



# Red Hat Developer Tools 2019.3

## Go Toolset の使用

Go Toolset 1.11 のインストールおよび使用



# Red Hat Developer Tools 2019.3 Go Toolset の使用

---

Go Toolset 1.11 のインストールおよび使用

Peter Macko  
pmacko@redhat.com

Kevin Owen

Vladimir Slavik

## 法律上の通知

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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 プラットフォームで開発者向けの Red Hat 製品です。Using Go Toolset では、この製品の概要、ツールの Go Toolset バージョンの呼び出しおよび使用方法、および詳細な情報を含むリソースへのリンクを説明します。

## 目次

|   |           |
|---|-----------|
| <b>第1章 GO TOOLSET</b> .....                                       | <b>3</b>  |
| 1.1. GO TOOLSET について  | 3         |
| 1.2. 互換性  | 3         |
| 1.3. RED HAT ENTERPRISE LINUX 7 で GO TOOLSET へのアクセス               | 3         |
| 1.4. GO TOOLSET のインストール   | 4         |
| 1.5. 関連資料   | 5         |
| <b>第2章 GO</b> .....   | <b>6</b>  |
| 2.1. インストール方法   | 6         |
| 2.2. GO 1.11.6 プログラムの作成   | 6         |
| 2.3. GO コンパイラーの使用   | 7         |
| 2.4. GO プログラムの実行  | 7         |
| 2.5. 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   | 8         |
| 2.7. 関連資料   | 9         |
| <b>第3章 GOFMT</b> .....  | <b>10</b> |
| 3.1. GOFMT のインストール  | 10        |
| 3.2. コードのフォーマット   | 10        |
| 3.3. コードの変更のプレビュー   | 10        |
| 3.4. コードの簡素化  | 11        |
| 3.5. コードのリファクタリング   | 11        |
| 3.6. 関連資料   | 12        |
| <b>第4章 GO RACE DETECTOR</b> .....                                 | <b>13</b> |
| 4.1. RACE DETECTOR ツールのインストール                                     | 13        |
| 4.2. レ이스検出ベクトルの使用   | 13        |
| 4.3. 関連資料   | 13        |
| <b>第5章 GO TOOLSET FOR RHEL 7 のコンテナイメージ</b> .....                  | <b>15</b> |
| 5.1. イメージの内容  | 15        |
| 5.2. イメージへのアクセス   | 15        |
| 5.3. SOURCE-TO-IMAGE での BUILDER イメージとしての使用状況                      | 15        |
| 5.4. 関連資料   | 16        |
| <b>第6章 RED HAT DEVELOPER TOOLS 2019.3 の GO TOOLSET の変更点</b> ..... | <b>17</b> |
| 6.1. GO   | 17        |



# 第1章 GO TOOLSET

## 1.1. GO TOOLSET について

Go Toolset は、Red Hat Enterprise Linux プラットフォームで開発者向けの Red Hat 製品です。Go Toolset 1.13 は、Go プログラミング言語ツールおよびライブラリーを提供します。Go は、**golang** としても知られています。

Go Toolset は、Red Hat Enterprise Linux 7 の一部として配布されており、Red Hat Enterprise Linux 8 でモジュールとして利用できます。

以下のコンポーネントは、Go Toolset の一部として利用できます。

表1.1 Go Toolset 1.11.6 コンポーネント

| 名前     | バージョン  | 説明         |
|--------|--------|------------|
| golang | 1.11.6 | Go コンパイラー。 |

## 1.2. 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。

- 64 ビット Intel および AMD アーキテクチャー
- 64 ビット ARM アーキテクチャー
- IBM Power Systems アーキテクチャー
- IBM Power Systems アーキテクチャーのリトルエンディアンバリエーション
- IBM Z Systems アーキテクチャー

## 1.3. RED HAT ENTERPRISE LINUX 7 で GO TOOLSET へのアクセス

Go Toolset は、Red Hat Developer Tools コンテンツセットの一部として配布されるオフリングです。Red Hat Enterprise Linux 7 のデプロイメントをご利用いただけます。Go Toolset をインストールするには、Red Hat Subscription Management を使用して Red Hat Developer Tools および Red Hat Software Collections リポジトリを有効にし、システムに Red Hat Developer Tools GPG キーを追加します。

