



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
バグ修正 RPM のブリッジングの自動インストール	リリース 7.5.1	この機能により、GISO に基づいて、ワンステップかつプロンプトなしで簡単にアップグレードまたはダウングレードできるため、アップグレードまたはダウングレードを実行する前に、バグ修正 RPM のブリッジングを手動でインストールする必要がなくなります。

ゴールデンISO (GISO) は、ユーザー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。ユーザー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。

Cisco IOS XR リリース 7.5.1 以降、バグ修正 RPM のブリッジングの自動インストール機能を使用して、システムアップグレードまたはダウングレードの前提条件であるバグ修正 RPM のブリッジングをインストールできます。シスコゴールデンISO (GISO) ビルドスクリプト `gisobuild.py` を使用してカスタマイズされた ISO ビルドに、必要なバグ修正 RPM のブリッジ

グを追加する必要があります。GISOには、複数のリリースのバグ修正RPMのブリッジングを含めることができますが、ターゲットリリースに必要な特定のバグ修正RPMのブリッジングのみをインストールします。バグ修正RPMのブリッジングは、次のシナリオで使用できます。

- アップグレードを停止する可能性のあるバグを解決するため、
- 最新バージョンには、以前のバージョンでは満たされていない新しい前提条件の要件があります。

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 \(16 ページ\)](#) 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 \(4 ページ\)](#)
- [ゴールデン ISO を使用したインストールのカスタマイズ \(3 ページ\)](#)
- [ゴールデン ISO ワークフロー \(5 ページ\)](#)
- [ゴールデン ISO の構築 \(6 ページ\)](#)
- [ゴールデン ISO のインストール \(16 ページ\)](#)

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。

- サイズが 1.8 GB を超える GISO イメージはサポートされていません。RSP880-LT-SE/TR の最大イメージサイズは 1.599 GB です。
- 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。
- IPv6 を介したインストール操作はサポートされていません。

- GISO を使用した IOS XR 32 ビットから 64 ビット OS への移行には、次の制限があります。
 - IOS XR 32 ビットから 64 ビットへの変換スクリプトでは、48 文字を超えるファイル名はサポートされていません。
 - IOS XR 32 ビット OS の最大ファイルサイズの制限は 2 GB です。GISO のサイズがこの制限を超えないようにしてください。
- 移行方法とシステム要件の詳細については、[Cisco ASR 9000 シリーズルータの移行ガイド \[英語\]](#) を参照してください。

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 2: 機能の履歴 (表)

機能名	リリース情報	説明
バグ修正 RPM のブリッジングの自動インストール	リリース 7.5.1	この機能により、GISO に基づいて、ワンステップかつプロンプトなしで簡単にアップグレードまたはダウングレードできるため、アップグレードまたはダウングレードを実行する前に、バグ修正 RPM のブリッジングを手動でインストールする必要がなくなります。

ゴールデン ISO (GISO) は、ユーザー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。ユーザー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。

Cisco IOS XR リリース 7.5.1 以降、バグ修正 RPM のブリッジングの自動インストール機能を使用して、システムアップグレードまたはダウングレードの前提条件であるバグ修正 RPM のブリッジングをインストールできます。シスコゴールデン ISO (GISO) ビルドスクリプト `gisobuild.py` を使用してカスタマイズされた ISO ビルドに、必要なバグ修正 RPM のブリッジ

グを追加する必要があります。GISOには、複数のリリースのバグ修正RPMのブリッジングを含めることができますが、ターゲットリリースに必要な特定のバグ修正RPMのブリッジングのみをインストールします。バグ修正RPMのブリッジングは、次のシナリオで使用できます。

- アップグレードを停止する可能性のあるバグを解決するため、
- 最新バージョンには、以前のバージョンでは満たされていない新しい前提条件の要件があります。

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 \(16 ページ\)](#) 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

