スケジューラの設定

この章の内容は、次のとおりです。

- スケジューラの概要，1 ページ
- スケジューラのライセンス要件，2 ページ
- スケジューラの注意事項および制約事項，3 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，10 ページ
- スケジューラの設定例，11 ページ
- スケジューラの標準，12 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を認証します。 リモート認証からのユーザ 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。

スケジューラのライセンス要件

この機能には、ライセンスは必要ありません。 ライセンスパッケージに含まれていない機能はすべてCiscoNX-OSシステムイメージにバンドルされており、追加費用は一切発生しません。 Cisco NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

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。
- 時刻が設定されていることを確認します。 スケジューラ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。
- ジョブは開始されると非インタラクティブ方式で実行されるため、ジョブの定義中、インタラクティブなコマンドや中断を伴うコマンド（例：**copy bootflash:** *file* **ftp:***URI*、**write erase**、その他類似のコマンド）が指定されていないことを確認してください。

スケジューラのデフォルト設定

表 *1*：コマンド スケジューラのパラメータのデフォルト

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

スケジューラの設定

スケジューラのイネーブル化

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。

	コマンドまたはアクション	目的
ステップ 2	switch(config) # **feature scheduler**	スケジューラをイネーブルにします。
ステップ 3	switch(config) # **show scheduler config**	（任意） スケジューラ設定を表示します。
ステップ 4	switch(config)# **copy running-config startup-config**	（任意） 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、変更を永続的に保存します。

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

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	switch(config)# **copy running-config startup-config**	（任意） 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、変更を永続的に保存します。

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。

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

手順

	コマンドまたはアクション	目的
ステップ **1**	switch# **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 aaa-authentication password** [**0** | **7**] *password*	現在ログインしているユーザ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、**0** を入力します。 暗号化パスワードを設定するには、**7** を入力します。
ステップ **3**	switch(config) # **scheduler aaa-authentication username** *name* **password** [**0** | **7**] *password*	リモート ユーザのクリア テキスト パスワードを設定します。
ステップ **4**	switch(config) # **show running-config | include "scheduler aaa-authentication"**	（任意） 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**	switch(config)# **copy running-config startup-config**	（任意） 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、変更を永続的に保存します。

次に、NewUser という名前のリモート ユーザのクリア 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。

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 コンフィギュレーション モードを開始します。 *name* 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。 コマンドはスペースとセミコロン（;）で区切ります。 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されます。
ステップ 4	switch(config-job) # **show scheduler job** [*name*]	（任意） ジョブ情報を表示します。 *name* は 31 文字までに制限されています。
ステップ 5	switch(config-job) # **copy running-config startup-config**	（任意） 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、変更を永続的に保存します。

次に、backup-cfg という名前のスケジューラ ジョブを作成し、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し、そのファイルをブートフラッシュから TFTP サーバにコピーし、変更をスタートアップ コンフィギュレーションに保存する例を示します。

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

ジョブの削除

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 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。 *name* は 31 文字までに制限されています。
ステップ 3	switch(config-job) # **show scheduler job** [*name*]	（任意） ジョブ情報を表示します。
ステップ 4	switch(config-job) # **copy running-config startup-config**	（任意） 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、変更を永続的に保存します。

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 コマンドで時刻を設定しない場合は、スケジューラは現在の時刻を使用します。 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。

- スケジューラは、**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。
- スケジューラは、**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。
- スケジューラは、**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。
- スケジューラは、**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。

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 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。 *name* 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。 *name* 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** [[*day-of-week*:] *HH*:] *MM*	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（たとえば、日曜日は **1**、月曜日は **2**）または略語（たとえば、**sun**、 **mon**）で表します。 引数全体の最大長は 10 文字です。
ステップ 6	switch(config-schedule) # **time monthly** [[*day-of-month*:] *HH*:] *MM*	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* | *delta-time* [**repeat** *repeat-interval*]}	ジョブが定期的に開始することを意味します。 start-time の形式は [[[[yyyy:]mmm:]dd:]HH]:MM です。 • *delta-time*：スケジュールの設定後、ジョブの開始までの待機時間を指定します。

	コマンドまたはアクション	目的
		• **now**：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**	switch(config-schedule) # **show scheduler config**	（任意） スケジューラの情報を表示します。
ステップ **9**	switch(config-schedule) # **copy running-config startup-config**	（任意） 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、変更を永続的に保存します。

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)#
```

スケジューラ ログ ファイルの消去

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) # **clear scheduler logfile**	スケジューラ ログ ファイルの消去

次に、スケジューラ 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

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**	スケジューラをディセーブルにします。
ステップ 3	switch(config) # **show scheduler config**	（任意） スケジューラ設定を表示します。
ステップ 4	switch(config)# **copy running-config startup-config**	（任意） 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、変更を永続的に保存します。

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

設定を確認するには、次のいずれかのコマンドを使用します。

表 *2*：スケジューラの ***show*** コマンド

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

スケジューラの設定例

スケジューラ ジョブの作成

次に、実行中のコンフィギュレーションを bootflash 内のファイルに保存し、ファイルを bootflash から TFTP サーバにコピーするスケジューラ ジョブを作成する例を示します（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されます）。

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

スケジューラ ジョブのスケジューリング

次に、backup-cfg という名前のスケジューラ ジョブを、毎日午前 1 時に実行するようスケジュー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)#
```

スケジューラ ジョブの実行結果の表示

次に、スケジューラによって実行されたスケジューラ 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#
```

スケジューラの標準

この機能では、新規の標準がサポートされることも、一部変更された標準がサポートされることもありません。また、既存の標準に対するサポートが変更されることもありません。