



bash

- [Bash について \(1 ページ\)](#)
- [注意事項と制約事項 \(1 ページ\)](#)
- [Bash へのアクセス \(2 ページ\)](#)
- [権限をルートにエスカレーションする \(3 ページ\)](#)
- [Bash コマンドの例 \(5 ページ\)](#)
- [機能 RPM の管理 \(6 ページ\)](#)
- [パッチ RPM の管理 \(9 ページ\)](#)
- [SDK または ISO で構築されたサードパーティプロセスの永続的なデーモン化 \(16 ページ\)](#)
- [ネイティブ Bash シェルからのアプリケーションの永続的な起動 \(17 ページ\)](#)
- [ネイティブ Bash シェルのアプリケーション例 \(18 ページ\)](#)

Bash について

Cisco NX-OS CLIに加えて、Cisco Nexus 9000 シリーズ スイッチは Bourne-Again SHell (Bash) へのアクセスをサポートします。Bash は、ユーザーが入力したコマンドまたはシェル スクリプトから読み取られたコマンドを解釈します。Bash を使用すると、デバイス上の基盤となる Linux システムにアクセスしてシステムを管理できます。

注意事項と制約事項

Bash シェルには、次の注意事項と制約事項があります。

- NX-OS 7.0 (3) F3 (1) 以降、Bash シェルは Cisco Nexus 9508 スイッチでサポートされません。
- /isan フォルダにあるバイナリは、**run bash** コマンドから入力されたシェルとは異なる設定の環境で実行するためのものです。Bash シェルからこれらのバイナリを使用しないことをお勧めします。その環境内での動作は予測できないからです。

- Cisco Python モジュールをインポートする場合は、Bash シェルから Python を使用しないでください。代わりに NX-OS VSH 内で、より新しい Python を使用します。

Bash へのアクセス

Cisco NX-OS では、Cisco NX-OS dev-ops ロールまたは Cisco NX-OS network-admin ロールに関連付けられたユーザアカウントから Bash にアクセスできます。

次の例は、dev-ops ロールと network-admin ロールの権限を示しています。

```
switch# show role name dev-ops

Role: dev-ops
Description: Predefined system role for devops access. This role
cannot be modified.
Vlan policy: permit (default)
Interface policy: permit (default)
Vrf policy: permit (default)
-----
Rule      Perm   Type      Scope      Entity
-----
4         permit command   conf t ; username *
3         permit command   bcm module *
2         permit command   run bash *
1         permit command   python *

switch# show role name network-admin

Role: network-admin
Description: Predefined network admin role has access to all commands
on the switch
-----
Rule      Perm   Type      Scope      Entity
-----
1         permit read-write

switch#
```

feature bash-shell コマンドを実行すると、Bash が有効になります。

この **run bash** コマンドは Bash を読み込み、ユーザーのホームディレクトリから開始します。

次の例は、Bash シェル機能を有効にする方法と、Bash を実行する方法を示しています。

```
switch# configure terminal
switch(config)# feature bash-shell

switch# run?
run          Execute/run program
run-script   Run shell scripts

switch# run bash?
bash        Linux-bash

switch# run bash
bash-4.2$ whoami
admin
bash-4.2$ pwd
```

```
/bootflash/home/admin
bash-4.2$
```



(注) **run bash** コマンドで Bash コマンドを実行することもできます。

たとえば、**run bash** コマンドを使用して **whoami** を実行することもできます。

```
run bash whoami
```

ユーザー **shelltype** を構成して Bash を実行することもできます。

```
username foo shelltype bash
```

このコマンドにより、ログイン時に Bash シェルを直接実行できるようになります。この場合、**feature bash-shell** を有効にする必要はありません。