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。

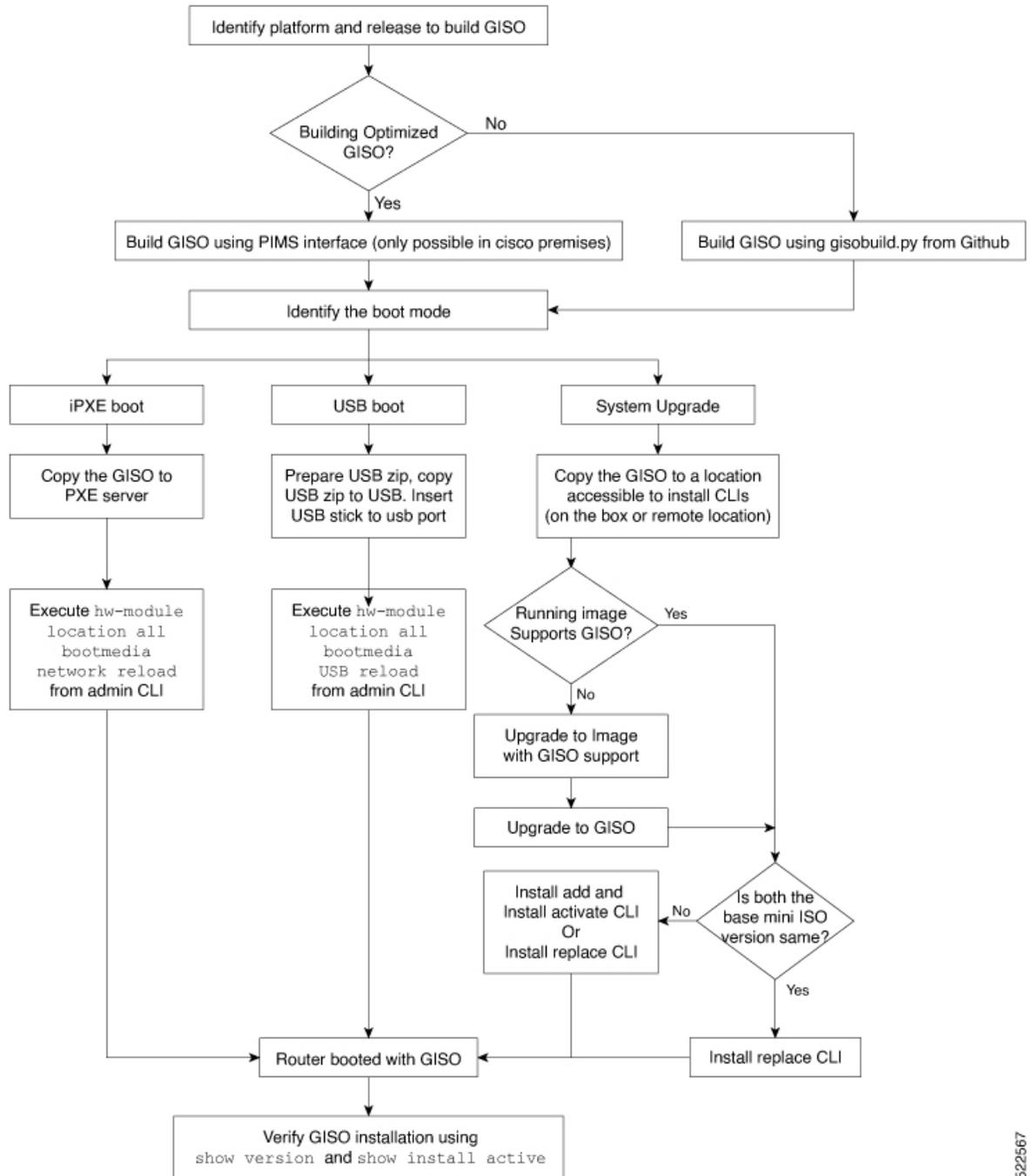
- サイズが 1.8 GB を超える GISO イメージはサポートされていません。RSP880-LT-SE/TR の最大イメージサイズは 1.599 GB です。
- 非同期パッケージの GISO (ISO とは異なるリリースのパッケージ) の作成と起動はサポートされていません。
- GISO 作成スクリプト `gisobuild.py` は XR 設定の確認をサポートしていません。
- GISO ビルドの名前を変更し、その名前を変更した GISO ビルドのインストールはサポートされていません。
- IPv6 を介したインストール操作はサポートされていません。
- GISO を使用した IOS XR 32 ビットから 64 ビット OS への移行には、次の制限があります。
 - IOS XR 32 ビットから 64 ビットへの変換スクリプトでは、48 文字を超えるファイル名はサポートされていません。
 - IOS XR 32 ビット OS の最大ファイルサイズの制限は 2 GB です。GISO のサイズがこの制限を超えないようにしてください。

移行方法とシステム要件の詳細については、[Cisco ASR 9000 シリーズルータの移行ガイド \[英語\]](#) を参照してください。

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。

図 1: ゴールデン ISO ワークフロー



522567

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。
- リポジトリをスキャンし、入力 ISO と一致する関連の Cisco RPM を選択します。
- 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 asr9k-mpls-te-rsvp は親 RPM asr9k-mpls 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。

スクリプトを使用したゴールデン ISO の構築

表 3: 機能の履歴 (表)

機能名	リリース情報	説明
強化されたゴールデン ISO ビルドツール	リリース 7.5.1	この拡張機能により、gisobuild.py ツールを使用して、カスタマイズされたインストール要件に合わせて Cisco IOS XR ソフトウェアコマンド、YAML ベースのテンプレートファイル、または Docker 機能を使用して GISO イメージを柔軟に構築できます。GISO を作成する場合は、基本イメージとオプションの RPM を使用してルータを自動的にプロビジョニングすることに加えて、ゼロタッチプロビジョニング (ZTP) 初期化ファイル、スクリプト初期化ファイル、Cisco IOS XR 構成ファイル、および SMU の指定もできます。

