
3.2. FORMATTING CODE	10
3.3. PREVIEWING CHANGES TO CODE	10
3.4. SIMPLIFYING CODE	11
3.5. REFACTORING CODE	11
3.6. ADDITIONAL RESOURCES	11
Online documentation	12
See also	12
CHAPTER 4. GO RACE DETECTOR	13
4.1. INSTALLING THE RACE DETECTOR	13
See also	13
4.2. USING THE RACE DETECTOR	13
4.3. ADDITIONAL RESOURCES	13
Online documentation	13
See also	14
CHAPTER 5. CONTAINER IMAGES WITH GO TOOLSET	15
5.1. IMAGES CONTENTS	15
5.2. ACCESSING THE IMAGES	15
5.3. USING AS BUILDER IMAGES WITH SOURCE-TO-IMAGE	15
5.4. ADDITIONAL RESOURCES	16
CHAPTER 6. CHANGES IN THE GO 1.12.12 TOOLSET	17
6.1. GO	17

CHAPTER 1. GO

1.1. ABOUT THE GO TOOLSET

The Go toolset is a Red Hat offering for developers on the Red Hat Enterprise Linux platform. It provides the Go programming language tools and libraries. Go is alternatively known as **golang**.

The Go toolset is distributed as a part of Red Hat Developer Tools for Red Hat Enterprise Linux 7 and is available as a module in Red Hat Enterprise Linux 8.

The following components are available as a part of the Go toolset:

Table 1.1. Go 1.12.12 Components

Name	Version	Description
golang	1.12.12	A Go compiler.

1.2. COMPATIBILITY

The Go toolset is available for Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 8 on the following architectures:

- The 64-bit Intel and AMD architectures
- The 64-bit ARM architecture
- The IBM Power Systems architecture
- The little-endian variant of IBM Power Systems architecture
- The IBM Z Systems architecture

1.3. GETTING ACCESS TO THE GO TOOLSET ON RED HAT ENTERPRISE LINUX 7

The Go toolset is an offering that is distributed as a part of the Red Hat Developer Tools content set, which is available to customers with deployments of Red Hat Enterprise Linux 7. To install the Go toolset, enable the Red Hat Developer Tools and Red Hat Software Collections repositories by using the Red Hat Subscription Management and add the Red Hat Developer Tools GPG key to your system.

1. Enable the **rhel-7-variant-devtools-rpms** repository:

```
# subscription-manager repos --enable rhel-7-variant-devtools-rpms
```

Replace *variant* with the Red Hat Enterprise Linux system variant (**server** or **workstation**).



NOTE

We recommend developers to use Red Hat Enterprise Linux Server for access to the widest range of development tools.

2. Enable the **rhel-variant-rhsc1-7-rpms** repository:

```
# subscription-manager repos --enable rhel-variant-rhsc1-7-rpms
```

Replace *variant* with the Red Hat Enterprise Linux system variant (**server** or **workstation**).

3. Add the Red Hat Developer Tools key to your system:

```
# cd /etc/pki/rpm-gpg
# wget -O RPM-GPG-KEY-redhat-devel https://www.redhat.com/security/data/a5787476.txt
# rpm --import RPM-GPG-KEY-redhat-devel
```

After the subscription is attached to the system, and the repositories are enabled, you can install the Go toolset as described in [Section 1.4, "Installing the Go toolset"](#) .

Additional Resources

- For more information on how to register your system using Red Hat Subscription Management and associate it with subscriptions, see the [Red Hat Subscription Management](#) collection of guides.
- For detailed instructions on subscription to Red Hat Software Collections, see the *Red Hat Developer Toolset User Guide*, [Section 1.4. Getting Access to Red Hat Developer Toolset](#) .

1.4. INSTALLING THE GO TOOLSET

The Go toolset is distributed as a collection of RPM packages that can be installed, updated, uninstalled, and inspected by using the standard package management tools that are included in Red Hat Enterprise Linux. Note that a valid subscription that provides access to the Red Hat Developer Tools content set is required to install the Go toolset on a Red Hat Enterprise Linux 7 system. For detailed instructions on how to associate your Red Hat Enterprise Linux 7 system with an appropriate subscription and get access to the Go toolset, see [Section 1.3, "Getting access to the Go toolset on Red Hat Enterprise Linux 7"](#) .



IMPORTANT

Before installing the Go toolset, install all available Red Hat Enterprise Linux updates.

1. Install all the components included in the Go toolset for your operating system:

- On Red Hat Enterprise Linux 7, install the **go-toolset-1.12.12** package:

```
# yum install go-toolset-1.12
```

- On Red Hat Enterprise Linux 8, install the **go-toolset** module:

```
# yum module install go-toolset
```

This installs all development and debugging tools, and other dependent packages to the system.

