



スケジューラの設定

この章は、次の項で構成されています。

- [スケジューラの概要](#) (1 ページ)
- [スケジューラの注意事項および制約事項](#) (2 ページ)
- [スケジューラのデフォルト設定](#) (3 ページ)
- [スケジューラの設定](#) (3 ページ)
- [スケジューラの設定確認](#) (12 ページ)
- [スケジューラの設定例](#) (13 ページ)
- [スケジューラの標準](#) (14 ページ)

スケジューラの概要

スケジューラを使用すると、次のようなメンテナンス作業のタイムテーブルを定義し、設定することができます。

- QoS (Quality of Service) ポリシーの変更
- データのバックアップ
- 設定の保存

ジョブは、定期的な作業を定義する単一または複数のコマンドで構成されています。ジョブは、1 回だけ、または定期的な間隔でスケジューリングすることができます。

スケジューラでは、ジョブと、そのタイムテーブルを次のように定義できます。

ジョブ

コマンドリストとして定義され、指定されたスケジュールに従って実行される定期的なタスク。

スケジュール

ジョブを実行するためのタイムテーブル。1 つのスケジュールに複数のジョブを割り当てるすることができます。

1 つのスケジュールは、定期的、または 1 回だけ実行するように定義されます。

- 定期モード：ジョブを削除するまで続行される繰り返しの間隔。次のタイプの定期的な間隔を設定できます。
 - Daily：ジョブは1日1回実行されます。
 - Weekly：ジョブは毎週1回実行されます。
 - Monthly：ジョブは毎月1回実行されます。
 - Delta：ジョブは、指定した時間に開始され、以後、指定した間隔（days:hours:minutes）で実行されます。
- 1回限定モード：ジョブは、指定した時間に1回だけ実行されます。

リモート ユーザ認証

ジョブの開始前に、スケジューラはジョブを作成したユーザーを認証します。リモート認証からのユーザークレデンシャルは、スケジュールされたジョブをサポートできるだけの十分に長い時間保持されないため、ジョブを作成するユーザーの認証パスワードをローカルで設定する必要があります。これらのパスワードは、スケジューラのコンフィギュレーションに含まれ、ローカル設定のユーザとは見なされません。

ジョブを開始する前に、スケジューラはローカルパスワードとリモート認証サーバに保存されたパスワードを照合します。

スケジューラ ログ ファイル

スケジューラは、ジョブ出力を含むログ ファイルを管理します。ジョブ出力のサイズがログ ファイルのサイズより大きい場合、出力内容は切り捨てられます。

スケジューラの注意事項および制約事項

- ジョブの実行中に次のいずれかの状況が発生した場合、スケジューラは失敗する可能性があります。
 - 機能ライセンスが、その機能のジョブがスケジュールされている時間に期限切れになった場合。
 - 機能が、その機能を使用するジョブがスケジューリングされている時間にディセーブルになっている場合。
 - 機能 ID= nxos-7k-only。3k はモジュラ シャーシではありません。
スロットからモジュールを取り外したにもかかわらず、そのスロットを対象にしたジョブがスケジューリングされている場合。

- 時刻が設定されていることを確認します。スケジュールはデフォルトのタイムテーブルを適用しません。スケジュールを作成し、ジョブを割り当てても、時刻を設定しなければ、ジョブは開始されません。
- ジョブを定義するときは、対話型コマンドや中断を伴うコマンド（例：**copy bootflash:file ftp:URI**、**write erase**、**reload**、およびその他の同様のコマンド）がジョブが非インタラクティブに開始され、実行されるため、指定されていないことを確認します。特定の時間にリロードジョブがスケジュールされ、実行されると、スイッチはブートループに入ります。したがって、スケジュール構成では使用しないでください。

スケジュールのデフォルト設定

表 1: コマンドスケジュールのパラメータのデフォルト

パラメータ	デフォルト
スケジュールの状態	ディセーブル
ログファイルサイズ	16 KB

スケジュールの設定

スケジュールのイネーブル化

始める前に

正しい VDC を使用していることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

手順の概要

1. switch# **configure terminal**
2. switch(config) # **feature scheduler**
3. (任意) switch(config) # **show scheduler config**
4. (任意) switch(config) # **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	switch# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	switch(config) # feature scheduler	スケジューラを現在の VDC で有効にします。
ステップ 3	(任意) switch(config) # show scheduler config	スケジューラ設定を表示します。
ステップ 4	(任意) switch(config)# copy running-config startup-config	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