権限をルートにエスカレーションする

管理者ユーザーの特権は、ルートアクセスの特権をエスカレーションできます。

以下は、権限をエスカレーションするためのガイドラインです：

- 管理者権限ユーザー (**network-admin/vdc-admin**) は、NX-OS における、Linux の **root** 権限ユーザーに相当します。
- 認証された管理者ユーザーのみが権限を **root** に昇格できます。認証された管理者権限ユーザーにパスワードは必要ありません。
- 権限をエスカレーションする前に、**Bash** を有効にする必要があります。
- 非管理インターフェイスを介した **root** ユーザー名を使用したスイッチへの SSH では、**root** ユーザーの Linux Bash シェルタイプアクセスがデフォルトになります。NX-OS シェルアクセスに戻るために **vsh** を入力します。

NX-OS ネットワーク管理者ユーザーは、次の場合に **root** にエスカレーションして、構成コマンドを NX-OS VSH に渡す必要があります。

- NX-OS ユーザーはシェルタイプの **Bash** を使用しており、シェルタイプの **Bash** を使用してスイッチにログインしています。
- **Bash** でスイッチにログインした NX-OS ユーザーは、引き続きスイッチで **Bash** を使用します。

sudo su 'vsh -c "<configuration commands>" または **sudo bash -c 'vsh -c "<configuration commands>"** を実行します。

次の例は、デフォルトのシェルタイプが **Bash** であるネットワーク管理者ユーザー **MyUser** が、**sudo** を使用して構成コマンドを NX-OS に渡す方法を示しています。

```
ssh -l MyUser 1.2.3.4
-bash-4.2$ sudo vsh -c "configure terminal ; interface eth1/2 ; shutdown ; sleep 2 ;
show interface eth1/2 brief"
```

```

-----
Ethernet      VLAN      Type Mode      Status Reason                               Speed      Port
Interface                                           Ch #
-----
Eth1/2        --        eth  routed down  Administratively down                auto(D)  --

```

次の例は、デフォルトのシェルタイプがBashであるネットワーク管理者ユーザー MyUserが、NX-OS に入り、NX-OS で Bash を実行する方法を示しています。

```

ssh -l MyUser 1.2.3.4
-bash-4.2$ vsh -h
Cisco NX-OS Software
Copyright (c) 2002-2016, Cisco Systems, Inc. All rights reserved.
Nexus 9000v software ("Nexus 9000v Software") and related documentation,
files or other reference materials ("Documentation") are
the proprietary property and confidential information of Cisco
Systems, Inc. ("Cisco") and are protected, without limitation,
pursuant to United States and International copyright and trademark
laws in the applicable jurisdiction which provide civil and criminal
penalties for copying or distribution without Cisco's authorization.

Any use or disclosure, in whole or in part, of the Nexus 9000v Software
or Documentation to any third party for any purposes is expressly
prohibited except as otherwise authorized by Cisco in writing.
The copyrights to certain works contained herein are owned by other
third parties and are used and distributed under license. Some parts
of this software may be covered under the GNU Public License or the
GNU Lesser General Public License. A copy of each such license is
available at
http://www.gnu.org/licenses/gpl.html and
http://www.gnu.org/licenses/lgpl.html
*****
* Nexus 9000v is strictly limited to use for evaluation, demonstration      *
* and NX-OS education. Any use or disclosure, in whole or in part of        *
* the Nexus 9000v Software or Documentation to any third party for any      *
* purposes is expressly prohibited except as otherwise authorized by        *
* Cisco in writing.                                                           *
*****
switch# run bash
bash-4.2$ vsh -c "configure terminal ; interface eth1/2 ; shutdown ; sleep 2 ; show
interface eth1/2 brief"

```

```

-----
Ethernet      VLAN      Type Mode      Status Reason                               Speed      Port
Interface                                           Ch #
-----
Eth1/2        --        eth  routed down  Administratively down                auto(D)  --

```



(注) **sudo su -** は使用しないでください。使用すると、システムがハングします。

次の例は、特権を root にエスカレーションする方法と、エスカレーションを確認する方法を表示しています。

```

switch# run bash
bash-4.2$ sudo su root
bash-4.2# whoami
root
bash-4.2# exit