2. Create the Go language workspace directory and environment variable:


```
$ mkdir -p workspace_dir
$ echo 'export GOPATH=workspace_dir' >> $HOME/.bashrc
$ source $HOME/.bashrc
```

Select an appropriate value for the ***workspace_dir*** directory. A common choice is **\$HOME/go**.

If the **GOPATH** variable is not set, the **go** compiler uses the **~/go** directory.

Additional Resources

- [Workspaces](#) – Description of the Go language workspace organization. Official documentation for the Go programming language.

1.5. ADDITIONAL RESOURCES

A detailed description of the Go and all its features is beyond the scope of this book. For more information, see the resources listed below.

Online Documentation

- [The Go programming language Documentation](#) – Official documentation for the Go programming language, tools, and libraries.

CHAPTER 2. GO

go is a build tool and dependency manager for the Go programming language.

The Go toolset is distributed with **go 1.12.12**.

2.1. INSTALLING GO

In the Go toolset on Red Hat Enterprise Linux 7, **go** is provided by the **go-toolset-1.12.12-golang** package and is automatically installed with the **go-toolset-1.12.12** package. On Red Hat Enterprise Linux 8, **go** is provided by the **go-toolset** module. See [Section 1.4, "Installing the Go toolset"](#).

2.2. WRITING GO 1.12.12 PROGRAMS

When creating a Go program, developers must follow the rules for Go workspace layout. The **.go** source files must be placed in the subdirectory of **\$GOPATH/src**.

Example 2.1. Creating a Go program

Consider a program named **hello** consisting of a single source file named **hello.go**:

```
$ mkdir -p $GOPATH/src/hello
$ cd $GOPATH/src/hello
$ touch hello.go
```

Edit the file **hello.go**, in a text editor of your choice, to add the following text:

```
package main

import (
    "fmt"
    "net/http"
)

func Welcome(w http.ResponseWriter, req *http.Request) {

    fmt.Fprintf(w, "<h1>Welcome to the Go toolset.</h1>")

}

func main() {

    fmt.Println("Hello.")
    fmt.Println("Starting http server.")
    // Register handler function
    http.HandleFunc("/welcome", Welcome)
    fmt.Println("Go to localhost:8080/welcome")
    fmt.Println("To terminate press CTRL+C.")
    // Start server
    http.ListenAndServe(":8080", nil)

}
```

Additional Resources

- [Workspaces](#) – Description of the Go language workspace organization. Official documentation for the Go programming language.

2.3. USING THE GO COMPILER

To build a Go program using the command line, change to the project directory and run the **go** compiler as follows:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'go build -o output_file go_main_package'
```

- For Red Hat Enterprise Linux 8:

```
$ go build -o output_file go_main_package
```

This creates a binary file named ***output_file*** in the current working directory. If the **-o** option is omitted, the compiler creates a file named after the *go_main_package*, ***go_main_package***.

If *go_main_package* is not a main package or if multiple projects or ***.go** files are specified, the resulting binaries are discarded. In that case, the **go build** command is used to verify that the supplied projects or files can be built.

Note that you can execute any command using the **scl** utility on Red Hat Enterprise Linux 7, causing it to be run with the Go binaries available. This allows you to run a shell session with Go **go** directly available:

```
$ scl enable go-toolset-1.12.12 'bash'
```

Example 2.2. Compiling a Go program using the command line

After you have successfully created the program **hello** as shown in [Example 2.1, “Creating a Go program”](#), compile the program:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'go build hello.go'
```

- For Red Hat Enterprise Linux 8:

```
$ go build hello.go
```

This creates a new binary file called **hello** in the current working directory.

2.4. RUNNING A GO PROGRAM

When **go** compiles a program, it creates an executable binary file. To run this program on the command line, change to the directory with the executable file and run the program:

```
$ ./file_name
```

Example 2.3. Running a Go program on the command line

After you have successfully compiled the **hello** binary file as shown in [Example 2.2, "Compiling a Go program using the command line"](#), run it by typing the following at a shell prompt:

```
$ ./hello
Hello.
Starting http server.
Go to localhost:8080/welcome
To terminate press CTRL+C.
```

2.5. INSTALLING GO PROJECTS

Installing a Go project means that its executable files and libraries are compiled, and copied to appropriate directories in the Go workspace. The **go** tool can then use the executable files and libraries in further projects. Dependencies of the installed project are installed, too.

