



# スケジュール オプション

- ・ [導入スケジュール オプション \(1 ページ\)](#)

## 導入スケジュール オプション

### スケジュールの作成

#### 手順

	コマンドまたはアクション	目的
ステップ 1	UCS-A# <b>scope system</b>	システム モードを開始します。
ステップ 2	UCS-A /system # <b>create scheduler</b> <i>sched-name</i>	スケジューラを作成し、スケジューラ モードを開始します。
ステップ 3	UCS-A /system/scheduler # <b>commit-buffer</b>	トランザクションをシステムの設定にコ ミットします。

#### 例

次の例では、`maintenancesched` という名前のスケジューラを作成し、トランザクションをコミットします。

```
UCS-A# scope system
UCS-A /system # create scheduler maintenancesched
UCS-A /system/scheduler* # commit-buffer
UCS-A /system/scheduler #
```

#### 次のタスク

スケジュールのワнтаイム オカレンスまたは繰り返しオカレンスを作成します。

## スケジュールのワンタイム オカレンスの作成

### 手順

	コマンドまたはアクション	目的
ステップ 1	UCS-A# <b>scope system</b>	システム モードを開始します。
ステップ 2	UCS-A /system # <b>scope schedule</b> <i>sched-name</i>	スケジューラ システム モードを開始します。
ステップ 3	UCS-A /system/scheduler # <b>create</b> <b>occurrence one-time</b> <i>occurrence-name</i>	ワンタイム オカレンスを作成します。
ステップ 4	UCS-A /system/scheduler/one-time # <b>set</b> <b>date</b> <i>month day-of-month year hour</i> <i>minute</i>	このオカレンスを実行する日時を設定します。
ステップ 5	(任意) UCS-A /system/scheduler/one-time # <b>set</b> <b>concur-tasks</b> { <b>unlimited</b>   <i>max-num-concur-tasks</i>	このオカレンスの間に同時実行可能なタスクの最大数を設定します。  タスクの最大数に達すると、スケジューラは新しいタスクをスケジュールする前に、[ <b>minimum interval</b> ] プロパティで設定された時間だけ待機します。
ステップ 6	(任意) UCS-A /system/scheduler/one-time # <b>set</b> <b>max-duration</b> { <b>none</b>   <i>num-of-days</i> <i>num-of-hours num-of-minutes</i> <i>num-of-seconds</i> }	このスケジュール オカレンスを実行できる最長時間を設定します。ここで指定された時間内に、Cisco UCS は可能な限り多くのスケジュール済みタスクを完了します。
ステップ 7	(任意) UCS-A /system/scheduler/one-time # <b>set</b> <b>min-interval</b> { <b>none</b>   <i>num-of-days</i> <i>num-of-hours num-of-minutes</i> <i>num-of-seconds</i> }	システムが新しいタスクを開始するまで待機する時間の最小長を設定します。
ステップ 8	(任意) UCS-A /system/scheduler/one-time # <b>set proc-cap</b> { <b>unlimited</b>   <i>max-num-of-tasks</i> }	このオカレンスの間に実行可能な、スケジュール設定されたタスクの最大数を設定します。
ステップ 9	UCS-A /system/scheduler/one-time # <b>commit-buffer</b>	トランザクションをシステムの設定にコミットします。

## 例

次の例では、`onetimemaint` という名前のワンタイム オカレンスを `maintsched` という名前のスケジューラに作成します。また、同時実行タスクの最大数を 5 に設定し、開始日時を 2011 年 4 月 1 日 11:00 に設定して、トランザクションをコミットします。

```
UCS-A# scope system
UCS-A /system # scope scheduler maintsched
UCS-A /system/scheduler # create occurrence one-time onetimemaint
UCS-A /system/scheduler/one-time* # set date apr 1 2011 11 00
UCS-A /system/scheduler/one-time* # set concur-tasks 5
UCS-A /system/scheduler/one-time* # commit-buffer
UCS-A /system/scheduler/one-time #
```

