



ゴールデンISOを使用したインストールのカスタマイズ

表 1:機能の履歴（表）

機能名	リリース情報	説明
ゴールデン ISO（GISO）	リリース 7.3.1	ゴールデン ISO は、カスタマイズ可能な .iso イメージファイルです。このファイルには、iPXE またはシステムアップグレードを使用してインストールできる、導入準備が完了した最小限のソフトウェアイメージ、構成ファイル、およびパッケージが含まれています。GISO ファイルは、要件に基づいてオプションパッケージとSMUを含めるように作成できます。この機能は、Cisco IOS XRv 9000 ルータでサポートされるようになりました。

ゴールデン ISO（GISO）は、ユーザがインストール要件に合わせて作成できるカスタマイズ ISO です。ユーザはインストール可能なイメージをカスタマイズして、基本的な機能コンポーネントを備えた標準的な基本イメージを含め、要件に基づいて RPM、SMU、および設定ファイルを追加することができます。

インストールが簡単であること、およびシステムをシームレスにインストールまたはアップグレードするためにかかる時間は、クラウド規模のネットワークで重要な役割を果たします。時間がかかる複雑なインストール プロセスは、ネットワークの復元力と拡張性に影響します。GISO によってインストール プロセスが簡素化され、インストール ワークフローが自動化され、RPM および SMU の依存関係が自動的に管理されます。

GISO は、[github](#) の場所 ([Github](#) の場所) で利用可能なスクリプト `gisobuild.py` を使用して作成します。作成スクリプトと GISO の作成手順の詳細については、[ゴールデン ISO の構築 \(3 ページ\)](#) を参照してください。

システムを GISO を使用して起動すると、GISO 内の追加の SMU および RPM が自動的にインストールされ、ルータは GISO の XR 設定で事前に設定されます。GISO のダウンロードおよびインストールの詳細については、[ゴールデン ISO のインストール \(5 ページ\)](#) を参照してください。

GISO の機能は次の場合に使用できます。

- IOS XR 32 ビットから IOS XR 64 ビットへの移行
- ルータの初期展開
- ソフトウェア ディザスタ リカバリ
- 1 つの基本バージョンから別のバージョンへのシステム アップグレード
- 追加 SMU を使用した同じ基本バージョンからのシステム アップグレード
- 更新プログラムのインストールと依存パッケージの識別および更新
- [制限事項 \(2 ページ\)](#)
- [ゴールデン ISO ワークフロー \(2 ページ\)](#)
- [ゴールデン ISO の構築 \(3 ページ\)](#)
- [ゴールデン ISO のインストール \(5 ページ\)](#)
- [ゴールデン ISO への置換のインストール \(7 ページ\)](#)

制限事項

次に、カスタム ISO に関する既知の問題および制限事項を示します。

- 非同期パッケージの GISO (ISO とは異なるリリースのパッケージ) の作成と起動はサポートされていません。
- GISO 作成スクリプト `gisobuild.py` は XR 設定の確認をサポートしていません。
- GISO ビルドの名前を変更し、その名前を変更した GISO ビルドのインストールはサポートされていません。

ゴールデン ISO ワークフロー

次の図は、ゴールデン ISO を構築してインストールするためのワークフローを示しています。

ゴールデン ISO の構築

カスタマイズした ISO は、[Github](#) の場所で利用可能なシスコゴールデン ISO（GISO）作成スクリプト `gisobuild.py` を使用して構築します。

GISO 作成スクリプトは、自動依存関係管理をサポートし、次の機能を提供します。

- パッケージリポジトリ内に存在するすべてのパッケージの RPM データベースを構築します。
- `mini-x.iso` バージョンと一致しない Cisco RPM をスキップおよび削除します。
- `mini-x.iso` 内にすでに存在するサードパーティ製の基本パッケージの SMU ではないサードパーティの RPM をスキップおよび削除します。
- 同じリリースで異なるバージョンの基本 RPM が複数ある場合、エラーを表示し、作成プロセスを終了します。
- すべての RPM の互換性チェックと依存関係チェックを実行します。たとえば、子 RPM は親 RPM に依存します。子 RPM のみが含まれる場合、ゴールデン ISO の作成は失敗します。