例

次に、スケジューラをイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# feature scheduler
switch(config)# show scheduler config
config terminal
    feature scheduler
    scheduler logfile size 16
end
switch(config)#
```

スケジューラ ログ ファイル サイズの定義

始める前に

正しい VDC を使用していることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

手順の概要

1. switch# **configure terminal**
2. switch(config) # **scheduler logfile size value**
3. (任意) switch(config)# **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	switch# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	switch(config) # scheduler logfile size value	スケジューラ ログ ファイル サイズをキロバイト (KB) で定義します。 範囲は 16 ~ 1024 です。デフォルトのログ ファイル サイズは 16 です。 (注) ジョブ出力のサイズがログ ファイルのサイズより大きい場合、出力内容は切り捨てられます。
ステップ 3	(任意) switch(config)# copy running-config startup-config	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

例

次に、スケジューラ ログ ファイルのサイズを定義する例を示します。

```
switch# configure terminal
switch(config) # scheduler logfile size 1024
switch(config) #
```

リモート ユーザ認証の設定

リモートユーザは、ジョブを作成および設定する前に、クリア テキスト パスワードを使用して認証する必要があります。

次のコマンドの出力では、リモート ユーザー パスワードは常に暗号化された状態で表示されます。**show running-config** コマンドコマンドの暗号化オプション (7) は、ASCII デバイス構成をサポートします。

始める前に

正しい VDC を使用していることを確認します。VDC を変更するには、**switchto vdc** コマンド

手順の概要

1. switch# **configure terminal**
2. switch(config) # **scheduler aaa-authentication password [0 | 7] password**
3. switch(config) # **scheduler aaa-authentication username name password [0 | 7] password**

4. (任意) `switch(config) # show running-config |include "scheduler aaa-authentication"`
5. (任意) `switch(config)# copy running-config startup-config`

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	<code>switch# configure terminal</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<code>switch(config) # scheduler aaa-authentication password [0 7] password</code>	現在ログインしているユーザーのパスワードを設定します。 クリア テキストパスワードを設定するには、 <code>0</code> を入力します。 暗号化されたパスワードを設定するには、 <code>7</code> を入力します。
ステップ 3	<code>switch(config) # scheduler aaa-authentication username name password [0 7] password</code>	リモート ユーザーのクリア テキスト パスワードを設定します。
ステップ 4	(任意) <code>switch(config) # show running-config include "scheduler aaa-authentication"</code>	スケジューラのパスワード情報を表示します。
ステップ 5	(任意) <code>switch(config)# copy running-config startup-config</code>	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

例

次に、NewUser という名前のリモート ユーザーのクリア テキスト パスワードを設定する例を示します。

```
switch# configure terminal
switch(config) # scheduler aaa-authentication
username NewUser password z98y76x54b
switch(config) # copy running-config startup-config
switch(config) #
```

ジョブの定義

一旦ジョブを定義すると、コマンドの変更、削除はできません。ジョブを変更するには、そのジョブを削除して新しいジョブを作成する必要があります。

始める前に

正しい VDC を使用していることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

手順の概要

1. switch# **configure terminal**
2. switch(config) # **scheduler job name name**
3. switch(config-job) # **command1 ; [command2 ;command3 ; ...**
4. (任意) switch(config-job) # **show scheduler job [name]**
5. (任意) switch(config-job) # **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	switch# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	switch(config) # scheduler job name name	ジョブを指定された名前で作成し、ジョブ構成モードを開始します。 事前 <i>name</i> は、31 文字までに制限されています。
ステップ 3	switch(config-job) # command1 ; [command2 ;command3 ; ...	特定のジョブに対応するコマンドシーケンスを定義します。複数のコマンドは、スペースとセミコロンで (;) で区切る必要があります。 ファイル名は現在のタイムスタンプとスイッチ名を使用して作成します。
ステップ 4	(任意) switch(config-job) # show scheduler job [name]	ジョブ情報を表示します。 事前 <i>name</i> は、31 文字までに制限されています。
ステップ 5	(任意) switch(config-job) # copy running-config startup-config	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

例

次の例は、次の方法を示します。

- 「backup-cfg」という名前のスケジューラ ジョブを作成示します。
- 実行中の構成をブートフラッシュ上のファイルに保存します。

- ファイルをブートフラッシュから TFTP サーバーにコピーします。
- 変更がスタートアップ構成に保存されます。

```
switch# configure terminal
switch(config) # scheduler job name backup-cfg
switch(config-job) # copy running-config
tftp://1.2.3.4/$(SWITCHNAME)-cfg.$(TIMESTAMP) vrf management
switch(config-job) # copy running-config startup-config
```

ジョブの削除

始める前に

正しい VDC を使用していることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

手順の概要

1. switch# **configure terminal**
2. switch(config) # **no scheduler job name name**
3. (任意) switch(config-job) # **show scheduler job [name]**
4. (任意) switch(config-job) # **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	switch# configure terminal	グローバル構成モードを開始します。
ステップ 2	switch(config) # no scheduler job name name	特定のジョブおよびそこで定義されたすべてのコマンドを削除します。 事前 <i>name</i> は、31 文字までに制限されています。
ステップ 3	(任意) switch(config-job) # show scheduler job [name]	ジョブ情報を表示します。
ステップ 4	(任意) switch(config-job) # copy running-config startup-config	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

例

次に、configsave という名前のジョブを削除する例を示します。

