ネットワーク タイム プロトコルの設定

ネットワーク タイム プロトコル（NTP）は、ネットワーク内でデバイスの時刻同期を行うように設計されたプロトコルです。Cisco IOS XR ソフトウェア は NTPv4 を実装しています。NTPv4 は以前の NTP バージョンである NTPv3、NTPv2 との後方互換性はありますが、セキュリティ脆弱性のため中止となった NTPv1 との互換性はありません。

Cisco IOS XR ソフトウェアで NTP を実装するための前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

NTP の実装について

NTP を使用すると、分散されたタイム サーバとクライアントの間で時刻が同期されます。同期化により、システム ログ作成時または時間に関するイベントの発生時に、各イベントを関連付けることができます。

NTP ではトランスポート プロトコルとして、ユーザ データグラム プロトコル（UDP）を使用します。NTP の通信はすべて協定世界時（UTC）を使用します。NTP のネットワークでは通常、タイム サーバに接続された電波時計や原子時計など正規の時刻源から時刻を取得します。NTP はこの時刻をネットワーク全体に配信します。NTP はきわめて効率的で、毎分 1 パケット以下で 2 台のマシンを相互に 1 ミリ秒以内に同期します。

NTP では、各マシンが信頼できる時刻源から何 NTP ホップ隔たっているかを表すために「ストラタム」という概念を使用します。「ストラタム 1」タイム サーバには通常、正規の時刻源（電波時計、原子時計、GPS 時刻源など）が直接接続されています。「ストラタム 2」タイム サーバは、「ストラタム 1」タイム サーバから NTP を介して時刻を受信し、それ以降のサーバも続きます。

NTP では、2 つの方法で時刻が間違っている可能性のあるマシンとの同期を回避します。まず、NTP はそれ自身で同期を行わないマシンとの同期を回避します。次に、複数のマシンから報告された時間と大幅に時間が異なっているマシンがある場合、ストラタムの番号が小さくても同期しません。このようにして、NTP サーバのツリーは効率よく自律的に編成されています。

シスコの NTP 実装では、ストラタム 1 サービスをサポートしていないため、電波時計や原子時計に接続することはできません（ただし、いくつかの特定のプラットフォームでは、GPS 時刻源デバイスに接続できます）。ネットワークのタイム サービスは、IP インターネットで利用できる公開 NTP サーバから取得することを推奨します。

ネットワークがインターネットから切り離されている場合、シスコの NTP 実装では、実際には他の方法で時刻を決定している場合でも、NTP を介して同期されているものとして動作するようにマシンを設定できます。これにより、他のマシンが NTP を介してそのマシンと同期できるようになります。

自社のホスト システムに NTP ソフトウェアを組み込んでいるメーカーが数社あり、UNIX システム用のバージョンやその派生ソフトウェアも一般に入手できます。また、このソフトウェアにより UNIX 派生サーバは原子時計から時刻を直接取得することができ、シスコ ルータに時刻情報を伝えるようにすることもできます。

NTP を実行しているマシン間の通信（アソシエーション）は通常、静的に設定されており、各マシンには、アソシエーションを形成する必要があるすべてのマシンの IP アドレスが通知されます。アソシエーションが設定されたマシンの各ペアの間で NTP メッセージを交換することにより、正確な時刻管理が可能になります。

シスコの NTP 実装には、ネットワーク デバイスがネットワーク上で NTP 時刻情報を取得できる 3 つの方法があります。

- ホスト サーバへのポーリング
- NTP ブロードキャストのリスニング
- NTP マルチキャストのリスニング
- ホスト サーバへのポーリング
- NTP ブロードキャストのリスニング

LAN 環境では、IP ブロードキャスト メッセージまたは IP マルチキャスト メッセージを使用するように NTP を設定できます。ポーリングと比べ IP ブロードキャストまたは IP マルチキャスト メッセージではマシンごとにメッセージの送受信を設定するだけなので、複雑な設定作業が軽減されます。ただし、情報の流れが一方向に限定されるため、時刻管理の精度がわずかに低下します。

NTP ブロードキャスト クライアントは、指定した IPv4 アドレスにある NTP ブロードキャスト サーバから送信されるブロードキャスト メッセージをリスニングします。クライアントは最初に受信したブロードキャスト メッセージを使って、ローカルの時計を同期します。

NTP マルチキャスト サーバは、指定された IPv4 または IPv6 ローカル マルチキャスト グループ アドレスに定期的にメッセージを送信します。NTP マルチキャスト クライアントは、このアドレスで NTP メッセージをリスニングします。