1. **rhel-7-variant-devtools-rpms** リポジトリを有効にするには、以下を行います。

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

**variant** を、Red Hat Enterprise Linux システムのバリエーション ( **server** または **workstation** ) に置き換えます。



### 注記

開発者は、Red Hat Enterprise Linux Server を使用して、最も幅広い開発ツールを利用することが推奨されます。

2. **rhel-variant-rhsc1-7-rpms** リポジトリを有効にします。

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

**variant** を、Red Hat Enterprise Linux システムのバリエーション (**server** または **workstation**) に置き換えます。

3. Red Hat Developer Tools キーをシステムに追加します。

```
# 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
```

サブスクリプションがシステムに割り当てられ、リポジトリを有効にしたら、「[Go Toolset のインストール](#)」の説明に従って Red Hat Go Toolset をインストールできます。

## 関連資料

- Red Hat Subscription Management を使用してシステムを登録し、サブスクリプションに関連付ける方法は、[Red Hat Subscription Management](#) のガイドを参照してください。
- Red Hat Software Collections のサブスクリプションに関する詳しい説明は、[Red Hat Developer Toolset User Guide](#) の [セクション 1.4](#) を参照してください。[Red Hat Developer Toolset へのアクセスの取得](#)

## 1.4. GO TOOLSET のインストール

Go Toolset は、Red Hat Enterprise Linux に含まれる標準のパッケージ管理ツールを使用してインストール、更新、アンインストール、および検査できる RPM パッケージのコレクションとして配布されています。Go Toolset を Red Hat Enterprise Linux 7 システムにインストールするには、Red Hat Developer Tool コンテンツセットへのアクセスを提供する有効なサブスクリプションが必要です。Red Hat Enterprise Linux 7 システムを適切なサブスクリプションに関連付け、Go Toolset にアクセスする方法は、「[Red Hat Enterprise Linux 7 で Go Toolset へのアクセス](#)」を参照してください。



### 重要

Go Toolset をインストールする前に、利用可能なすべての Red Hat Enterprise Linux 更新をインストールします。

1. オペレーティングシステムの Go Toolset に含まれるすべてのコンポーネントをインストールします。

- Red Hat Enterprise Linux 7 で、**go-toolset-1.11** パッケージをインストールします。

```
# yum install go-toolset-1.11
```

- Red Hat Enterprise Linux 8 に、**go-toolset** モジュールをインストールします。

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

これにより、すべての開発およびデバッグツール、および依存するパッケージがシステムにインストールされます。

2. Go 言語ワークスペースディレクトリおよび環境変数を作成します。



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

**workspace\_dir** ディレクトリーに適切な値を選択します。一般的な選択肢は **\$HOME/go** です。

**GOPATH** 変数が設定されていない場合、**go** コンパイラーは **~/go** ディレクトリーを使用します。

## 関連資料

- [Workspace](#): Go 言語ワークスペース組織の説明Go プログラミング言語の公式ドキュメント。

## 1.5. 関連資料

Go Toolset およびその機能の詳細は、本書の対象外となります。詳細は、以下に記載のドキュメントを参照してください。

### オンラインドキュメント

- [Go プログラミング言語ドキュメント](#): Go プログラミング言語、ツール、ライブラリーの公式ドキュメント

## 第2章 GO

**go** 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。

Go Toolset には **go 1.11.6** が同梱されています。

### 2.1. インストール方法

Red Hat Enterprise Linux 7 の Go Toolset では、**go** は **go-toolset-1.11-golang** パッケージで提供され、**go-toolset-1.11** パッケージで自動的にインストールされます。Red Hat Enterprise Linux 8 では、**go** は **go-toolset** モジュールにより提供されます。「[Go Toolset のインストール](#)」を参照してください。

### 2.2. GO 1.11.6 プログラムの作成

Go プログラムを作成する場合、開発者は Go ワークスペースレイアウトのルールに従う必要があります。**.go** ソースファイルは、**\$GOPATH/src** のサブディレクトリーに配置する必要があります。