```

```
exit
```

Bash コマンドの例

このセクションには、Bash コマンドと出力の例が含まれています。

システム統計情報の表示

次の例は、システム統計情報の表示方法を示しています：

```
switch# run bash
bash-4.2$ cat /proc/meminfo
<snip>
MemTotal:      16402560 kB
MemFree:       14098136 kB
Buffers:       11492 kB
Cached:        1287880 kB
SwapCached:    0 kB
Active:        1109448 kB
Inactive:      717036 kB
Active(anon):  817856 kB
Inactive(anon): 702880 kB
Active(file):  291592 kB
Inactive(file): 14156 kB
Unevictable:   0 kB
Mlocked:       0 kB
SwapTotal:     0 kB
SwapFree:      0 kB
Dirty:         32 kB
Writeback:     0 kB
AnonPages:    527088 kB
Mapped:        97832 kB
<\snip>
```

CLI からの Bash の実行

次に、`run bash` コマンドを使用して Bash から `ps` を実行する例を示します。

```
switch# run bash ps -el
F S  UID  PID  PPID  C  PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0    1    0  0  80   0  -   528 poll_s ?           00:00:03 init
1 S   0    2    0  0  80   0  -     0 kthrea ?           00:00:00 kthreadd
1 S   0    3    2  0  80   0  -     0 run_ks ?           00:00:56 ksoftirqd/0
1 S   0    6    2  0 -40  -  -     0 cpu_st ?           00:00:00 migration/0
1 S   0    7    2  0 -40  -  -     0 watchd ?           00:00:00 watchdog/0
1 S   0    8    2  0 -40  -  -     0 cpu_st ?           00:00:00 migration/1
1 S   0    9    2  0  80   0  -     0 worker ?           00:00:00 kworker/1:0
1 S   0   10    2  0  80   0  -     0 run_ks ?           00:00:00 ksoftirqd/1
```

機能 RPM の管理

RPM インストールの前提条件

RPM をインストールまたは追加する前に、次の手順によりシステムの準備ができていることを確認します。

手順

	コマンドまたはアクション	目的
ステップ 1	switch# <code>show logging logfile grep -i "System ready"</code>	Bash を実行する前に、この手順によって、RPM をインストールまたは追加する前のシステムの準備ができていることを確認します。 以下のような出力が表示されれば、続行します。 2018 Mar 27 17:24:22 switch %ASCII-CFG-2-CONF_CONTROL: System ready
ステップ 2	switch# <code>run bash sudo su</code> 例： switch# <code>run bash sudo su</code> bash-4.2#	Bash をロードします。

Bash からの機能 RPM のインストール

手順

	コマンドまたはアクション	目的
ステップ 1	switch# <code>sudo yum installed grep platform</code>	スイッチにインストールされている NX-OS 機能 RPM のリストを表示します。
ステップ 2	switch# <code>yum list available</code>	使用可能な RPM のリストを表示します。
ステップ 3	switch# <code>sudo yum -y install rpm</code>	使用可能な RPM をインストールします。

例

次に、bfd RPM をインストールする例を示します。

```
bash-4.2$ yum list installed | grep n9000
base-files.n9000                3.0.14-r74.2                installed
bfd.lib32_n9000                1.0.0-r0                    installed
core.lib32_n9000              1.0.0-r0                    installed
eigrp.lib32_n9000             1.0.0-r0                    installed
eth.lib32_n9000               1.0.0-r0                    installed
isis.lib32_n9000              1.0.0-r0                    installed
lACP.lib32_n9000              1.0.0-r0                    installed
linecard.lib32_n9000          1.0.0-r0                    installed
lldp.lib32_n9000              1.0.0-r0                    installed
ntp.lib32_n9000               1.0.0-r0                    installed
nxos-ssh.lib32_n9000          1.0.0-r0                    installed
ospf.lib32_n9000              1.0.0-r0                    installed
perf-cisco.n9000_gdb          3.12-r0                     installed
platform.lib32_n9000          1.0.0-r0                    installed
shadow-securetty.n9000_gdb    4.1.4.3-r1                  installed
snmp.lib32_n9000              1.0.0-r0                    installed
svi.lib32_n9000               1.0.0-r0                    installed
sysvinit-inittab.n9000_gdb    2.88dsf-r14                 installed
tacacs.lib32_n9000            1.0.0-r0                    installed
task-nxos-base.n9000_gdb      1.0-r0                      installed
tor.lib32_n9000               1.0.0-r0                    installed
vtp.lib32_n9000               1.0.0-r0                    installed
bash-4.2$ yum list available
bgp.lib32_n9000                1.0.0-r0
bash-4.2$ sudo yum -y install bfd
```