To install a Go project, run the **go** tool:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'go install go_project'
```

- For Red Hat Enterprise Linux 8:

```
$ go install go_project
```

The **install** command accepts the same options as the [build command](#).

2.6. DOWNLOADING GO PROJECTS

To download a 3rd party Go project from an online source and install it, run the **go** tool:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'go get 3rd_party_go_project'
```

- For Red Hat Enterprise Linux 8:

```
$ go get 3rd_party_go_project
```

For more details about the possible values of the *3rd_party_go_project* option, run the following command:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'go help importpath'
```

- For Red Hat Enterprise Linux 8:

```
$ go help importpath
```

2.7. ADDITIONAL RESOURCES

A detailed description of the **go** compiler and its features is beyond the scope of this book. For more information, see the resources listed below.

Installed documentation

- The Go compiler **help** command provides information on its usage. To show the help index:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'go help'
```

- For Red Hat Enterprise Linux 8:

```
$ go help
```

- The Go compiler **doc** command shows documentation for packages. To show documentation for package *package_name*:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'go doc package_name'
```

To learn more about the **doc** command:

```
$ scl enable go-toolset-1.12.12 'go help doc'
```

- For Red Hat Enterprise Linux 8:

```
$ go doc package_name
```

To learn more about the **doc** command:

```
$ go help doc
```

Online Documentation

- [Command go](#) – Official documentation of the **go** compiler.

See Also

- [Chapter 1, Go](#) – An overview of Go and more information on how to install it on your system.

CHAPTER 3. GOFMT

gofmt is a code formatting tool for the Go programming language, packaged together with the **go** compiler.

The Go toolset is distributed with **gofmt 1.12.12**.

3.1. INSTALLING GOFMT

In the Go toolset, **gofmt** is provided by the **go-toolset-1.12.12-golang** package and is automatically installed with the **go-toolset-1.12.12** package. See [Section 1.4, “Installing the Go toolset”](#).

3.2. FORMATTING CODE

To format all code in the *code_path* path, run the **gofmt** tool as follows:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'gofmt -w code_path'
```

- For Red Hat Enterprise Linux 8:

```
$ gofmt -w code_path
```

This command, directly changes the code in the *code_path* path. When *code_path* is a single file, the changes apply only to the file. When *code_path* is a directory, all **.go** files in the directory are processed.

When the *code_path* is omitted, **gofmt** reads standard input instead.

To print the formatted code to standard output instead of writing to the original file, omit the **-w** option.

It is possible to invoke **gofmt** through the **go** compiler with the **fmt** command. To achieve the same results as the command above, run:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'gofmt -l -w -s code_path'
```

- For Red Hat Enterprise Linux 8:

```
$ gofmt -l -w -s code_path
```

3.3. PREVIEWING CHANGES TO CODE

To preview changes done by formatting code in a given path *code_path*, run the **gofmt** tool with the **-d** option as follows:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'gofmt -d code_path'
```

- For Red Hat Enterprise Linux 8:

-

```
$ gofmt -d code_path
```

The output in unified diff format is printed to standard output.

It is possible to combine both the **-d** and **-w** options.

3.4. SIMPLIFYING CODE

To simplify code in a given path *code_path*, run the **gofmt** tool with the **-s** option as follows:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'gofmt -s code_path'
```

- For Red Hat Enterprise Linux 8:

```
$ gofmt -s code_path
```

The code under *code_path* is simplified. Use the **-d** option to show the differences, and use the **-w** option to apply the changes to the code.

3.5. REFACTORING CODE

The **gofmt** tool can be used to refactor code by applying arbitrary substitutions. To refactor code in a given path *code_path* according to a rule *rewrite_rule*, run the **gofmt** tool with the **-r** option as follows:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'gofmt -r rewrite_rule code_path'
```

- For Red Hat Enterprise Linux 8:

```
$ gofmt -r rewrite_rule code_path
```

The code under *code_path* is refactored according to the rule *rewrite_rule*. Use the **-d** option to show the differences, and use the **-w** option to apply the changes to the code. The additional options must be placed after the rule *rewrite_rule*:

- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'gofmt -r rewrite_rule -d code_path'
```

- For Red Hat Enterprise Linux 8:

```
$ gofmt -r rewrite_rule -d code_path
```

Detailed description of the rewrite rules is beyond the scope of this book. For more information, see the resources listed in [Section 3.6, "Additional Resources"](#).

3.6. ADDITIONAL RESOURCES

A detailed description of the **gofmt** tool and its features is beyond the scope of this book. For more information, see the resources listed below.

Online documentation

- [Command gofmt](#) – Official documentation of the **gofmt** tool.

See also

- [Chapter 1, Go](#) – An overview of the Go toolset and more information on how to install it on your system.

CHAPTER 4. GO RACE DETECTOR

The Go toolset includes the Go race detector. The race detector is a feature of the Go standard library. The race detector must be enabled at compile time and is used at runtime.

4.1. INSTALLING THE RACE DETECTOR



NOTE

Red Hat Enterprise Linux 8 go-toolset module contains Go race detector. If you are on Red Hat Enterprise Linux 8, skip this step.

In the Go toolset, the race detector is provided by the **go-toolset-1.12.12-golang-race** package. To install the race detector:

```
# yum install go-toolset-1.12.12-golang-race
```

This command installs a variant of the Go standard library with the runtime race detection.

See also

- [Section 1.4, “Installing the Go toolset”](#).

4.2. USING THE RACE DETECTOR

To use the runtime race detector in a Go project, add the **-race** option to the **go** tool commands used when manipulating the project.

For a minimal approach to using the race detector, build the project with the **-race** option:

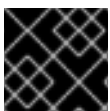
- For Red Hat Enterprise Linux 7:

```
$ scl enable go-toolset-1.12.12 'go build -race -o output_file go_main_package'
```

- For Red Hat Enterprise Linux 8:

```
$ go build -race -o output_file go_main_package
```

Run the resulting executable binary file, and the race detector prints warnings to the standard output when a race is detected.



IMPORTANT

The race detector has a significant runtime resource overhead.

4.3. ADDITIONAL RESOURCES

A detailed description of the Go race detector and its features is beyond the scope of this book. For more information, see the resources listed below.

Online documentation

- [Data Race Detector](#) Official documentation of the Go race detector

- [Data Race Detector](#) – Official documentation of the Go race detector.

See also

- [Chapter 1, Go](#) – An overview of Go and more information on how to install it on your system.

CHAPTER 5. CONTAINER IMAGES WITH GO TOOLSET

Go Toolset is available as container images for RHEL 7 and RHEL 8. They can be downloaded from the Red Hat Container Registry.

5.1. IMAGES CONTENTS

The RHEL 7 and RHEL 8 container images provide content corresponding to the following packages:

Component	Version	Package
Go	1.12	go-toolset-1.12.z

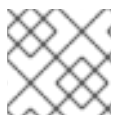
5.2. ACCESSING THE IMAGES

To pull the required image, run the following command as **root**:

- For the RHEL 7 container images:

```
# podman pull registry.redhat.io/devtools/go-toolset-rhel7
```

```
# podman pull registry.redhat.io/ubi7/go-toolset
```



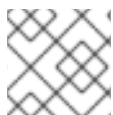
NOTE

Two RHEL 7 container images provide the same content.

- For the RHEL 8 container images:

```
# podman pull registry.redhat.io/rhel8/go-toolset
```

```
# podman pull registry.redhat.io/ubi8/go-toolset
```



NOTE

Two RHEL 8 container images provide the same content.

5.3. USING AS BUILDER IMAGES WITH SOURCE-TO-IMAGE

The Go toolset container image is prepared for use as a Source-to-Image (S2I) builder image.

To do so, set the following build environment variables:

IMPORT_URL

Set this variable to a URL specifying the location of the code. The rules for the [go get command](#) option apply.

INSTALL_URL

Set this variable to a URL specifying the location of the package that will provide the application's main executable file when built. The rules for the [go install command](#) option apply. This variable can be omitted if the main package location is identical with the location specified by the **IMPORT_URL** variable.

Example 5.1. Building a Go application image using Source-to-Image

To build the **md2man** package from its GitHub repository:

```
$ s2i build -e IMPORT_URL='github.com/cpuguy83/go-md2man' -e  
INSTALL_URL='github.com/cpuguy83/go-md2man' git://github.com/cpuguy83/go-md2man  
registry.access.redhat.com/devtools/go-toolset-rhel7  
md2man-app
```

A locally available application image **md2man-app** is built from the repository on GitHub using the **go-toolset** container image.

To fully leverage the Go as a S2I builder image, build custom images based on it, with modified S2I assemble scripts and further modifications to accommodate the particular application being built.

A detailed description of the Go usage with Source-to-Image is beyond the scope of this document. For more information about Source-to-Image, see:

- *OpenShift Container Platform 4.2 Image Creation Guide*, [OpenShift Container Platform-specific guidelines](#)
- *Using Red Hat Software Collections Container Images*, [Chapter 2. Using Source-to-Image \(S2I\)](#).

5.4. ADDITIONAL RESOURCES

- [Go 1.12 container images](#) - entries in the Red Hat Container Catalog
- [Using Red Hat Software Collections Container Images](#)

CHAPTER 6. CHANGES IN THE GO 1.12.12 TOOLSET

This chapter lists some notable changes in the Go 1.12.12 toolset since its previous release.

6.1. GO

Go has been updated from version 1.11.5 to 1.12.12.