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 (任意)
- ホスト、XR、およびシステム管理用の 1 つまたは複数のサードパーティ製 SMU (任意)
- ゴールデン ISO のラベル (任意)
- オプションの RPM
- ZTP 初期化 `ztp.ini` ファイル (任意)
- スクリプト初期化 `script.ini` ファイル (任意)

GISO スクリプトは XR 設定の検証をサポートしていません。



- (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]
```

Cisco IOS XR リリース 7.5.1 では、`gisobuild.py` GISO ビルドツールの拡張機能が導入されています。`ztp.ini` ZTP 初期化ファイルと `script.ini` スクリプト初期化ファイルも追加できます。ZTP 設定は、現在のソフトウェアバージョンが GISO イメージを使用するバージョンに置き換えられたとき、またはロールバックされたときにルータに適用され、ルータを自動的にプロビジョニングするために ZTP が実行されるたびに使用されます。このビルドツールでは複数のリポジトリがサポートされています。CLI コマンド、Docker、または YAML ファイルを使用して GISO を構築できます。



- (注)
- Cisco ASR 9000 シリーズルータの IOS XR 32 ビットソフトウェアから IOS XR 64 ビットソフトウェアに移行する場合は、`[migration]` オプションを `[true]` に設定します。
 - 最初の GISO を作成後に同じビルドディレクトリを使用する場合は、`[clean]` オプションを `[true]` に設定します。後続のすべての GISO ビルドでオプションを `[true]` に設定していることを確認します。
 - Docker を使用して GISO を構築する場合は、`[docker]` オプションを `[true]` に設定します。
 - GISO 構築時のエラーを回避するために、YAML ファイルの形式と構文が変更されていないことを確認します。たとえば、`:` 記号が欠落している場合、またはサポートされていない記号がテンプレートで使用されている場合、GISO の構築時にエラーが表示されます。

`gisobuild.py` ツールはネイティブに実行できます。また、Docker サービスが有効になっている、公開された Docker イメージをプルできるシステムで実行できます。追加の権限は不要なため、Docker を使用してイメージを構築することを推奨します。



- (注) [full-iso] オプションは、Cisco IOS XRv 9000 ルータに固有の完全な ISO イメージ `xrv9k-full-x-7.5.1.iso` を作成するために使用されます。Cisco IOS XR リリース 7.8.1 以降では、GISO の構築には完全な ISO イメージを使用できません。

GISO を作成するには、次の手順を実行します。

始める前に

- 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 3.6 以降が必要です。
 - システムには、最低 12 GB の空きディスク容量が必要です。
 - 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。ただし、`Docker` を使用している場合は、これらのユーティリティは不要です。
 - 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。
 - ユーザーは `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.py` スクリプトが実行されるシステムには、ルートログイン情報が必要ですが、`Docker` コンテナ内でイメージを構築する場合は必須ではありません。
- `gisobuild.py` スクリプトを使用する前に `git pull` 操作を実行して、Python バージョン用のスクリプトの最新バージョンを取得することを推奨します。

ステップ 1 スクリプト `gisobuild.py` を [GitHub](#) リポジトリから、GISO を構築するオフラインシステムまたは外部サーバーにコピーします。このシステムが上記の「始める前に」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または 1 つのリポジトリ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 64 です。

```
usage: gisobuild.py [-h] [--iso ISO] [--repo REPO [REPO ...]]
                  [--bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]]
                  [--xrconfig XRCONFIG] [--ztp-ini ZTP_INI] [--label LABEL]
                  [--out-directory OUT_DIRECTORY] [--yamlfile CLI_YAML] [--clean]
                  [--pkglist PKGLIST [PKGLIST ...]] [--script SCRIPT] [--docker]
                  [--x86-only] [--migration]
```

```

[--remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]]
[--skip-usb-image] [--copy-dir COPY_DIRECTORY]
[--clear-bridging-fixes] [--verbose-dep-check] [--debug]
[--version]

```

Utility to build Golden ISO for IOS-XR.

optional arguments:

```

-h, --help          show this help message and exit
--iso ISO           Path to Mini.iso/Full.iso file
--repo REPO [REPO ...]
                    Path to RPM repository. For LNT, user can specify .rpm, .tgz,
                    .tar filenames, or directories. RPMs are only used if already
                    included in the ISO, or specified by the user via the
                    --pkglist option.
--bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]
                    Bridging rpms to package. For EXR, takes from-release or rpm
                    names; for LNT, the user can specify the same file types as for
                    the --repo option.
--xrconfig XRCONFIG Path to XR config file
--ztp-ini ZTP_INI   Path to user ztp ini file
--label LABEL, -l LABEL
                    Golden ISO Label
--out-directory OUT_DIRECTORY
                    Output Directory
--yamlfile CLI_YAML Cli arguments via yaml
--clean            Delete output dir before proceeding
--pkglist PKGLIST [PKGLIST ...]
                    Packages to be added to the output GISO. For eXR: optional rpm
                    or smu to package. For LNT: either full package filenames or
                    package names for user installable packages can be specified.
                    Full package filenames can be specified to choose a particular
                    version of a package, the rest of the block that the package is
                    in will be included as well. Package names can be specified to
                    include optional packages in the output GISO.
--docker, --use-container
                    Build GISO in container environment. Pulls and run pre-built
                    container image to build GISO.
--version          Print version of this script and exit

```

EXR only build options:

```

--script SCRIPT    Path to user executable script executed as part of bootup post
                    activate.
--x86-only         Use only x86_64 rpms even if other architectures are
                    applicable.
--migration        To build Migration tar only for ASR9k

```

LNT only build options:

```

--remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]
                    Remove RPMs, specified in a comma separated list. These are
                    matched against user installable package names, and must be the
                    whole package name, e.g: xr-bgp
--skip-usb-image  Do not build the USB image
--copy-dir COPY_DIRECTORY
                    Copy built artefacts to specified directory if provided. The
                    specified directory must already exist, be writable by the
                    builder and must not contain a previously built artefact with
                    the same name.
--clear-bridging-fixes
                    Remove all bridging bugfixes from the input ISO
--verbose-dep-check
                    Verbose output for the dependency check.
--debug          Output debug logs to console

```

例

例 : Docker ベースの GISO イメージの構築

この例では、GISO イメージは Docker を使用して構築されます。

```
[root@xr src]# ./gisobuild.py --docker --iso /auto/ncs5500giso/ncs5500-mini-x-7.5.1.iso

--repo /auto/ncs5500giso --pkglist ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm ncs5500-isis-2.1.0.0-r751.x86_64.rpm
ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm
ncs5500-li-1.0.0.0-r751.x86_64.rpm ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
ncs5500-mppls-2.1.0.0-r751.x86_64.rpm ncs5500-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
ncs5500-parser-2.0.0.0-r751.x86_64.rpm --label dockerbasedgiso

Local System requirements check [PASS]
Pulling gisobuild image from hub. Please wait...
\
Done...
System requirements check [PASS]

Platform: ncs5500 Version: 7.5.1

Scanning repository [/auto/ncs5500giso]...

Building RPM Database...

Total 11 RPM(s) present in the repository path provided in CLI
[ 1] ncs5500-mppls-2.1.0.0-r751.x86_64.rpm
[ 2] ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
[ 3] ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
[ 4] ncs5500-parser-2.0.0.0-r751.x86_64.rpm
[ 5] ncs5500-isis-2.1.0.0-r751.x86_64.rpm
[ 6] ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
[ 7] ncs5500-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
[ 8] ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
[ 9] ncs5500-li-1.0.0.0-r751.x86_64.rpm
[10] ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
[11] ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

Following XR x86_64 rpm(s) will be used for building Golden ISO:

(+) ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
(+) ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
(+) ncs5500-parser-2.0.0.0-r751.x86_64.rpm
(+) ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
(+) ncs5500-li-1.0.0.0-r751.x86_64.rpm
(+) ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
(+) ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
(+) ncs5500-mppls-2.1.0.0-r751.x86_64.rpm
(+) ncs5500-mppls-te-rsvp-3.1.0.0-r751.x86_64.rpm
(+) ncs5500-isis-2.1.0.0-r751.x86_64.rpm
(+) ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

...RPM signature check [PASS]

Skipping following rpms from repository since they are already present in base ISO:

(-) ncs5500-parser-2.0.0.0-r751.x86_64.rpm
(-) ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
```

```

...RPM compatibility check [PASS]

Building Golden ISO...
Summary .....
```

XR rpms:

```

ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
ncs5500-isis-2.1.0.0-r751.x86_64.rpm
ncs5500-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
ncs5500-mpls-2.1.0.0-r751.x86_64.rpm
ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
ncs5500-li-1.0.0.0-r751.x86_64.rpm
ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm

...Golden ISO creation SUCCESS.
```

Golden ISO Image Location: /var/tmp/giso/gisobuild-toolkit-master/src/output_gisobuild/ncs5500-golden-x-7.5.1-dockerbasedgiso.iso

```

[root@xr src]# ./gisobuild.py --docker --iso /auto/asr9kgiso/asr9k-mini-x-7.5.1.iso
--repo /auto/asr9kgiso --pkglst asr9k-7.5.1.CSCsb88888 asr9k-bgp-2.0.0.0-r751.x86_64.rpm
asr9k-eigrp-1.0.0.0-r751.x86_64.rpm asr9k-isis-2.1.0.0-r751.x86_64.rpm
asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
asr9k-li-1.0.0.0-r751.x86_64.rpm asr9k-mcast-3.0.0.0-r751.x86_64.rpm
asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
asr9k-mpls-2.1.0.0-r751.x86_64.rpm asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
asr9k-ospf-2.0.0.0-r751.x86_64.rpm
asr9k-parser-2.0.0.0-r751.x86_64.rpm --label dockerbasedgiso

Local System requirements check [PASS]
Pulling gisobuild image from hub. Please wait...
\
Done...
System requirements check [PASS]

Platform: asr9000 Version: 7.5.1

Scanning repository [/auto/asr9000giso]...