#### 例2.1 Go プログラムの作成

**hello.go** という単一のソースファイルで構成される **hello** というプログラムについて考えてみましょう。

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

お気に入りテキストエディターで **hello.go** ファイルを編集して、以下のテキストを追加します。

```
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)
}
```

## 関連資料

- [Workspace](#): Go 言語ワークスペース組織の説明Go プログラミング言語の公式ドキュメント。

## 2.3. GO コンパイラーの使用

コマンドラインを使用して Go プログラムを構築するには、プロジェクトディレクトリーに移動し、以下のように **go** コンパイラーを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'go build -o output_file go_main_package'
```

- Red Hat Enterprise Linux 8 の場合:

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

これにより、現在の作業ディレクトリーに **output\_file** という名前のバイナリーファイルが作成されます。**-o** オプションを省略すると、コンパイラーは **go\_main\_package**、**go\_main\_package** の名前でファイルを作成します。

**go\_main\_package** がメインパッケージではない場合や、複数のプロジェクトや **\*.go** ファイルが指定されている場合は、生成されるバイナリーは破棄されます。この場合は、**go build** コマンドを使用して、提供されたプロジェクトまたはファイルがビルド可能であることを確認します。

Red Hat Enterprise Linux 7 で **scl** ユーティリティーを使用してコマンドを実行すると、利用可能な Go Toolset バイナリーで実行することができることに注意してください。これにより、Go Toolset **go** でシェルセッションを直接実行できるようになります。

```
$ scl enable go-toolset-1.11 'bash'
```

### 例2.2 コマンドラインを使用した Go プログラムのコンパイル

例2.1「Go プログラムの作成」にあるようにプログラム **hello** が正常に作成済みであると仮定して、プログラムをコンパイルします。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'go build hello.go'
```

- Red Hat Enterprise Linux 8 の場合:

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

これにより、現在の作業ディレクトリー **hello** にという名前のバイナリーファイルが作成されます。

## 2.4. GO プログラムの実行

プログラムを **go** コンパイルすると、実行可能なバイナリーファイルが作成されます。コマンドラインでこのプログラムを実行するには、実行ファイルがあるディレクトリーに移動し、プログラムを実行します。

```
$ ./file_name
```

### 例2.3 コマンドラインで Go プログラムの実行

例2.2「[コマンドラインを使用した Go プログラムのコンパイル](#)」にあるように **hello** バイナリーファイルを正常にコンパイルしたと仮定して、シェルプロンプトで以下のコマンドを実行します。

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

## 2.5. GO プロジェクトのインストール

Go プロジェクトをインストールすると、実行ファイルとライブラリーがコンパイルされ、Go Workspace の適切なディレクトリーにコピーされます。その後、**go** ツールは、追加のプロジェクトで実行ファイルおよびライブラリーを使用できます。インストールされたプロジェクトの依存関係もインストールされます。

Go プロジェクトをインストールするには、**go** ツールを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'go install go_project'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ go install go_project
```

この **install** コマンドは、**build** コマンドと同じオプションを受け入れます。

## 2.6. GO プロジェクトのダウンロード

サードパーティーの Go プロジェクトをオンラインソースからダウンロードし、インストールするには、**go** ツールを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'go get 3rd_party_go_project'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ go get 3rd_party_go_project
```

**3rd\_party\_go\_project** オプションで使用できる値の詳細は、以下のコマンドを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'go help importpath'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ go help importpath
```

## 2.7. 関連資料

go およびその機能の詳しい説明は、本書の対象外となっています。詳細は、以下に記載のドキュメントを参照してください。

インストールされているドキュメント

- Go compiler **help** コマンドは、その使用方法についての情報を提供します。ヘルプインデックスを表示するには、次のコマンドを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'go help'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ go help
```

- Go コンパイラー **doc** コマンドは、パッケージのドキュメントを表示します。 **package\_name** のドキュメントを表示するには、次のコマンドを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'go doc package_name'
```

**doc** コマンドの詳細を確認するには、以下のコマンドを実行します。