マシン上の時刻は重要な情報であるため、NTP のセキュリティ機能を使用して、不正な時刻を誤って（または悪意を持って）設定できないように保護することを強く推奨します。その方法として、アクセス リストベースの制約方式と暗号化認証方式があります。

複数の時刻源（VINES、ハードウェア クロック、手動による設定）がある場合、NTP は常により信頼できる時刻源とされます。NTP の時刻は、他の方法による時刻に優先します。

GPS 週数ロールオーバー（WNRO）による問題の防止

- NTP ソース チェーンまたはサーバ チェーンに GPS ソースがない場合、GPS 週数ロールオーバー（WNRO）による影響はありません。
- GPS WNRO は、ユーザ トラフィックではなく、システム クロックにのみ影響を与えます。
- この条件に該当する GPS ソースを修正するには、GPS の製造元にお問い合わせください。

GPS WNRO の対象となる GPS ソースの影響を軽減するには、次のオプションの回避策を実行します。

- GPS ソースが 2019 年 4 月 6 日（またはそれ以降）に中断される可能性を引き起こす原因であると特定されている場合は、このソースに接続されているデバイスであるシスコとストラタム 1 デバイス上のクロックに ntp master を設定し、予防的措置として隔離します。この設定により、デバイスが下流の NTP クライアントへの同期用として独自のクロックを表示できるようになります。

（注） 前述のように、ntp master コマンドの使用は、この条件に対する回避策にすぎません。このコマンドは、ネットワーク全体への誤ったクロック値の分布を防止するために、GPS ソース関連の状況が解決されるまで使用します。

- 複数の NTP サーバ（3 つ以上、理想的には 4 つ）をネットワークのストラタム 2 レベルで構成し、ストラタム 2 レベルの NTP クライアントを有効にして、複数のストラタム 1 サーバからクロックを取得します。これにより、WNRO の影響を受けるストラタム 1 サーバは、WNRO の影響を受けないその他のストラタム 1 サーバと比較したときに、‘false ticker’ または ‘outlier’ クロック ソースとしてマークされるように調整されます。

NTP-PTP インターワーキング

NTP-PTP インターワーキングは、オペレーティング システムに時刻源として PTP、および Data over Cable Service Interface Specification（DOCSIS）Timing Interface（DTI）や Global Positioning System（GPS）などの他の有効な時刻（TOD）ソースを使用する機能を提供します。NTP-PTP

インターワーキングがサポートされる前は、オペレーティング システム時刻にはバックプレーンの時刻だけがサポートされていました。

NTP-PTP インターワーキングは、PTP および NTP プロセス間のステータス変更を通知する手段も提供します。また、起動、スイッチオーバー、またはカードおよびプロセス障害時に、オペレーティング システム時刻とバックプレーンの時刻の一義的な制御もサポートします。

Poll-Based アソシエーションの設定

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

ルータとその他のデバイス（ルータも可）間に、次のタイプの Poll-Based アソシエーションを設定できます。

- クライアント モード
- 対称アクティブ モード

クライアント モードと対称アクティブ モードは、高レベルの時刻の精度と信頼性を提供するために NTP が必要になる場合に使用します。

クライアント モードで動作しているネットワーキング デバイスは、自身に割り当てられている時刻提供ホストをポーリングして現在の時刻を取得します。次に、ネットワーキング デバイスは、ポーリングされたすべてのタイム サーバから、同期に使用するホストを選択します。この場合は、確立された関係がクライアントホスト関係なので、ホストがローカル クライアント デバイスから送信された時刻情報をキャプチャしたり使用したりすることはありません。このモードが最も適しているのは、他のローカル クライアントにどのような形式の時刻同期も提供する必要のない、ファイル サーバおよびワークステーションのクライアントです。ネットワーキング デバイスを同期させる時刻提供ホストを個別に指定し、クライアント モードで動作するようにネットワーキング デバイスを設定するには、 **server** コマンドを使用します。

対称アクティブ モードで動作しているネットワーキング デバイスは、自身に割り当てられている時刻提供ホストをポーリングして現在の時刻を取得し、そのホストによるポーリングに応答します。これはピアツーピアの関係であるため、ホストは通信相手のローカル ネットワーキング デバイスに関する時刻関連情報も保持します。相互に冗長な複数のサーバがダイバース ネットワーク パスを使用して相互に接続されている場合は、このモードを使用してください。現在のインターネットでは、ストラタム 1 サーバおよびストラタム 2 サーバのほとんどが、この形式のネットワーク設定を採用しています。ネットワーキング デバイスを同期させる時刻提供ホストを個別に指定し、対称アクティブ モードで動作するようにネットワーキング デバイスを設定するには、 **peer** コマンドを使用します。