Building RPM Database...

Total 11 RPM(s) present in the repository path provided in CLI
[ 1] asr9k-mpls-2.1.0.0-r751.x86_64.rpm
[ 2] asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
[ 3] asr9k-bgp-2.0.0.0-r751.x86_64.rpm
[ 4] asr9k-parser-2.0.0.0-r751.x86_64.rpm
[ 5] asr9k-isis-2.1.0.0-r751.x86_64.rpm
[ 6] asr9k-mcast-3.0.0.0-r751.x86_64.rpm
[ 7] asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
[ 8] asr9k-ospf-2.0.0.0-r751.x86_64.rpm
[ 9] asr9k-li-1.0.0.0-r751.x86_64.rpm
[10] asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
[11] asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
Following XR x86_64 rpm(s) will be used for building Golden ISO:

(+) asr9k-ospf-2.0.0.0-r751.x86_64.rpm
(+) asr9k-bgp-2.0.0.0-r751.x86_64.rpm
(+) asr9k-parser-2.0.0.0-r751.x86_64.rpm
(+) asr9k-mcast-3.0.0.0-r751.x86_64.rpm
(+) asr9k-li-1.0.0.0-r751.x86_64.rpm
```

```
(+) asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
(+) asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
(+) asr9k-mpls-2.1.0.0-r75114I.x86_64.rpm
(+) asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
(+) asr9k-isis-2.1.0.0-r751.x86_64.rpm
(+) asr9k-k9sec-3.1.0.0-r751.x86_64.rpm

...RPM signature check [PASS]

Skipping following rpms from repository since they are already present in base ISO:

(-) asr9k-parser-2.0.0.0-r751.x86_64.rpm
(-) asr9k-bgp-2.0.0.0-r751.x86_64.rpm

...RPM compatibility check [PASS]

Building Golden ISO...
Summary .....

XR rpms:
asr9k-mcast-3.0.0.0-r751.x86_64.rpm
asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
asr9k-isis-2.1.0.0-r751.x86_64.rpm
asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
asr9k-mpls-2.1.0.0-r751.x86_64.rpm
asr9k-ospf-2.0.0.0-r751.x86_64.rpm
asr9k-li-1.0.0.0-r751.x86_64.rpm
asr9k-k9sec-3.1.0.0-r751.x86_64.rpm

...Golden ISO creation SUCCESS.

Golden ISO Image Location: /var/tmp/giso/gisobuild-toolkit-master/src/output_gisobuild/
                           asr9k-golden-x-7.5.1-dockerbasedgiso.iso
```

GISO ファイルが正常に作成されたことを確認します。

```
[root@xr src]# ls
exrmod  gisobuild.py  lntmod  output_gisobuild  utils

[root@xr src]# cd output_gisobuild/
[root@xr output_gisobuild]# ls
img_built_name.txt  logs  ncs5500asr9k-golden-x-7.5.1-dockerbasedgiso.iso
rpms_packaged_in_giso.txt
```

例：YAML ベースの GISO イメージの構築

YAML は、パッケージリストを提供し、ビルドオプションを管理するためのテンプレートとして機能するマークアップファイルです。

次に、サンプルの YAML テンプレートの例を示します。

```
# Options below correspond to the tool input options.
# --iso ISO                Path to Mini.iso/golden.iso file
# --repo REPO [REPO ...]
#
#                          Path to list of RPM repositories. RPMs are only used if already
#                          included in the ISO, or specified by the user via the --pkglist
#                          option.
# --pkglist PKGLIST [PKGLIST ...]
#
#                          Optional list of rpm or smu to add to the ISO.
# --remove-packages REMOVE_PACKAGES [REMOVE_PACKAGES ...]
#
#                          Remove named RPMs, specified in a space separated list. Valid
#                          build
#
#                          option for LNT only. eXR builds simply ignores this option.
```