```
switch# configure terminal
switch(config)# no scheduler job name configsave
switch(config-job)# copy running-config startup-config
switch(config-job)#
```

タイムテーブルの定義

タイムテーブルを設定する必要があります。設定しないと、ジョブがスケジューリングされません。

time コマンドの時間を指定しない場合、スケジューラは現在時刻を想定します。たとえば、現在の時刻が 2008 年 3 月 24 日の 22 時 00 分である場合、ジョブは次のように開始されます。

- **time start 23:00 repeat 4:00:00** コマンドの開始時刻が、2008 年 3 月 24 日 23 時 00 分であると見なします。
- **time daily 55** コマンドの開始時刻が、毎日 22 時 55 分であると見なします。
- **time weekly 23:00** コマンドの開始時刻が、毎週金曜日の 23 時 00 分であると見なします。
- **time monthly 23:00** コマンドの開始時刻が、毎月 24 日の 23 時 00 分であると見なします。



- (注) スケジューラは、1 つ前のジョブが完了しない限り、次のジョブを開始しません。たとえば、1 分間隔で実行するジョブを 22 時 00 分に開始するようジョブをスケジューリングしたが、ジョブを完了するには 2 分間必要である場合、ジョブは次のように実行されます。スケジューラは 22 時 00 分に最初のジョブを開始し、22 時 02 分に完了します。次に 1 分間待機し、22 時 03 分に次のジョブを開始します。

始める前に

正しい VDC を使用していることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

手順の概要

1. switch# **configure terminal**
2. switch(config)# **scheduler schedule name name**
3. switch(config-schedule)# **job name name**
4. switch(config-schedule)# **time daily time**
5. switch(config-schedule)# **time weekly** [[day-of-week:] HH:] MM
6. switch(config-schedule)# **time monthly** [[day-of-month:] HH:] MM
7. switch(config-schedule)# **time start** {now repeat repeat-interval | delta-time [repeat repeat-interval]}
8. (任意) switch(config-schedule)# **show scheduler config**
9. (任意) switch(config-schedule)# **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	switch# configure terminal	グローバル構成モードを開始します。
ステップ 2	switch(config) # scheduler schedule name name	新しいスケジューラを作成し、そのスケジュールのスケジュール コンフィギュレーション モードを開始します。 事前 <i>name</i> は、31 文字までに制限されています。
ステップ 3	switch(config-schedule) # job name name	このスケジュールにジョブを関連付けます。1つのスケジュールに複数のジョブを追加できます。 事前 <i>name</i> は、31 文字までに制限されています。
ステップ 4	switch(config-schedule) # time daily time	ジョブが毎日 HH:MM の形式で指定された時刻に開始することを意味します。
ステップ 5	switch(config-schedule) # time weekly [[<i>day-of-week:</i>] <i>HH:</i>] <i>MM</i>	ジョブが週の指定された曜日に開始することを意味します。 曜日は整数で表されます（例：1 は日曜日、2 月曜日）または略語（たとえば、 sun 、 mon ）。 引数全体の最大長は 10 文字です。
ステップ 6	switch(config-schedule) # time monthly [[<i>day-of-month:</i>] <i>HH:</i>] <i>MM</i>	ジョブが月の特定の日に開始することを意味します。 29、30 または 31 のいずれかを指定した場合、そのジョブは各月の最終日に開始されます。
ステップ 7	switch(config-schedule) # time start { now repeat repeat-interval <i>delta-time</i> [repeat repeat-interval]}	ジョブが定期的に開始することを意味します。 <i>start-time</i> の形式は [[[[<i>yyyy:</i>] <i>mmm:</i>] <i>dd:</i>] <i>HH:</i>] <i>MM</i> です。 <ul style="list-style-type: none"> • <i>delta-time</i>– スケジュールの構成後、ジョブの開始までの待機時間を指定します。 • 確認する– ジョブが今から 2 分後に開始することを指定します。 • repeat repeat-interval– ジョブを反復する回数を指定します。
ステップ 8	(任意) switch(config-schedule) # show scheduler config	スケジューラを表示します。

	コマンドまたはアクション	目的
ステップ 9	(任意) <code>switch(config-schedule)# copy running-config startup-config</code>	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

例

次に、ジョブが毎月 28 日の 23 時 00 分に開始するタイムテーブルを定義する例を示します。