- (注) 起動時のスイッチのリロード時に、永続的な RPM のため、yum の代わりに rpm コマンドを使用します。そうしなかった場合、最初に yum bash または install cli を使用してインストールされた RPM は、インストール済みとしてではなく、リポジトリ名またはファイル名で表示されます。

機能 RPM のアップグレード

始める前に

yum リポジトリに RPM の上位バージョンが存在する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	<code>sudo yum -y upgrade rpm</code>	インストールされている RPM をアップグレードします。

例

次に、**bfd** RPM のアップグレードの例を示します。

```
bash-4.2$ sudo yum -y upgrade bfd
```

機能 RPM のダウングレード

手順

	コマンドまたはアクション	目的
ステップ 1	<code>sudo yum -y downgrade rpm</code>	いずれかの dnf リポジトリに下位バージョンの RPM がある場合に、RPM をダウングレードします。

例

次に、**bfd** RPM をダウングレードする例を示します。

```
bash-4.2$ sudo yum -y downgrade bfd
```

機能 RPM の消去



(注) SNMP RPM および NTP RPM は保護されており、消去できません。

これらの RPM をアップグレードまたはダウングレードすることはできます。アップグレードまたはダウングレードを有効にするには、システムのリロードが必要です。

保護された RPM のリストについては、`/etc/yum/protected.d/protected_pkgs.conf` を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	<code>sudo yum -y erase rpm</code>	RPM を消去します。

例

次の例は、**bfd** RPM を消去する方法を示しています。

```
bash-4.2$ sudo yum -y erase bfd
```


パッチ RPM の管理

RPM インストールの前提条件

RPM をインストールまたは追加する前に、次の手順によりシステムの準備ができていないことを確認します。

手順

	コマンドまたはアクション	目的
ステップ 1	switch# <code>show logging logfile grep -i "System ready"</code>	Bash を実行する前に、この手順によって、RPM をインストールまたは追加する前のシステムの準備ができていないことを確認します。 以下のような出力が表示されれば、続行します。 2018 Mar 27 17:24:22 switch %ASCII-CFG-2-CONF_CONTROL: System ready
ステップ 2	switch# <code>run bash sudo su</code> 例： switch# <code>run bash sudo su</code> bash-4.2#	Bash をロードします。

Bash からのパッチ RPM の追加

手順

	コマンドまたはアクション	目的
ステップ 1	switch# <code>yum list --patch-only</code>	スイッチに存在するパッチ RPM のリストを表示します。
ステップ 2	switch# <code>sudo yum install --add URL_of_patch</code>	リポジトリにパッチを追加します。ここで <code>URL_of_patch</code> は、標準的な Linux 形式の <code>/bootflash/patch</code> などではなく、 <code>bootflash:/patch</code> などの明確に定義された形式です。

	コマンドまたはアクション	目的
ステップ 3	yum list --patch-only available	リポジトリに追加されているものの非アクティブ状態のパッチのリストを表示します。

