



Red Hat Developer Tools 1

Go 1.19.6 Toolset の使用

Go 1.19.6 Toolset のインストールおよび使用

Red Hat Developer Tools 1 Go 1.19.6 Toolset の使用

Go 1.19.6 Toolset のインストールおよび使用

Jacob Valdez

jvaldez@redhat.com

法律上の通知

Copyright © 2023 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.

概要

Go Toolset は、Red Hat Enterprise Linux (RHEL) オペレーティングシステムの開発者向けの Red Hat 製品です。このガイドでは、Go Toolset の概要、さまざまなバージョンの Go ツールを起動して使用する方法、および詳細な情報を含むリソースの検索方法を説明します。

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 GO TOOLSET	5
1.1. GO TOOLSET のコンポーネント	5
1.2. GO TOOLSET の互換性	5
1.3. GO TOOLSET のインストール	5
1.4. GO ドキュメントのインストール	6
1.5. 関連情報	6
第2章 GO コンパイラー	7
2.1. 前提条件	7
2.2. GO ワークスペースの設定	7
2.3. GO プログラムのコンパイル	7
2.4. GO プログラムの実行	8
2.5. コンパイル済み GO プロジェクトのインストール	8
2.6. GO プロジェクトのダウンロードおよびインストール	9
2.7. 関連情報	9
第3章 GOFMT フォーマットツール	11
3.1. 前提条件	11
3.2. コードのフォーマット	11
3.3. コードの変更のプレビュー	11
3.4. コードの簡素化	12
3.5. コードのリファクタリング	12
3.6. 関連情報	12
第4章 GO 競合検出	13
4.1. 前提条件	13
4.2. GO 競合検出の使用	13
4.3. 関連情報	13
第5章 GO TOOLSET が組み込まれたコンテナイメージ	14
5.1. RED HAT ENTERPRISE LINUX GO TOOLSET コンテナイメージのコンテンツ	14
5.2. RED HAT ENTERPRISE LINUX コンテナイメージへのアクセス	14
5.3. RHEL 8 での UBI GO TOOLSET コンテナイメージへのアクセス	14
5.4. RHEL 8 のベース UBI コンテナイメージから GO TOOLSET にアクセスする	14
5.5. 関連情報	15
第6章 GO TOOLSET の変更点	16

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) を参照してください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

特定の文章に関するコメントの送信

1. **Multi-page HTML** 形式でドキュメントを表示し、ページが完全にロードされてから右上隅に **Feedback** ボタンが表示されていることを確認します。
2. カーソルを使用して、コメントを追加するテキスト部分を強調表示します。
3. 強調表示されたテキストの近くに表示される **Add Feedback** ボタンをクリックします。
4. フィードバックを追加し、**Submit** をクリックします。

Jira からのフィードバック送信 (アカウントが必要)

1. [Jira](#) の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. ダイアログの下部にある **Create** をクリックします。

第1章 GO TOOLSET

Go Toolset は、Red Hat Enterprise Linux (RHEL) の開発者向けの Red Hat 製品です。このツールセットは、Go プログラミング言語のツールとライブラリーを提供します。Go は golang と呼ばれています。

Go Toolset は、Red Hat Enterprise Linux 8 のモジュールとして利用できます。また、Red Hat Enterprise Linux 9 のパッケージとしても利用できます。

1.1. GO TOOLSET のコンポーネント

次のコンポーネントが Go Toolset の一部として提供されています。

Name	バージョン	説明
golang	RHEL 7 - 1.5.0 RHEL 8 - 1.5.0	Go コンパイラー。
delve	RHEL 7 - 1.5.0 RHEL 8 - 1.5.0	Go デバッガー。

1.2. GO TOOLSET の互換性

Go Toolset は、Red Hat Enterprise Linux 7 および Red Hat Enterprise Linux 8 では、以下のアーキテクチャーで利用できます。

- AMD および Intel 64 ビット
- 64 ビット ARM
- IBM Power Systems、リトルエンディアン
- 64 ビット IBM Z

1.3. GO TOOLSET のインストール

すべての依存パッケージを含む Go Toolset をインストールするには、以下の手順を実行します。

前提条件

- 利用可能な Red Hat Enterprise Linux のすべての更新がインストールされている。

手順

Red Hat Enterprise Linux 8 の場合、以下のコマンドを実行して **go-toolset** モジュールをインストールします。

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

Red Hat Enterprise Linux 9 の場合、以下のコマンドを実行して **go-toolset** パッケージをインストールします。

```
# dnf install go-toolset
```