## スケジュールへの繰り返しオカレンスの作成

## 手順

	コマンドまたはアクション	目的
ステップ 1	UCS-A# <b>scope system</b>	システム モードを開始します。
ステップ 2	UCS-A /system # <b>scope schedule</b> <i>sched-name</i>	スケジューラシステムモードを開始します。
ステップ 3	UCS-A /system/scheduler # <b>create</b> <b>occurrence recurring</b> <i>occurrence-name</i>	繰り返しオカレンスを作成します。
ステップ 4	(任意) UCS-A /system/scheduler/recurring # <b>set day</b> { <b>even-day</b>   <b>every-day</b>   <b>friday</b>   <b>monday</b>   <b>never</b>   <b>odd-day</b>   <b>saturday</b>   <b>sunday</b>   <b>thursday</b>   <b>tuesday</b>   <b>wednesday</b> }	Cisco UCS でこのスケジュールのオカレンスを実行する曜日を選択します。 デフォルトでは、このプロパティは <code>never</code> に設定されています。
ステップ 5	(任意) UCS-A /system/scheduler/recurring # <b>set hour</b> <i>hour</i>	このオカレンスが開始する時間 (時) を指定します。  (注) Cisco UCS は、最大長に達していない場合でも、すべての繰り返しオカレンスをそれが開始したのと同じ日に終了させます。たとえば、開始時刻を午後 11 時、最長継続時間を 3 時間に指定すると、Cisco UCS はこのオカレンスを午後 11 時に開始しますが、59 分しか経過していない午後 11 時 59 分に終了します。

	コマンドまたはアクション	目的
ステップ 6	(任意) UCS-A /system/scheduler/recurring # <b>set minute</b> <i>minute</i>	このオカレンスが開始する時間 (分) を指定します。
ステップ 7	(任意) UCS-A /system/scheduler/recurring # <b>set</b> <b>concur-tasks</b> { <b>unlimited</b>   <i>max-num-concur-tasks</i>	このオカレンスの間に同時実行可能なタスクの最大数を設定します。  タスクの最大数に達すると、スケジューラは新しいタスクをスケジュールする前に、[minimum interval] プロパティで設定された時間だけ待機します。
ステップ 8	(任意) UCS-A /system/scheduler/recurring # <b>set</b> <b>max-duration</b> { <b>none</b>   <i>num-of-hours</i> <i>num-of-minutes</i> <i>num-of-seconds</i> }	このスケジュールオカレンスを実行できる最長時間を設定します。ここで指定された時間内に、Cisco UCS は可能な限り多くのスケジュール済みタスクを完了します。
ステップ 9	(任意) UCS-A /system/scheduler/recurring # <b>set</b> <b>min-interval</b> { <b>none</b>   <i>num-of-days</i> <i>num-of-hours</i> <i>num-of-minutes</i> <i>num-of-seconds</i> }	システムが新しいタスクを開始するまで待機する時間の最小長を設定します。
ステップ 10	(任意) UCS-A /system/scheduler/recurring # <b>set proc-cap</b> { <b>unlimited</b>   <i>max-num-of-tasks</i> }	このオカレンスの間に実行可能な、スケジュール設定されたタスクの最大数を設定します。
ステップ 11	UCS-A /system/scheduler/recurring # <b>commit-buffer</b>	トランザクションをシステムの設定にコミットします。

### 例

次の例は、`maintsched` というスケジューラに `recurringmaint` という繰り返しオカレンスを作成し、同時実行タスクの最大数を 5 に設定し、このオカレンスの実行日を偶数日に設定し、11:05 から開始するように時間を設定してトランザクションをコミットします。

```
UCS-A# scope system
UCS-A /system # scope scheduler maintsched
UCS-A /system/scheduler # create occurrence recurring recurringmaint
UCS-A /system/scheduler/recurring* # set day even-day
UCS-A /system/scheduler/recurring* # set hour 11
UCS-A /system/scheduler/recurring* # set minute 5
UCS-A /system/scheduler/recurring* # set concur-tasks 5
UCS-A /system/scheduler/recurring* # commit-buffer
UCS-A /system/scheduler/recurring #
```