```
switch# configure terminal
switch(config)# scheduler schedule name weekendbackupqos
switch(config-scheduler)# job name offpeakzoning
switch(config-scheduler)# time monthly 28:23:00
switch(config-scheduler)# copy running-config startup-config
switch(config-scheduler)#
```

スケジュール ログ ファイルの消去

始める前に

正しい VDC を使用していることを確認します。VDC を変更するには、`switchto vdc` コマンドを使用します。

手順の概要

1. `switch# configure terminal`
2. `switch(config)# clear scheduler logfile`

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	<code>switch# configure terminal</code>	グローバル構成モードを開始します。
ステップ 2	<code>switch(config)# clear scheduler logfile</code>	スケジュール ログ ファイルを消去します。

例

次に、スケジュール ログ ファイルを消去する例を示します。

```
switch# configure terminal
switch(config)# clear scheduler logfile
```

スケジューラのディセーブル化

始める前に

正しい VDC を使用していることを確認します。VDC を変更するには、**switchto vdc** コマンドを使用します。

手順の概要

1. switch# **configure terminal**
2. switch(config) # **no feature scheduler**
3. (任意) switch(config) # **show scheduler config**
4. (任意) switch(config) # **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	switch# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	switch(config) # no feature scheduler	現在の VDC のスケジューラを無効化にします。
ステップ 3	(任意) switch(config) # show scheduler config	スケジューラ構成を表示します。
ステップ 4	(任意) switch(config) # copy running-config startup-config	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

例

次に、スケジューラをディセーブルにする例を示します。

```
switch# configure terminal
switch(config) # no feature scheduler
switch(config) # copy running-config startup-config
switch(config) #
```

スケジューラの設定確認

次のいずれかのコマンドを使用して、設定を確認します。

表 2: スケジュールの `show` コマンド

コマンド	目的
<code>show scheduler config</code>	スケジュール設定を表示します。
<code>show scheduler job [name name]</code>	設定されているジョブを表示します。
<code>show scheduler logfile</code>	スケジュールログファイルの内容を表示します。
<code>show scheduler schedule [name name]</code>	設定されているスケジュールを表示します。

スケジュールの設定例

スケジュール ジョブの作成

この例では、実行コンフィギュレーションをブートフラッシュ内のファイルに保存するスケジュールジョブを作成する方法を示します。このジョブは、その後で、ブートフラッシュから TFTP サーバにファイルをコピーします（現在のタイムスタンプとスイッチ名を使用してファイル名を作成します）。

```
switch# configure terminal
switch(config)# scheduler job name backup-cfg
switch(config-job)# copy running-config
tftp://1.2.3.4/$(SWITCHNAME)-cfg.$(TIMESTAMP) vrf management
switch(config-job)# end
switch(config)#
```

スケジュール ジョブのスケジュールリング

次に、`backup-cfg` という名前のスケジュール ジョブを、毎日午前1時に実行するようスケジュールリングする例を示します。

```
switch# configure terminal
switch(config)# scheduler schedule name daily
switch(config-schedule)# job name backup-cfg
switch(config-schedule)# time daily 1:00
switch(config-schedule)# end
switch(config)#
```

ジョブ スケジュールの表示

次に、ジョブ スケジュールを表示する例を示します。

```
switch# show scheduler schedule
Schedule Name      : daily
-----
User Name          : admin
Schedule Type      : Run every day at 1 Hrs 00 Mins
```

```

Last Execution Time : Fri Jan 2 1:00:00 2009
Last Completion Time: Fri Jan 2 1:00:01 2009
Execution count      : 2
-----
Job Name              Last Execution Status
-----
back-cfg              Success (0)
switch(config)#

```

スケジュール ジョブの実行結果の表示

次に、スケジュールによって実行されたスケジュール ジョブの結果を表示する例を示します。

```

switch# show scheduler logfile
Job Name      : back-cfg              Job Status: Failed (1)
Schedule Name : daily                User Name  : admin
Completion time: Fri Jan 1  1:00:01 2009
----- Job Output -----
`cli var name timestamp 2009-01-01-01.00.00`
`copy running-config bootflash:/${HOSTNAME}-cfg.${timestamp}`
`copy bootflash:/switch-cfg.2009-01-01-01.00.00 tftp://1.2.3.4/ vrf management `
copy: cannot access file '/bootflash/switch-cfg.2009-01-01-01.00.00'
=====
Job Name      : back-cfg              Job Status: Success (0)
Schedule Name : daily                User Name  : admin
Completion time: Fri Jan 2  1:00:01 2009
----- Job Output -----
`cli var name timestamp 2009-01-02-01.00.00`
`copy running-config bootflash:/switch-cfg.2009-01-02-01.00.00`
`copy bootflash:/switch-cfg.2009--01-02-01.00.00 tftp://1.2.3.4/ vrf management `
Connection to Server Established.
[                               ] 0.50KBTrying to connect to tftp server.....
[#####                        ] 24.50KB
TFTP put operation was successful
=====
switch#

```

スケジュールの標準

この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。

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。