1.4. GO ドキュメントのインストール

Go プログラミング言語のドキュメントは、ローカルシステムにインストールできます。

前提条件

- Go Toolset がインストールされている。
詳細は、[Go Toolset のインストール](#) を参照してください。

手順

golang-docs パッケージをインストールするには、以下のコマンドを実行します。

- Red Hat Enterprise Linux 8 の場合

```
# yum install golang-docs
```

ドキュメントは、`/usr/lib/golang/doc/go_spec.html` にあります。

- Red Hat Enterprise Linux 9 の場合:

```
# dnf install golang-docs
```

ドキュメントは、`/usr/lib/golang/doc/go_spec.html` にあります。

1.5. 関連情報

- Go プログラミング言語、ツール、およびライブラリーの詳細は、[公式の Go ドキュメント](#) を参照してください。

第2章 GO コンパイラー

Go コンパイラーは、Go プログラミング言語のビルドツールおよび依存関係マネージャーです。コードのエラーチェックおよび最適化機能を提供します。

2.1. 前提条件

- Go Toolset がインストールされている。
詳細は、[Go Toolset のインストール](#) を参照してください。

2.2. GO ワークスペースの設定

Go プログラムをコンパイルするには、Go ワークスペースを設定する必要があります。

手順

1. ワークスペースディレクトリーを **\$GOPATH/src** のサブディレクトリーとして作成します。
一般的な選択肢は **\$HOME/go** です。
2. ソースファイルをワークスペースディレクトリーに配置します。
3. 以下を実行して、ワークスペースディレクトリーの場所を環境変数として **\$HOME/.bashrc** ファイルに設定します。

```
$ echo 'export GOPATH=<workspace_dir>' >> $HOME/.bashrc  
$ source $HOME/.bashrc
```

<workspace_dir> は、ワークスペースディレクトリーの名前に置き換えます。

関連情報

- [公式の Go ワークスペースのドキュメント](#)

2.3. GO プログラムのコンパイル

Go コンパイラーを使用すると、Go プログラムをコンパイルできます。Go コンパイラーは、コンパイルの結果として、実行可能なバイナリーファイルを作成します。

前提条件

- モジュールが設定された Go ワークスペースが設定されている。
ワークスペースの設定方法は、[Go ワークスペースの設定](#) を参照してください。

手順

プロジェクトディレクトリーで以下を実行します。

- Red Hat Enterprise Linux 8 の場合

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

- <output_file> は出力ファイルの名前に、<go_main_package> はメインパッケージの名前に置き換えます。

- Red Hat Enterprise Linux 9 の場合:

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

- **<output_file>** は出力ファイルの名前に、**<go_main_package>** はメインパッケージの名前に置き換えます。

2.4. GO プログラムの実行

Go コンパイラーは、コンパイルの結果として、実行可能なバイナリーファイルを作成します。このファイルを実行し、プログラムを実行するには、以下の手順を実行します。

前提条件

- プログラムがコンパイルされている。
プログラムをコンパイルする方法は、[Go プログラムのコンパイル](#) を参照してください。

手順

プログラムを実行するには、実行ファイルが含まれるディレクトリーで実行します。

```
$ ./<file_name>
```

- **<file_name>** は、実行ファイルの名前に置き換えます。

2.5. コンパイル済み GO プロジェクトのインストール

コンパイル済みの Go プロジェクトをインストールすると、その実行ファイルとライブラリーを他の Go プロジェクトで使用できます。インストール後、プロジェクトの実行ファイルとライブラリーは、Go ワークスペースのディレクトリーに従ってコピーされます。その依存関係もインストールされません。

前提条件

- モジュールが設定された Go ワークスペース。
詳細は、[Go ワークスペースの設定](#) を参照してください。

手順

Go プロジェクトをインストールするには、以下を実行します。

- Red Hat Enterprise Linux 8 の場合

```
$ go install <go_project>
```

- **<go_project>** は、インストールする Go プロジェクトの名前に置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ go install <go_project>
```

- **<go_project>** は、インストールする Go プロジェクトの名前に置き換えます。

2.6. GO プロジェクトのダウンロードおよびインストール

オンラインリソースからサードパーティーの Go プロジェクトをダウンロードしてインストールすると、その実行ファイルとライブラリーを他の Go プロジェクトで使用できます。インストール後、プロジェクトの実行ファイルとライブラリーは、Go ワークスペースのディレクトリーに従ってコピーされます。その依存関係もインストールされます。

前提条件

- Go ワークスペース。
詳細は、[Go ワークスペースの設定](#) を参照してください。

手順

- Go プロジェクトをダウンロードしてインストールするには、以下を実行します。

- Red Hat Enterprise Linux 8 の場合

```
$ go install <third_party_go_project>
```

- **<third_party_go_project>** は、ダウンロードするプロジェクトの名前に置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ go install <third_party_go_project>
```