例

次に、**nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000** RPM をインストールする例を示します。

```
bash-4.2# yum list --patch-only
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
groups-repo                | 1.1 kB    00:00 ...
localdb                    | 951 B    00:00 ...
patching                   | 951 B    00:00 ...
thirdparty                 | 951 B    00:00 ...
bash-4.2#
bash-4.2# sudo yum install --add
bootflash:/nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000.rpm
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
groups-repo                | 1.1 kB    00:00 ...
localdb                    | 951 B    00:00 ...
patching                   | 951 B    00:00 ...
thirdparty                 | 951 B    00:00 ...
[#####] 70%Install operation 135 completed successfully at Tue Mar 27
17:45:34 2018.

[#####] 100%
bash-4.2#
```

パッチ RPM がインストールされたら、正しくインストールされたことを確認します。次のコマンドは、リポジトリに追加され、非アクティブ状態のパッチのリストを表示します。

```
bash-4.2# yum list --patch-only available
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
groups-repo                | 1.1 kB    00:00 ...
localdb                    | 951 B    00:00 ...
patching                   | 951 B    00:00 ...
thirdparty                 | 951 B    00:00 ...
nxos.CSCab00001-n9k_ALL.lib32_n9000  1.0.0-7.0.3.I7.3  patching
bash-4.2#
```

RPM がバンドルされている tar ファイルを使用して、tar ファイルからリポジトリにパッチを追加することもできます。次に、**nxos.CSCab00002_CSCab00003-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000** tar ファイルに含まれる 2 つの RPM をパッチ リポジトリに追加する例を示します。

```
bash-4.2# sudo yum install --add
bootflash:/nxos.CSCab00002_CSCab00003-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000.tar
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
```

```

groups-repo | 1.1 kB 00:00 ...
localdb | 951 B 00:00 ...
patching | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
[##### ] 70%Install operation 146 completed successfully at Tue Mar 27
21:17:39 2018.

[#####] 100%
bash-4.2#
bash-4.2# yum list --patch-only
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
groups-repo | 1.1 kB 00:00 ...
localdb | 951 B 00:00 ...
patching | 951 B 00:00 ...
patching/primary | 942 B 00:00 ...
patching | 2/2
thirdparty | 951 B 00:00 ...
nxos.CSCab00003-n9k_ALL.lib32_n9000 1.0.0-7.0.3.I7.3 patching
nxos.CSCab00002-n9k_ALL.lib32_n9000 1.0.0-7.0.3.I7.3 patching
bash-4.2#

```

パッチ RPM のアクティブ化

始める前に

[Bash からのパッチ RPM の追加 \(9 ページ\)](#) の手順に従って、必要なパッチ RPM がリポジトリに追加されていることを確認します。

手順

	コマンドまたはアクション	目的
ステップ 1	<code>sudo yum install patch_RPM --nocommit</code>	<p>パッチ RPM をアクティブにします。ここで、<code>patch_RPM</code> はリポジトリにあるパッチです。この手順では、パッチの場所を指定しないでください。</p> <p>(注) <code>--nocommit</code> フラグをコマンドに追加すると、パッチ RPM がこの手順でアクティブになりますが、コミットされません。パッチ RPM をアクティブ化した後にコミットする手順については、パッチ RPM のコミット (13 ページ) を参照してください。</p>

例

次に、**nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000** パッチ RPM をアクティブにする例を示します。

```
bash-4.2# sudo yum install nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000 --nocommit
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
groups-repo                               | 1.1 kB      00:00 ...
localdb                                    | 951 B       00:00 ...
patching                                   | 951 B       00:00 ...
thirdparty                                | 951 B       00:00 ...
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package nxos.CSCab00001-n9k_ALL.lib32_n9000 0:1.0.0-7.0.3.I7.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch            Version          Repository      Size
=====
Installing:
nxos.CSCab00001-n9k_ALL                lib32_n9000     1.0.0-7.0.3.I7.3  patching        28 k

Transaction Summary
=====
Install      1 Package

Total download size: 28 k
Installed size: 82 k
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000      1/1
[##### ] 90%error: reading
/var/sysmgr/tmp/patches/CSCab00001-n9k_ALL/isan/bin/sysinfo manifest, non-printable
characters found

Installed:
  nxos.CSCab00001-n9k_ALL.lib32_n9000 0:1.0.0-7.0.3.I7.3

Complete!
Install operation 140 completed successfully at Tue Mar 27 18:07:40 2018.

[#####] 100%
bash-4.2#
```