```

# --bridging-fixes BRIDGE_FIXES [BRIDGE_FIXES ...]
#                               Bridging rpms to package. Takes from-release (supported for eXR)
#                               or rpm names.
# --xrconfig XRCONFIG           Path to XR config file
# --ztp-ini ZTP_INI             Path to user ztp ini file
# --script SCRIPT               Path to user executable script executed as part of
#                               bootup post activate. Valid build option for eXR only.
#                               LNT builds simply ignores.
# --label LABEL                 Golden ISO Label
# --out-directory OUT_DIRECTORY
#                               Output Directory. Built GISO and logs will be available post
gisobuild.
# --copy-directory COPY_DIRECTORY
#                               Copy built artefacts to specified directory if provided. Valid
build
#                               option for LNT only. eXR build ignores this option.
# --yamlfile CLI_YAML           Cli arguments via yaml.
# --clean                       Delete output dir before proceeding.
# --migration                   To build Migration tar only for ASR9k. Valid build option for
eXR only.
#                               LNT builds simply ignore this option.
# --docker                      Load and run pre-built docker image. Valid build option for eXR
only.
#                               LNT builds simply ignore this option.
# --x86-only                    Use only x86_64 rpms even if other architectures are applicable.
Valid build
#                               option for eXR only. LNT builds simply ignore this option.
# --version                    Print version of this script and exit

packages:
  iso: <path-to-iso>
  repo:
    - <path-to-repo1>
    - <path-to-repo2>
  pkglist:
    - <pkg1>
    - <pkg2>
  bridge-fixes:
    upgrade-from-release:
      - <dotted-release-1>
      - <dotted-release-2>
    rpms:
      - <pkg1>
      - <pkg2>
  remove_packages:
    - <pkg1>
    - <pkg2>

user-content:
  script: <path-to-script-sh>
  xrconfig: <path-to-router.cfg>
  ztp-ini: <path-to-ztp.ini>

output:
  label: <giso-label>
  out-directory: <path-to-output-directory>
  clean: <true/false>

options:
  docker: <true/false>
  migration: <true/false>
  x86-only: <true/false>

```

この例では、必要なファイルを指定して YAML ファイルを設定します。

```

packages:
  iso: /auto/751_repo/ncs5500-mini-x-7.5.1.iso
  repo:
    - /auto/751_repo/
  pkglist:
    - ncs5500-bgp-2.0.0.0-r751.x86_64.rpm
    - ncs5500-eigrp-1.0.0.0-r751.x86_64.rpm
    - ncs5500-isis-2.1.0.0-r751.x86_64.rpm
    - ncs5500-li-1.0.0.0-r751.x86_64.rpm
    - ncs5500-mcast-3.0.0.0-r751.x86_64.rpm
    - ncs5500-mgbl-3.0.0.0-r751.x86_64.rpm
    - ncs5500-mpls-2.1.0.0-r751.x86_64.rpm
    - ncs5500-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
    - ncs5500-ospf-2.0.0.0-r751.x86_64.rpm
    - ncs5500-parser-2.0.0.0-r751.x86_64.rpm
    - ncs5500-k9sec-3.1.0.0-r751.x86_64.rpm
    - ncs5500-mcast-3.0.0.1-r751.CSCxr33333.x86_64.rpm
    - ncs5500-os-5.0.0.1-r751.CSCxr11111.x86_64.rpm
    - ncs5500-sysadmin-hostos-7.5.1-r751.CSCho99999.admin.x86_64.rpm
    - ncs5500-sysadmin-hostos-7.5.1-r751.CSCho99999.host.x86_64.rpm
    - ncs5500-sysadmin-topo-7.5.1-r751.CSCcv55555.x86_64.rpm
    - ncs5500-sysadmin-system-7.5.1-r751.CSCcv44444.x86_64.rpm
    - openssh-scp-6.6p1.pl-r0.5.0.r751.CSCTp11111.xr.x86_64.rpm
    - cisco-klm-zermatt-0.1.pl-r0.0.r751.CSCTp11111.xr.x86_64.rpm

  remove_rpms: []

user-content:
  script: script.sh
  xrconfig: /auto/751_repo/gisoxrconfig.cfg
  ztp-ini: /auto/751_repo/ztp.ini

output:
  label: 751_yaml_install
  out-directory: /auto/751_repo/
  clean: true

options:
  docker: false
  full-iso: false
  migration: false
  x86-only: false

packages:
  iso: /auto/751_repo/asr9k-mini-x-7.5.1.iso
  repo:
    - /auto/751_repo/
  pkglist:
    - asr9k-bgp-2.0.0.0-r751.x86_64.rpm
    - asr9k-eigrp-1.0.0.0-r751.x86_64.rpm
    - asr9k-isis-2.1.0.0-r751.x86_64.rpm
    - asr9k-li-1.0.0.0-r751.x86_64.rpm
    - asr9k-mcast-3.0.0.0-r751.x86_64.rpm
    - asr9k-mgbl-3.0.0.0-r751.x86_64.rpm
    - asr9k-mpls-2.1.0.0-r751.x86_64.rpm
    - asr9k-mpls-te-rsvp-3.1.0.0-r751.x86_64.rpm
    - asr9k-ospf-2.0.0.0-r751.x86_64.rpm
    - asr9k-parser-2.0.0.0-r751.x86_64.rpm
    - asr9k-k9sec-3.1.0.0-r751.x86_64.rpm
    - asr9k-mcast-3.0.0.1-r751.CSCxr33333.x86_64.rpm
    - asr9k-os-5.0.0.1-r751.CSCxr11111.x86_64.rpm
    - asr9k-sysadmin-hostos-7.5.1-r751.CSCho99999.admin.x86_64.rpm
    - asr9k-sysadmin-hostos-7.5.1-r751.CSCho99999.host.x86_64.rpm
    - asr9k-sysadmin-topo-7.5.1-r751.CSCcv55555.x86_64.rpm
    - asr9k-sysadmin-system-7.5.1-r751.CSCcv44444.x86_64.rpm

```