GISO を作成するには、スクリプトに次の入力パラメータを指定します。

- 基本 `mini-x.iso`（必須）
- XR コンフィギュレーション ファイル（任意）
- ホスト、XR、およびシステム管理用の 1 つまたは複数のシスコ固有の SMU（必須）
- ホスト、XR、およびシステム管理用の 1 つまたは複数のサードパーティ SMU（必須）
- ゴールデン ISO のラベル（任意）



(注) ゴールデン ISO はミニ ISO からのみ作成できます。full または fullk9 バンドル ISO はサポートされていません。

GISO を作成する場合は、次の命名規則を使用します。

GISO ビルド	書式	例
k9sec RPM を使用しない GISO	<code><platform-name>-golden-x.iso-<version>.<label></code> <code><platform-name>-golden-x-<version>.iso.<label></code>	<プラットフォーム名> >-golden-x64.iso-<バージョン>.v1 <プラットフォーム名> >-golden-x64-<バージョン>.iso.v1

GISO ビルド	書式	例
k9sec RPM を使用した GISO	<code><platform-name>-goldenk9-x.iso-<version>.<label></code> <code><platform-name>-goldenk9-x-<version>.iso.<label></code>	<プラットフォーム名> >-goldenk9-x64.iso-<バージョン>.v1 <プラットフォーム名> >-goldenk9-x64-<バージョン>.iso.v1



(注) k9sec RPM を GISO に適切に追加するには、**chmod** コマンドを使用してファイルの権限を 644 に変更します。

```
chmod 644 [k9 sec rpm]
```

GISO を作成するには、次の手順を実行します。

始める前に

- 非 GISO から GISO バージョンにアップグレードするには、最初に GISO サポートを使用して ミニ ISO にアップグレードする必要があります。
- GISO が構築されているシステムは、次の要件を満たしている必要があります。
 - システムには Python バージョン 2.7 以降が必要です。
 - システムには、最低 3 ～ 4 GB の空きディスク領域が必要です。
 - システムに Linux ユーティリティ `mount`、`rm`、`cp`、`umount`、`zcat`、`chroot`、`mkisofs` があることを確認します。これらのユーティリティはスクリプトによって使用されます。これらすべての Linux コマンドを実行する権限があることを確認します。
 - システムのカーネルバージョンは、Cisco ISO のカーネルバージョンより後の 3.16 以降である必要があります。
 - Linux カーネルでサポートされている `libyaml rpm` が、ツールで `yaml` を正常に実行できることを確認します。
 - ユーザは `rpm` リポジトリのセキュリティ `rpm` (`k9sec-rpm`) に対する適切な権限を持っている必要があります。それ以外の場合は、ゴールデン ISO の作成でセキュリティ `rpm` が無視されます。
- `gisobuild` スクリプトが実行されるシステムには、`root` クレデンシャルを使用する必要があります。

ステップ 1 スクリプト `gisobuild.py` を Github の場所から、**GISO**を構築するオフラインシステムまたは外部サーバにコピーします。このシステムが上記の「はじめる前に」セクションに記載された前提条件を満たしていることを確認します。

ステップ 2 スクリプト `gisobuild.py` を実行し、ルータからゴールデン ISO を作成するためのパラメータを指定します。すべての RPM と SMU が同じディレクトリ内に存在することを確認します。ゴールデン ISO の作成に使用できる RPM と SMU の数は 128 です。

(注) `-i` オプションは必須で、`-r` と `-c` のいずれかまたはその両方を指定する必要があります。

```
[directory-path]$ gisobuild.py [-h] [-i <mini-x.iso>] [-r <rpm repository>]
[-c <config-file>] [-l <giso label>] [-m] [-v]
```