次のコマンドを入力して、パッチ RPM が正常にアクティブ化されたことを確認します。

```
bash-4.2# yum list --patch-only
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
groups-repo                               | 1.1 kB      00:00 ...
localdb                                    | 951 B       00:00 ...
patching                                   | 951 B       00:00 ...
```

```
thirdparty | 951 B 00:00 ...
nxos.CSCab00001-n9k_ALL.lib32_n9000 1.0.0-7.0.3.I7.3 installed
bash-4.2#
```

パッチ RPM のコミット

手順

	コマンドまたはアクション	目的
ステップ 1	<code>sudo yum install patch_RPM --commit</code>	パッチ RPM をコミットします。パッチ RPM は、リロード後もアクティブな状態を維持するためにコミットする必要があります。

例

次に、`nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000` パッチ RPM をコミットする例を示します。

```
bash-4.2# sudo yum install nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000 --commit

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
                : protect-packages
groups-repo | 1.1 kB 00:00 ...
localdb | 951 B 00:00 ...
patching | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
Install operation 142 completed successfully at Tue Mar 27 18:13:16 2018.

[#####] 100%
bash-4.2#
```

次のコマンドを入力して、パッチ RPM が正常にコミットされたことを確認します。

```
bash-4.2# yum list --patch-only committed
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
                : protect-packages
groups-repo | 1.1 kB 00:00 ...
localdb | 951 B 00:00 ...
patching | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
nxos.CSCab00001-n9k_ALL.lib32_n9000 1.0.0-7.0.3.I7.3 installed
bash-4.2#
```

パッチ RPM の非アクティブ化

手順

	コマンドまたはアクション	目的
ステップ 1	<code>sudo yum erase patch_RPM --nocommit</code>	パッチ RPM を非アクティブ化します。 (注) コマンドに <code>--nocommit</code> フラグを追加すると、パッチ RPM はこの手順でのみ非アクティブ化されます。
ステップ 2	<code>sudo yum install patch_RPM --commit</code>	パッチ RPM をコミットします。パッチ RPM をコミットしないまま削除しようとする、エラーメッセージが表示されます。

例

次に、`nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000` パッチ RPM を非アクティブにする例を示します。

```
bash-4.2# sudo yum erase nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000 --nocommit

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
                : protect-packages
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package nxos.CSCab00001-n9k_ALL.lib32_n9000 0:1.0.0-7.0.3.I7.3 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                        Arch      Version      Repository    Size
=====
Removing:
nxos.CSCab00001-n9k_ALL        lib32_n9000  1.0.0-7.0.3.I7.3  @patching    82 k

Transaction Summary
=====
Remove      1 Package

Installed size: 82 k
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
[#####          ] 30%error: reading
/var/sysmgr/tmp/patches/CSCab00001-n9k_ALL/isan/bin/sysinfo manifest, non-printable
characters found
```

```

Erasing      : nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000      1/1
[#####] 90%
Removed:
  nxos.CSCab00001-n9k_ALL.lib32_n9000 0:1.0.0-7.0.3.I7.3

Complete!
Install operation 143 completed successfully at Tue Mar 27 21:03:47 2018.

[#####] 100%
bash-4.2#

```

パッチ RPM は、非アクティブ化した後にコミットする必要があります。パッチ RPM を非アクティブ化した後にコミットしなかった場合に、[パッチ RPM の削除 \(15 ページ\)](#) の手順を使用してパッチ RPM を削除しようとすると、エラーメッセージが表示されます。

```

bash-4.2# sudo yum install nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000 --commit

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
                : protect-packages

groups-repo           | 1.1 kB      00:00 ...
localdb               | 951 B       00:00 ...
patching              | 951 B       00:00 ...
thirdparty            | 951 B       00:00 ...
Install operation 144 completed successfully at Tue Mar 27 21:09:28 2018.

[#####] 100%
bash-4.2#