```

- openssl-scp-6.6p1.p1-r0.5.0.r751.CSCTp11111.xr.x86_64.rpm
- cisco-klm-zermatt-0.1.p1-r0.0.r751.CSCTp11111.xr.x86_64.rpm

remove_rpms: []

user-content:
  script: script.sh
  xrconfig: /auto/751_repo/gisoxrconfig.cfg
  ztp-ini: /auto/751_repo/ztp.ini

output:
  label: 751_yaml_install
  out-directory: /auto/751_repo/
  clean: true

options:
  docker: false
  full-iso: false
  migration: false
  x86-only: false

```

CLI を使用してパッケージとパラメータのリストを指定しない場合は、YAML ファイルテンプレートを 사용할 ことができます。

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg>
```

YAML 構成ファイルの入力をオーバーライドするには、対応する CLI オプションを使用します。

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg> --label <new-label>
```

この新しいラベルで、YAML ファイルで指定されたラベルがオーバーライドされます。

ホストマシンでパッケージの依存関係が満たされていないが、Docker イメージのプルと実行が許可されている場合は、YAML ファイルの [docker] オプションを [true] に設定して、次のコマンドを実行します。

```
[directory-path]$ ./src/gisobuild.py --yamlfile <input-yaml-cfg>
```

ここで、input-yaml-cfg の [docker] オプションは [true] に設定されています。

次のタスク

ルータに GISO イメージを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 install 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。 DHCPClient 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。

PXE ブートモードを使用してルータを起動すると、既存の設定が削除されます。 パーマネントライセンス予約 (PLR) などのスマートライセンス設定を回復するには、ルータの起動後にこれらの設定を有効にします。

```
Router#configure
Router(config)#license smart reservation
Router(config)#commit
```

次に、PXE ブートを使用した GISO のインストールのログの例を示します。

```
...
Fri Dec 2 19:18:03 UTC 2016: ---Starting to prepare host logical volume---
...
Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.host.x86_64.rpm) from
installation
Fri Dec 02 19:18:14 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.host.x86_64.rpm)
from installation
Fri Dec 02 19:18:15 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.host.x86_64
[SUCCESS]
...
Fri Dec 2 19:18:23 UTC 2016: ---Starting to prepare calvados logical volume---
...
Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(openssh-scp-6.6p1-r0.0.admin.x86_64.rpm) from
installation
Fri Dec 02 19:18:48 UTC 2016: Skipping tp base rpm(kernel-modules-3.14-r0.1.admin.x86_64.rpm)
from installation
Fri Dec 02 19:18:49 UTC 2016: Installing asr9k-sysadmin-system-6.1.3-r613.CSCcv44444.x86_64
[SUCCESS]
Fri Dec 02 19:18:50 UTC 2016: Installing asr9k-sysadmin-shared-6.1.3-r613.CSCcv33333.x86_64
[SUCCESS]
Fri Dec 02 19:18:51 UTC 2016: Installing asr9k-sysadmin-hostos-6.1.3-r613.CSChu77777.admin.x86_64
[SUCCESS]
...
Fri Dec 2 19:19:07 UTC 2016: ---Starting to prepare repository---
Fri Dec 2 19:19:11 UTC 2016: File system creation on /test took 3 seconds
Fri Dec 2 19:19:11 UTC 2016: Copying /iso/host.iso to repository /iso directory
Fri Dec 2 19:19:11 UTC 2016: Copy Host rpms to repository
Fri Dec 2 19:19:13 UTC 2016: Copying /iso/asr9k-sysadmin.iso to repository /iso directory
Fri Dec 2 19:19:13 UTC 2016: Copy Sysadmin rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy HostOs rpms to repository
Fri Dec 2 19:19:16 UTC 2016: Copy XR rpms to repository
```

```
Fri Dec 2 19:19:16 UTC 2016: Copy giso_info.txt to repository
Fri Dec 2 19:19:17 UTC 2016: Copying /iso/asr9k-xr.iso to repository /iso directory
Fri Dec 2 19:19:21 UTC 2016: Copying all ISOs to repository took 10 seconds
```

...

- **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。
- **システムアップグレード** : システムアップグレード時には、**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。

重要 ルータ上の現在のバージョンとパッケージを GISO のバージョンと置き換えるには、コマンドとフォーマットの変更に注意してください。

- Cisco IOS XR リリース 6.3.3、6.4. x、および 6.5.1 よりも前のバージョンでは、次の **install update** コマンドを使用します。