次に、スクリプトの出力例を示します。

値は次のとおりです。

- `-i` は `mini-x.iso` へのパスです
- `-r` は RPM リポジトリへのパスです
- `-c` は XR config ファイルへのパスです
- `-l` はゴールデン ISO ラベルです
- `-h` はヘルプ メッセージを表示します
- `-v` は、作成ツール `gisobuild.py` のバージョンです
- `-m` は、IOS XR から IOS XR 64 ビットに移行するための移行 tar を構築します

GISO は、指定されたディレクトリ内の各フォルダに配置された RPM を使用して作成され、ログファイル `giso_summary.txt` および `gisobuild.log-<タイムスタンプ>` も含まれています。XR コンフィギュレーション ファイルはディレクトリ内に `router.cfg` として格納されます。



(注) GISO スクリプトは XR 設定の検証をサポートしていません。

次のタスク

ゴールデン ISO をルータにインストールします。

ゴールデン ISO のインストール

ゴールデン ISO (GISO) は、次のアクションを自動的に実行します。

- ホストおよびシステム管理 RPM をインストールします。

- RP でリポジトリと TFTP ブートをパーティションに分割します。
- システム管理モードおよび XR モードでソフトウェア プロファイルを作成します。
- XR RPM をインストールします。 **show instal active** コマンドを使用して RPM のリストを表示します。
- XR 設定を適用します。XR モードで **show running-config** コマンドを使用して確認します。

ステップ 1 次のいずれかのオプションを使用して、ルータに GISO イメージをダウンロードします。

- **PXE ブート**：ルータが起動すると、ブート モードが識別されます。PXE をブート モードとして検出すると、利用可能なすべてのイーサネットインターフェイスが起動し、各インターフェイスで DHCPClient が実行されます。DHClient スクリプトは HTTP または TFTP プロトコルを解析し、GISO がボックスにダウンロードされます。
- **USB ブートまたはディスク ブート**：ブート中に USB モードが検出され、GISO が識別されると、追加の RPM および XR 設定ファイルが抽出されてインストールされます。
- システムのアップグレード時の **システム アップグレード**では、**install add**、**install activate**、または **install replace** コマンドを使用して GISO をインストールできます。

次に、システムをアップグレードするオプションを示します。

- **非 GISO（GISO をサポートしていないイメージ）から GISO イメージへのシステム アップグレード**：システムが GISO をサポートしていないイメージを使用してバージョン 1 を実行している場合、システムは GISO をサポートするイメージのバージョン 2 に直接アップグレードすることはできません。その代わりに、バージョン 1 をバージョン 2 ミニ ISO にアップグレードし、次にバージョン 2 GISO にアップグレードする必要があります。
- **バージョン 1 GISO からバージョン 2 GISO へのリリースでのシステム アップグレード**：両方の GISO イメージの基本バージョンは同じでラベルが異なる場合、**install add** および **install activate** コマンドは同じバージョンの 2 つのイメージをサポートしません。その代わりに、**install update** コマンドを使用してデルタ RPM のみをインストールします。システムのリロードはデルタ RPM の再起動タイプに基づいています。
- **バージョン 1 GISO からバージョン 2 GISO へのリリース間でのシステム アップグレード**：両方の GISO イメージの基本バージョンが異なります。**install add** および **install activate** コマンド、または **install replace** コマンドを使用して、システムアップグレードを実行します。ルータは、バージョン 2 GISO イメージを使用したアップグレード後にリロードされます。

ステップ 2 システム管理モードで **show install repository all** コマンドを実行し、ホスト、システム管理、および XR の RPM と基本 ISO を表示します。

ステップ 3 **show install package <golden-iso>** コマンドを実行し、RPM のリストおよび GISO に組み込まれているパッケージを表示します。