```

次のコマンドを入力して、パッチ RPM が正常にコミットされたことを確認します。

```

bash-4.2# yum list --patch-only
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
                : protect-packages

groups-repo           | 1.1 kB      00:00 ...
localdb               | 951 B       00:00 ...
patching              | 951 B       00:00 ...
thirdparty            | 951 B       00:00 ...
nxos.CSCab00001-n9k_ALL.lib32_n9000  1.0.0-7.0.3.I7.3  patching
bash-4.2#

```

パッチ RPM の削除

手順

	コマンドまたはアクション	目的
ステップ 1	<code>sudo yum install --remove <i>patch_RPM</i></code>	非アクティブなパッチ RPM を削除します。

例

次に、`nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000` パッチ RPM を削除する例を示します。

```

bash-4.2# sudo yum install --remove nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
groups-repo                               | 1.1 kB      00:00 ...
localdb                                    | 951 B      00:00 ...
patching                                   | 951 B      00:00 ...
thirdparty                                 | 951 B      00:00 ...
[#####] 50%Install operation 145 completed successfully at Tue Mar 27
21:11:05 2018.

[#####] 100%
bash-4.2#

```



(注) パッチ RPM を削除しようとした後に、次のエラーメッセージが表示されたとします。

Install operation 11 "failed because patch was not committed". at Wed Mar 28 22:14:05 2018

これは、削除を試みる前にパッチ RPM をコミットしていなかったことを意味しています。パッチ RPM を削除する前にコミットする手順については、[パッチ RPM の非アクティブ化 \(14 ページ\)](#) を参照してください。

次のコマンドを入力して、非アクティブなパッチ RPM が正常に削除されたことを確認します。

```

bash-4.2# yum list --patch-only
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
               : protect-packages
groups-repo                               | 1.1 kB      00:00 ...
localdb                                    | 951 B      00:00 ...
patching                                   | 951 B      00:00 ...
patching/primary                           | 197 B      00:00 ...
thirdparty                                 | 951 B      00:00 ...
bash-4.2#

```

SDK または ISO で構築されたサードパーティ プロセスの永続的なデーモン化

アプリケーションには、`/etc/init.d/application_name` にインストールされる起動 Bash スクリプトが必要です。この起動 Bash スクリプトは、次の一般的な形式にする必要があります (この形式の詳細については、<http://linux.die.net/man/8/chkconfig> を参照してください)。

```

#!/bin/bash
#
# <application_name> Short description of your application
#
# chkconfig: 2345 15 85
# description: Short description of your application
#
### BEGIN INIT INFO
# Provides: <application_name>
# Required-Start: $local_fs $remote_fs $network $named

```



```
# Required-Stop: $local_fs $remote_fs $network
# Description: Short description of your application
### END INIT INFO
# See how we were called.
case "$1" in
start)
# Put your startup commands here
# Set RETVAL to 0 for success, non-0 for failure
;;
stop)
# Put your stop commands here
# Set RETVAL to 0 for success, non-0 for failure
;;
status)
# Put your status commands here
# Set RETVAL to 0 for success, non-0 for failure
;;
restart|force-reload|reload)
# Put your restart commands here
# Set RETVAL to 0 for success, non-0 for failure
;;
*)
echo $"Usage: $prog {start|stop|status|restart|force-reload}"
RETVAL=2
esac

exit $RETVAL
```