```
$ scl enable go-toolset-1.11 'go help doc'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ go doc package_name
```

**doc** コマンドの詳細を確認するには、以下のコマンドを実行します。

```
$ go help doc
```

オンラインドキュメント

- [Command go: go](#) コンパイラーの公式ドキュメント。

関連項目

- [1章Go Toolset](#): Go Toolset の概要およびそのシステムへのインストール方法の詳細

## 第3章 GOFMT

**gofmt** は、**go** コンパイラとともにパッケージ化された Go プログラミング言語用のコードフォーマットツールです。

Go Toolset には、**gofmt 1.11.6** が同梱されています。

### 3.1. GOFMT のインストール

Go Toolset では、**gofmt** は **go-toolset-1.11-golang** パッケージで提供され、**go-toolset-1.11** パッケージで自動的にインストールされます。「[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

**code\_path** パスのすべてのコードをフォーマットするには、以下のように **gofmt** ツールを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'gofmt -w code_path'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -w code_path
```

このコマンドは、**code\_path** パスを直接変更します。**code\_path** 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。**code\_path** がディレクトリーである場合は、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。

**code\_path** を省略すると、**gofmt** は代わりに標準入力を読み込みます。

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。

**fmt** コマンドで **go** コンパイラで **gofmt** を呼び出すことができます。上記のコマンドと同じ結果を実現するには、以下のコマンドを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'gofmt -l -w -s code_path'
```

- Red Hat Enterprise Linux 8 の場合:

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

### 3.3. コードの変更のプレビュー

指定したパス **code\_path** でコードをフォーマットしてプレビューするには、以下のように **-d** オプションを指定して **gofmt** を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'gofmt -d code_path'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -d code_path
```

統一された diff 形式の出力は、標準出力に出力されます。

**-d** と **-w** オプションの両方を組み合わせることができます。

### 3.4. コードの簡素化

指定されたパス **code\_path** でコードを簡素化するには、以下のように **-s** オプションを指定して **gofmt** オプションを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'gofmt -s code_path'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -s code_path
```

**code\_path** の下にあるコードは簡素化されます。**-d** オプションを使用して違いを表示し、**-w** オプションを使用してコードに変更を適用します。

### 3.5. コードのリファクタリング

**gofmt** ツールは、任意の置換を適用してコードをリファクタリングするために使用できます。ルール **rewrite\_rule** に従って、指定のパス **code\_path** のコードをリファクタリングするには、以下のように **-r** オプションを指定して **gofmt** を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'gofmt -r rewrite_rule code_path'
```

- Red Hat Enterprise Linux 8 の場合:

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

**code\_path** のコードは、ルール **rewrite\_rule** に従ってリファクタリングされます。**-d** オプションを使用して違いを表示し、**-w** オプションを使用してコードに変更を適用します。追加オプションは、ルール **rewrite\_rule** の後に配置する必要があります。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'gofmt -r rewrite_rule -d code_path'
```

- Red Hat Enterprise Linux 8 の場合:

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

書き換えルールに関する詳しい説明は、本書の対象外です。詳細は、「[関連資料](#)」に記載されているリソースを参照してください。

## 3.6. 関連資料

`gofmt` ツールおよびその機能の詳しい説明は、本書の対象外となっています。詳細は、以下に記載のドキュメントを参照してください。

### オンラインドキュメント

- [Command gofmt](#) – `gofmt` ツールの公式ドキュメント。

### 関連項目

- [1章 Go Toolset](#): Go Toolset の概要およびそのシステムへのインストール方法の詳細



## 第4章 GO RACE DETECTOR

Go Toolset には Go race detector が含まれます。この race detector は、Go 標準ライブラリーの機能です。race detector はコンパイル時に有効にされる必要があり、ランタイム時に使用されます。

### 4.1. RACE DETECTOR ツールのインストール



#### 注記

Red Hat Enterprise Linux 8 の go-toolset モジュールには、Go 競合検出が含まれています。この手順を省略します。

Go Toolset では、race detector は **go-toolset-1.11-golang-race** パッケージで提供されます。race detector をインストールするには、次のコマンドを実行します。