- **<third_party_go_project>** は、ダウンロードするプロジェクトの名前に置き換えます。

- サードパーティープロジェクトで使用可能な値を確認するには、以下を実行します。

- Red Hat Enterprise Linux 8 の場合

```
$ go help importpath
```

- Red Hat Enterprise Linux 9 の場合

```
$ go help importpath
```

2.7. 関連情報

- Go コンパイラーの詳細は、[公式の Go ドキュメント](#) を参照してください。
- Go Toolset に含まれる **help** インデックスを表示するには、以下を実行します。

- Red Hat Enterprise Linux 8 の場合

```
$ go help
```

- Red Hat Enterprise Linux 9 の場合:

```
$ go help
```

- 特定の Go パッケージのドキュメントを表示するには、以下を実行します。

- Red Hat Enterprise Linux 8 の場合

```
$ go doc <package_name>
```

- Red Hat Enterprise Linux 9 の場合:

```
$ go doc <package_name>
```

Go パッケージの概要は、[Go パッケージ](#) を参照してください。

第3章 GOFMT フォーマットツール

Go プログラミング言語は、スタイルガイドではなく、**gofmt** コードフォーマットツールを使用します。**gofmt** は、Go レイアウトルールに従ってコードを自動的にフォーマットします。

3.1. 前提条件

- Go Toolset がインストールされている。
詳細は、[Go Toolset のインストール](#) を参照してください。

3.2. コードのフォーマット

gofmt フォーマットツールを使用して、特定のパス内のコードをフォーマットできます。パスが1つのファイルである場合、変更はファイルにのみ適用されます。パスがディレクトリーである場合、ディレクトリー内のすべての **.go** ファイルが処理されます。

手順

特定のパス内のコードをフォーマットするには、以下を実行します。

- Red Hat Enterprise Linux 8 の場合

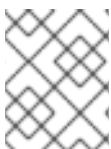
```
$ gofmt -w <code_path>
```

- **<code_path>** は、フォーマットするコードへのパスに置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -w <code_path>
```

- **<code_path>** は、フォーマットするコードへのパスに置き換えます。



注記

元のファイルに書き込みせずにフォーマットされたコードを標準出力に出力するには、**-w** オプションを省略します。

3.3. コードの変更のプレビュー

gofmt フォーマットツールを使用して、特定のパス内のコードをフォーマットすることによって行われた変更をプレビューできます。統一された diff 形式の出力が、標準出力に出力されます。

手順

特定のパス内のコードの違いを表示するには、以下を実行します。

- Red Hat Enterprise Linux 8 の場合

```
$ gofmt -w -d <code_path>
```

- **<code_path>** は、比較するコードへのパスに置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -w -d <code_path>
```

- **<code_path>** は、比較するコードへのパスに置き換えます。

3.4. コードの簡素化

gofmt フォーマットツールを使用して、コードを簡素化できます。

手順

- 特定のパス内の `anf` 形式のコードを簡素化するには、以下を実行します。
 - Red Hat Enterprise Linux 8 の場合

```
$ gofmt -s -w <code_path>
```

- **<code_path>** は、簡素化するコードへのパスに置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -s -w <code_path>
```

- **<code_path>** は、簡素化するコードへのパスに置き換えます。

3.5. コードのリファクタリング

gofmt フォーマットツールを使用して、任意の置換を適用してコードをリファクタリングできます。

手順

- 特定のパスにコードをリファクタリングしてフォーマットするには、以下を実行します。
 - Red Hat Enterprise Linux 8 の場合

```
$ gofmt -r -w <rewrite_rule> <code_path>
```

- **<code_path>** はリファクタリングするコードへのパスに、**<rewrite_rule>** はコードを書き換えるルールに置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -r -w <rewrite_rule> <code_path>
```

- **<code_path>** はリファクタリングするコードへのパスに、**<rewrite_rule>** はコードを書き換えるルールに置き換えます。

3.6. 関連情報

- [公式の gofmt ドキュメント](#)。

第4章 GO 競合検出

Go Toolset には、競合状態を検出する Go 標準ライブラリーのツールである Go 競合検出ツールが含まれています。

競合検出を使用すると、ランタイムリソースのオーバーヘッドが著しく増大することに注意してください。

4.1. 前提条件

- Go Toolset がインストールされている。
詳細は、[Go Toolset のインストール](#) を参照してください。

4.2. GO 競合検出の使用

Go 競合検出を使用して、競合状態のコードを確認します。

手順

競合検出を使用するには、以下を実行します。

- Red Hat Enterprise Linux 8 の場合

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

- **<output_file>** は実行ファイルの名前に、**<go_main_package>** はテストするパッケージ名に置き換えます。

- Red Hat Enterprise Linux 9 の場合:

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

- **<output_file>** は実行ファイルの名前に、**<go_main_package>** はテストするパッケージ名に置き換えます。

4.3. 関連情報

- [公式の Go 競合検出ドキュメント](#)。

第5章 GO TOOLSET が組み込まれたコンテナイメージ

Red Hat Enterprise Linux コンテナイメージまたは Red Hat Universal Base Images (UBI) から独自の Go Toolset コンテナをビルドできます。

5.1. RED HAT ENTERPRISE LINUX GO TOOLSET コンテナイメージのコンテンツ

Go Toolset の Red Hat Enterprise Linux 8 および Red Hat Enterprise Linux 9 コンテナイメージには、以下のパッケージが含まれています。

コンポーネント	バージョン	パッケージ
Go	1.20	RHEL 7 - go-toolset-1.15.11 RHEL 8 - go-toolset-1.15.7

5.2. RED HAT ENTERPRISE LINUX コンテナイメージへのアクセス

コンテナを実行し、アクションを実行する前に、Red Hat レジストリーからコンテナイメージをプルします。

手順

必要なイメージをプルするには、以下を実行します。

- Red Hat Enterprise Linux 8 をベースとするイメージの場合:

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

- Red Hat Enterprise Linux 9 をベースとするイメージの場合:

```
# podman pull registry.redhat.io/rhel9/go-toolset
```

5.3. RHEL 8 での UBI GO TOOLSET コンテナイメージへのアクセス

RHEL 8 では、UBI Go Toolset コンテナイメージをインストールして Go Toolset にアクセスします。

または、Go Toolset を RHEL8 ベースの UBI コンテナイメージにインストールすることもできます。詳細は、[RHEL 8 のベース UBI コンテナイメージから Go Toolset にアクセスする](#) を参照してください。

手順

Red Hat レジストリーから UBI Go Toolset コンテナイメージをプルするには、以下を実行します。

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

5.4. RHEL 8 のベース UBI コンテナイメージから GO TOOLSET にアクセスする

RHEL 8 では、Go Toolset パッケージは Red Hat Universal Base Images (UBI) リポジトリの一部です。つまり、ベース UBI コンテナイメージへの追加として Go Toolset をインストールできます。コンテナイメージのサイズを小さくするには、Go Toolset 全体ではなく、個々のパッケージだけをインストールします。

または、UBI Go Toolset コンテナイメージをインストールして、Go Toolset にアクセスすることもできます。詳細は、[RHEL 8 での UBI Go Toolset コンテナイメージへのアクセス](#) を参照してください。

前提条件

- 既存の Containerfile がある。
Containerfiles 作成の詳細は、[Dockerfile reference](#) のページを参照してください。

手順

- Go Toolset を含むコンテナイメージを作成するには、以下の行を Containerfile に追加します。

```
FROM registry.access.redhat.com/ubi8/ubi:latest
```

```
RUN yum module install -y go-toolset
```

- 個々のパッケージのみを含むコンテナイメージを作成するには、以下の行を Containerfile に追加します。

```
RUN yum install -y <package-name>
```

- <package-name> をインストールするパッケージの名前に置き換えます。

5.5. 関連情報

- [Go Toolset Container Images in the Red Hat Container Registry](#) .
- Red Hat UBI イメージの詳細は、[Working with Container Images](#) を参照してください。
- Red Hat UBI リポジトリの詳細は、[Universal Base Images \(UBI\): イメージ、リポジトリ、パッケージ、およびソースコード](#) を参照してください。

第6章 GO TOOLSET の変更点

Rust ツールセットが、RHEL 7 および RHEL 8 のバージョン 1.52.1 から 1.54.0 に更新されました。

主な変更点は、以下のとおりです。

- 内部表現に依存せずにスライスと文字列を処理するための新しい関数が **unsafe** パッケージに追加されました。
- 比較可能な型は、比較可能な制約を満たすことができるようになりました。
- 新しい **crypto/ecdh** パッケージ。
- **go build** および **go test** コマンドは、**-i** フラグを受け入れなくなりました。
- **go generate** および **go test** コマンドは、**-skip pattern** オプションを受け入れるようになりました。
- **go build**、**go install**、およびその他のビルド関連コマンドは、**-pgo** フラグと **-cover** フラグをサポートするようになりました。
- **go** コマンドは、C ツールチェーンのないシステムではデフォルトで **cgo** を無効にするようになりました。
- **go version -m** コマンドは、より多くの Go バイナリータイプの読み取りをサポートするようになりました。
- **go** コマンドは、C ツールチェーンのないシステムではデフォルトで **cgo** を無効にするようになりました。
- コードカバレッジプロファイルを単体テストからのみ収集するのではなく、アプリケーションと統合テストから収集するためのサポートが追加されました。

更新の詳細は、アップストリームの [Go 1.16 Release Notes](#) を参照してください。