ネイティブ Bash シェルからのアプリケーションの永続的な起動

手順

-
- ステップ 1** 作成したアプリケーション起動 Bash スクリプトを `/etc/init.d/application_name` にインストールします。
 - ステップ 2** `/etc/init.d/application_name start` でアプリケーションを開始します
 - ステップ 3** `chkconfig --add application_name` を入力します
 - ステップ 4** `chkconfig --level 3 application_name on` を入力します
実行レベル3は、標準のマルチユーザー実行レベルであり、スイッチが通常実行されるレベルです。
 - ステップ 5** `chkconfig --list application_name` を実行して、アプリケーションがレベル3で実行されるようにスケジュールされていることを確認し、レベル3が `on` に設定されていることを確認します。
 - ステップ 6** アプリケーションが `/etc/rc3.d` に記述されていることを確認します。次のような表示になるはずですが、「S」の後に数字が続き、アプリケーション名（この例では `tcollector`）が続きます。 `../init.d/application_name` には Bash 起動スクリプトへのリンクが表示されます。
-

```
bash-4.2# ls -l /etc/rc3.d/tcollector
lrwxrwxrwx 1 root root 20 Sep 25 22:56 /etc/rc3.d/S15tcollector -> ../init.d/tcollector
bash-4.2#
```

ネイティブ Bash シェルのアプリケーション例

次の例は、ネイティブ Bash シェルのアプリケーションを示しています。

```
bash-4.2# cat /etc/init.d/hello.sh
#!/bin/bash

PIDFILE=/tmp/hello.pid
OUTPUTFILE=/tmp/hello

echo $$ > $PIDFILE
rm -f $OUTPUTFILE
while true
do
    echo $(date) >> $OUTPUTFILE
    echo 'Hello World' >> $OUTPUTFILE
    sleep 10
done
bash-4.2#
bash-4.2#
bash-4.2# cat /etc/init.d/hello
#!/bin/bash
#
# hello Trivial "hello world" example Third Party App
#
# chkconfig: 2345 15 85
# description: Trivial example Third Party App
#
### BEGIN INIT INFO
# Provides: hello
# Required-Start: $local_fs $remote_fs $network $named
# Required-Stop: $local_fs $remote_fs $network
# Description: Trivial example Third Party App
### END INIT INFO

PIDFILE=/tmp/hello.pid

# See how we were called.
case "$1" in
start)
    /etc/init.d/hello.sh &
    RETVAL=$?
    ;;
stop)
    kill -9 `cat $PIDFILE`
    RETVAL=$?
    ;;
status)
    ps -p `cat $PIDFILE`
    RETVAL=$?
    ;;
restart|force-reload|reload)
    kill -9 `cat $PIDFILE`
    /etc/init.d/hello.sh &
    RETVAL=$?
```

```
;;
*)
echo $"Usage: $prog {start|stop|status|restart|force-reload}"
RETVAL=2
esac

exit $RETVAL
bash-4.2#
bash-4.2# chkconfig --add hello
bash-4.2# chkconfig --level 3 hello on
bash-4.2# chkconfig --list hello
hello          0:off  1:off  2:on   3:on   4:on   5:on   6:off
bash-4.2# ls -al /etc/rc3.d/*hello*
lrwxrwxrwx 1 root root 15 Sep 27 18:00 /etc/rc3.d/S15hello -> ../init.d/hello
bash-4.2#
bash-4.2# reboot
```

リロード後

```
bash-4.2# ps -ef | grep hello
root      8790      1  0 18:03 ?          00:00:00 /bin/bash /etc/init.d/hello.sh
root      8973     8775  0 18:04 ttyS0    00:00:00 grep hello
bash-4.2#
bash-4.2# ls -al /tmp/hello*
-rw-rw-rw- 1 root root 205 Sep 27 18:04 /tmp/hello
-rw-rw-rw- 1 root root   5 Sep 27 18:03 /tmp/hello.pid
bash-4.2# cat /tmp/hello.pid
8790
bash-4.2# cat /tmp/hello
Sun Sep 27 18:03:49 UTC 2015
Hello World
Sun Sep 27 18:03:59 UTC 2015
Hello World
Sun Sep 27 18:04:09 UTC 2015
Hello World
Sun Sep 27 18:04:19 UTC 2015
Hello World
Sun Sep 27 18:04:29 UTC 2015
Hello World
Sun Sep 27 18:04:39 UTC 2015
Hello World
bash-4.2#
```


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。