```
# yum install go-toolset-1.11-golang-race
```

このコマンドは、ランタイム競合検出で Go 標準ライブラリーのバリエーションをインストールします。

#### 関連項目

- [「Go Toolset のインストール」](#)

### 4.2. レ이스検出ベクトルの使用

Go プロジェクトでランタイム競合検出を使用するには、プロジェクトを操作するとき使用する go ツールコマンドに **-race** オプションを追加します。

競合検出を使用する最小限のアプローチには、**-race** オプションを指定してプロジェクトをビルドします。

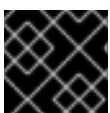
- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.11 'go build -race -o output_file go_main_package'
```

- Red Hat Enterprise Linux 8 の場合:

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

生成された実行可能なバイナリーファイルを実行し、競合検出が検出されると、標準出力に警告が出力されます。



#### 重要

競合検出には、大きなランタイムリソースのオーバーヘッドがあります。

### 4.3. 関連資料

Go 競合検出機能の詳細と、その機能は本ガイドの対象外です。詳細は、以下に記載のドキュメントを参照してください。

オンラインドキュメント

- [Data Race Detector](#): Go 競合検出の公式ドキュメント。

#### 関連項目

- [1章Go Toolset](#): Go Toolset の概要およびそのシステムへのインストール方法の詳細

## 第5章 GO TOOLSET FOR RHEL 7 のコンテナイメージ

Go Toolset は、Red Hat Container Registry からダウンロードできるコンテナイメージとして利用できます。

### 5.1. イメージの内容

devtools/go-toolset-rhel7 イメージは、以下のパッケージに対応するコンテンツを提供します。

| コンポーネント | バージョン  | パッケージ           |
|---------|--------|-----------------|
| Go      | 1.11.6 | go-toolset-1.11 |

### 5.2. イメージへのアクセス

devtools/go-toolset-rhel7 イメージをプルするには、root で以下のコマンドを実行します。

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

### 5.3. SOURCE-TO-IMAGE での BUILDER イメージとしての使用状況

Go Toolset コンテナイメージは、Source-to-Image(S2I) ビルダイメージとして使用するために準備されます。

これを行うには、以下のビルド環境変数を設定します。

#### IMPORT\_URL

この変数を、コードの場所を指定する URL に設定します。go get コマンド オプションのルールが適用されます。

#### INSTALL\_URL

この変数を、ビルド時にアプリケーションのメイン実行可能ファイルを提供するパッケージの場所を指定する URL に設定します。go install コマンド オプションのルールが適用されます。この変数は、メインパッケージの場所が IMPORT\_URL 変数で指定された場所と同一である場合に省略できます。

#### 例5.1 Source-to-Image を使用した Go アプリケーションイメージの構築

md2man パッケージを GitHub リポジトリからビルドするには、以下を実行します。

```
$ 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
```

ローカルで利用可能なアプリケーションイメージ md2man-app は、go-toolset-1.11 コンテナイメージを使用して GitHub のリポジトリからビルドされます。

Go Toolset を S2I ビルダイメージとしてフルに活用するには、変更した S2I assemble スクリプトを使用し、ビルドしている特定のアプリケーションをさらに調整してカスタムイメージをビルドします。

Source-to-Image を使用した Go Toolset の使用方法の詳細な説明は、本書の対象外です。Source-to-Image についての詳細は、[OpenShift Container Platform 3.7 イメージ作成ガイドの第 4 章を参照してください](#)。S2I の要件 および [使用 Red Hat Software Collections コンテナイメージの第 2 章 Source-to-Image \(S2I\) の使用](#)

## 5.4. 関連資料

- [Go Toolset コンテナイメージ](#): Red Hat Container Catalog のエントリー
- [Red Hat Software Collections コンテナイメージの使用](#)

## 第6章 RED HAT DEVELOPER TOOLS 2019.3 の GO TOOLSET の 変更点

本章では、以前のリリース以降の Go Toolset の主な変更点を紹介します。

### 6.1. GO

Go のバージョンが 1.11.5 から 1.11.6 に更新されました。