GISO 内の ISO、SMU、およびパッケージがルータにインストールされます。

ゴールデン ISO への置換のインストール

ゴールデン ISO (GISO) は、単一の操作でソフトウェア メンテナンス アップデート (SMU) の事前定義されたリストを持つバージョンにルータをアップグレードします。ただし、異なる SMU セットを使用した同じバージョンに更新するには、2 段階のプロセスが必要です。

この2段階のプロセスを回避するには、**install replace** コマンドを使用して、現在アクティブなバージョンを、新しく追加した GISO のイメージと SMU を含む完全なパッケージに置き換えます。

このプロセスでは、GISO をアップグレードしてデルタ SMU を追加し、使用されていない SMU を手動で非アクティブにする必要があります。さらにこれは、オプションの実行中の一連の RPM のサブセットであるさまざまなオプション RPM を含んでいる GISO にアップグレードする唯一の方法です。たとえば、GISO の V1 は、V1 mini とオプション RPM の V1 mpls、V1 mpls-te、V1 mgbl、および V1 k9sec を含む実行中のバージョンです。GISO の V2 に V2 k9sec が含まれていない場合は、**install replace** を使用して V2 のオプション RPM にアップグレードします。

ステップ 1 **install replace** <GISO-location> [commit|noprompt]

例：

```
Router#install replace harddisk:/<giso-image>.iso
+++++
Install operation 11 started by root:
exec-timeout is suspended.
No install operation in progress at this moment
Label = More_Pkgs
ISO <giso-iso-image>.iso in input package list. Going to upgrade the system to

version <new-giso-image>.
System is in committed state
Current full-label: <giso-image>_R_Commit
Current only-label: R_Commit
Current label: R_Commit
Updating contents of golden ISO
Scheme : localdisk
Hostname : localhost
Username : None
SourceDir : /ws
Collecting software state..
Getting platform
Getting supported architecture
Getting active packages from XR
Getting inactive packages from XR
Getting list of RPMs in local repo
Getting list of provides of all active packages
Getting provides of each rpm in repo
Getting requires of each rpm in repo
Fetching .... <giso-image>.iso
Label within GISO: More_Pkgs
Skipping <platform>-mgbl-3.0.0.0-<release>.x86_64.rpm from GISO as it's active
Adding packages
<platform>-golden-x-<release>-<Label>.iso
RP/0/RP0/CPU0:Jun 20 14:43:59.349 UTC: sdr_instmgr[1164]: %INSTALL-INSTMGR-2-OPERATION_SUCCESS :
```

```

Install operation 12 finished successfully
Install add operation successful
Activating <platform>-golden-x-<release>-<Label>
Jun 20 14:44:05 Install operation 13 started by root:
  install activate pkg <platform>-golden-x-<release>-<Label> replace noprompt
Jun 20 14:44:05 Package list:
Jun 20 14:44:05   <platform>-golden-x-<release>-<Label>.iso
Jun 20 14:44:29 Install operation will continue in the background
exec-timeout is resumed.
Router# Install operation 13 finished successfully
Router: sdr_instmgr[1164]: %INSTALL-INSTMGR-2-OPERATION_SUCCESS :

Install operation 13 finished successfully

```

新しく追加された GISO のバージョンおよびラベルは、現在アクティブなバージョンのバージョンおよびラベルと比較されます。不一致が特定されると、新しいパーティションが作成され、完全なパッケージがインストールされます。インストール後、システムは新しく追加された GISO からイメージおよびパッケージをリロードします。

ステップ2 show version

例：

```

Router#show version
Wed Jun 20 15:06:37.915 UTC
Cisco IOS XR Software, Version <new-giso-image>
Copyright (c) 2013-2018 by Cisco Systems, Inc.

Build Information:
Built By      : <user>
Built On     : <date>
Build Host   : <host-name>
Workspace    : <workspace-name>
Version      : <version>
Location     : <path>
Label        : <label-name>

cisco <platform> () processor
System uptime is 3 hours 51 minutes

```

システムは新しく追加された GISO からイメージおよびパッケージをリロードします。