```
install update source <source path> <Golden-ISO-name> replace
```

- Cisco IOS XR リリース 6.5.2 以降では、**install replace** コマンドを使用します。

```
install replace <absolute-path-of-Golden-ISO>
```

(注) ブート可能な外部 USB ディスクを作成するには、次の手順を実行します。

- USB ブートディスクに 8GB の最小ストレージがあること、および Linux マシンでブート可能なディスクを作成するための root/admin 権限または適切な権限があることを確認します。

1. Linux マシンで `usb-install` スクリプトをコピーして実行し、ブート可能な外部 USB を作成します。

```
Router#admin

sysadmin-vm:0_RSP0#run chvrf 0 ssh rp0_admin
[sysadmin-vm:0_RSP0:~]$ssh my_host
[host:~]$cd /misc/disk1/
[host:~]$. /usb-install-712-or-latest.sh asr9k-goldenk9-x64-7.0.2-dr.iso /dev/sdc
EFI

Preparing USB stick for EFI
parted gpt: Failed to create partition - continuing ...
Create filesystem on /dev/sdc1
Mounting source iso at //misc/disk1/cdtmp.CnuKnA
Mounting destination /dev/sdc1 at //misc/disk1/usbdev.SSBb4R
Copying image to USB stick
Initrd path is //misc/disk1/cdtmp.CnuKnA/boot/initrd.img
Getting boot
3749342 blocks
Copying boot
Copying initrd.img
Copying signature.initrd.img
Copying certs
Creating grub files
Copying /misc/disk1/asr9k-goldenk9-x64-7.0.2-dr.iso in USB Stick
USB stick set up for EFI boot!
```

2. RSP/RP をリセットし、ブート可能な USB を RSP/RP の前面パネルに接続します。USB は ROMMON で検出されます。システムが ROMMON で起動している場合、RSP/RP がリセットされるまで、前面パネルの外部 USB を追加しても USB は検出されません。

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 source` 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。

`install replace` コマンドを使用すると、ISO と既存のバージョン間の違いに関係なく、システムのリロードが実行されます。

- 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。

```
sysadmin-vm:0_RP0#show install repository all
Admin repository
-----
asr9k-sysadmin-6.1.1

asr9k-sysadmin-hostos-6.1.1-r611.CSCcv10001.admin.x86_64
asr9k-sysadmin-system-6.1.1-r611.CSCcv10005.x86_64
....

XR repository
-----
asr9k-iosxr-mgbl-3.0.0.0-r611.x86_64
asr9k-xr-6.1.1
....

Host repository
-----
host-6.1.1
```

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。

- (注) GISO の RPM を一覧表示するには、GISO がインストールリポジトリに存在する必要があります。

```
Router#show install package asr9k-goldenk9-x64-6.1.3

Sun Dec 4 13:52:48.279 UTC
This may take a while ...
ISO Name: asr9k-goldenk9-x64-6.1.3
ISO Type: bundle
ISO Bundled: asr9k-mini-x64-6.1.3
Golden ISO Label: temp
ISO Contents:
ISO Name: asr9k-xr-6.1.3
ISO Type: xr
rpms in xr ISO:
iosxr-os-asr9k-64-5.0.0.0-r613
iosxr-ce-asr9k-64-3.0.0.0-r613
iosxr-infra-asr9k-64-4.0.0.0-r613
iosxr-fwding-asr9k-64-4.0.0.0-r613
iosxr-routing-asr9k-64-3.1.0.0-r613
...

ISO Name: asr9k-sysadmin-6.1.3
ISO Type: sysadmin
rpms in sysadmin ISO:
asr9k-sysadmin-topo-6.1.3-r613
asr9k-sysadmin-shared-6.1.3-r613
asr9k-sysadmin-system-6.1.3-r613
asr9k-sysadmin-hostos-6.1.3-r613.admin
...
```

```

ISO Name: host-6.1.3
ISO Type: host
rpms in host ISO:
  asr9k-sysadmin-hostos-6.1.3-r613.host

Golden ISO Rpms:
  xr rpms in golden ISO:
    asr9k-k9sec-x64-2.2.0.1-r613.CSCxr33333.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.xr.x86_64.rpm
    openssh-scp-6.6p1-r0.0.xr.x86_64.rpm
    asr9k-mp1s-x64-2.1.0.0-r613.x86_64.rpm
    asr9k-k9sec-x64-2.2.0.0-r613.x86_64.rpm

  sysadmin rpms in golden ISO:
    asr9k-sysadmin-system-6.1.3-r613.CSCcv11111.x86_64.rpm
    openssh-scp-6.6p1-r0.0.admin.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.admin.x86_64.rpm

  host rpms in golden ISO:
    openssh-scp-6.6p1-r0.0.host.x86_64.rpm
    openssh-scp-6.6p1.p1-r0.0.CSCtp12345.host.x86_64.rpm
  
```

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。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。