## スケジュールからのワнтаイム オカレンスの削除

これがスケジュールにおける唯一の実行である場合には、そのスケジュールは実行なしで再設定されます。スケジュールがメンテナンスポリシーに含まれており、そのポリシーがサービスプロファイルに割り当てられている場合、サービスプロファイルに関連付けられているサーバに関連する保留中のアクティビティは展開できません。保留中のアクティビティを展開するには、ワнтаイムオカレンスまたは繰り返しオカレンスをスケジュールに追加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	UCS-A# <b>scope system</b>	システム モードを開始します。
ステップ 2	UCS-A /system # <b>scope scheduler</b> <i>sched-name</i>	スケジューラ システム モードを開始します。
ステップ 3	UCS-A /system/scheduler # <b>delete occurrence one-time</b> <i>occurrence-name</i>	指定されたワнтаイム オカレンスを削除します。
ステップ 4	UCS-A /system/scheduler # <b>commit-buffer</b>	トランザクションをシステムの設定にコミットします。

### 例

次の例は、maintsched スケジューラから onetimemaint というワнтаイム オカレンスを削除し、トランザクションをコミットします。

```
UCS-A# scope system
UCS-A /system # scope scheduler maintsched
UCS-A /system/scheduler # delete occurrence one-time onetimemaint
UCS-A /system/scheduler* # commit-buffer
UCS-A /system/scheduler #
```

## スケジュールからの繰り返しオカレンスの削除

これがスケジュールにおける唯一の実行である場合には、そのスケジュールは実行なしで再設定されます。スケジュールがメンテナンスポリシーに含まれており、そのポリシーがサービスプロファイルに割り当てられている場合、サービスプロファイルに関連付けられているサーバに関連する保留中のアクティビティは展開できません。保留中のアクティビティを展開するには、ワнтаイムオカレンスまたは繰り返しオカレンスをスケジュールに追加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	UCS-A# <b>scope system</b>	システム モードを開始します。
ステップ 2	UCS-A /system # <b>scope scheduler</b> <i>sched-name</i>	スケジューラ システム モードを開始します。
ステップ 3	UCS-A /system/scheduler # <b>delete</b> <b>occurrence recurring</b> <i>occurrence-name</i>	指定された繰り返しオカレンスを削除します。
ステップ 4	UCS-A /system/scheduler # <b>commit-buffer</b>	トランザクションをシステムの設定にコミットします。

## 例

次の例では、maintsched スケジューラから onetimemaint という名前の繰り返しオカレンスを削除し、トランザクションをコミットします。

```
UCS-A# scope system
UCS-A /system # scope scheduler maintsched
UCS-A /system/scheduler # delete occurrence recurring onetimemaint
UCS-A /system/scheduler* # commit-buffer
UCS-A /system/scheduler #
```

## スケジュールの削除

このスケジュールがメンテナンス ポリシーに含まれている場合、ポリシーはスケジュールなしで再設定されます。そのポリシーがサービスプロファイルに割り当てられている場合、サービスプロファイルに関連付けられているサーバに関連する保留中のアクティビティは展開できません。保留中のアクティビティを展開するには、スケジュールをメンテナンス ポリシーに追加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	UCS-A# <b>scope system</b>	システム モードを開始します。
ステップ 2	UCS-A /system # <b>delete scheduler</b> <i>sched-name</i>	スケジューラを削除し、スケジューラモードを開始します。
ステップ 3	UCS-A /system # <b>commit-buffer</b>	トランザクションをシステムの設定にコミットします。

**例**

次の例では、`maintenancesched` という名前のスケジューラを削除し、トランザクションをコミットします。

```
UCS-A# scope system  
UCS-A /system # delete scheduler maintenancesched  
UCS-A /system* # commit-buffer  
UCS-A /system #
```