他の複数のデバイスをポーリングして時刻を取得する場合、ルータは同期の対象となるデバイスを 1 台選択します。

（注） ルータと別のデバイス間のピアツーピア アソシエーションを設定するには、他のデバイスのピアとしてルータを設定する必要があります。

複数のピアおよびサーバを設定できますが、1 つの IP アドレスをピアとサーバの両方として同時に設定することはできません。

ピアからサーバ、またはサーバからピアへの特定の IP アドレスの設定を変更するには、 **peer** または **server** コマンドの **no** 形式を使用して、新しい設定を実行する前に現在の設定を削除します。新しい設定を実行する前に古い設定を削除しない場合、新しい設定によって古い設定は上書きされません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。

ステップ 3	**server** *ip-address* [**vrf** *vrf*] [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**] [**burst**] [**iburst**]
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# server 172.16.22.44`
	`minpoll 8 maxpoll 12`
		他のシステムとのサーバアソシエーションを形成します。この手順を必要に応じて繰り返し、複数のデバイスとのアソシエーションを形成できます。

ステップ 4	**peer** *ip-address* [**vrf** *vrf*] [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**]
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# peer 192.168.22.33`
	`minpoll 8 maxpoll 12 source tengige 0/0/0/1`
		他のシステムとのピア アソシエーションを形成します。この手順を必要に応じて繰り返し、複数のシステムとのアソシエーションを形成できます。
		（注） ルータとリモート デバイス間のピアツーピア アソシエーションの設定を完了するには、そのルータがリモート デバイス上でピアとして設定されている必要もあります。

ステップ 5	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

Broadcast-Based NTP アソシエーション

Broadcast-Based NTP アソシエーションでは、NTP サーバは、ネットワーク全体で NTP ブロードキャスト パケットを伝播します。ブロードキャスト クライアントは、NTP サーバによって伝搬されるブロードキャスト パケットをリッスンし、ポーリングには関与しません。

Broadcast-Based NTP アソシエーションは、時刻の精度および信頼性要件が緩やかであり、ネットワークがローカライズされ、クライアント数が多い（20 を超える）場合に使用します。また、帯域幅、システム メモリ、または CPU リソースが制限されているネットワークでも、

Broadcast-Based NTP アソシエーションの使用が推奨されます。情報の流れが一方向に限定されるため、Broadcast-Based NTP アソシエーションでは、時刻の精度がわずかに低下します。

ネットワークを通じて伝播される NTP ブロードキャスト パケットをリッスンするようにネットワーキング デバイスを設定するには、 **broadcast client** コマンドを使用します。ブロードキャスト クライアント モードが動作するためには、ブロードキャスト サーバとそのクライアントが同じサブネット上に存在する必要があります。 **broadcast** コマンドを使用して、NTP ブロードキャスト パケットを送信しているタイム サーバを特定のデバイスのインターフェイス上でイネーブルにする必要があります。

ネットワーキング デバイスが NTP ブロードキャスト パケットを送信するように設定するには、 **broadcast** コマンドを使用します。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	（任意） **broadcastdelay** *microseconds* 例： `RP/0/RP0/CPU0:router(config-ntp)# broadcastdelay 5000`	NTP ブロードキャストの推定ラウンドトリップ遅延を調整します。
ステップ 4	**interface** *type interface-path-id* 例： `RP/0/RP0/CPU0:router(config-ntp)# interface POS 0/1/0/0`	NTPインターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**broadcast client** 例： `RP/0/RP0/CPU0:router(config-ntp-int)# broadcast client`	指定されたインターフェイスが NTP ブロードキャスト パケットを受信するように設定します。 （注） インターフェイスが NTP ブロードキャスト パケットを送信するように設定するには、次の手順に移動します。

	コマンドまたはアクション	目的
ステップ 6	**broadcast** [**destination** *ip-address*] [**key** *key-id*] [**version** *number*] 例： `RP/0/RP0/CPU0:router(config-ntp-int)# broadcast destination 10.50.32.149`	指定されたインターフェイスが NTP ブロードキャスト パケットを送信するように設定します。 （注） インターフェイスが NTP ブロードキャスト パケットを受信するように設定するには、前の手順に移動します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp-int)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp-int)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

NTP アクセス グループの設定

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

アクセス リストベースの制限スキームを使用すると、ネットワーク全体、ネットワーク内のサブネット、またはサブネット内のホストに対し、特定のアクセス権限を許可または拒否できます。NTP 通信は、時刻要求と制御クエリーで構成されます。時刻要求とは、NTP サーバからの時刻同期の要求です。制御クエリーとは、NTP サーバからの設定情報の要求です。

アクセス グループのオプションは、次の順序で制限の緩いものから厳しいものへとスキャンされます。

1. **peer**：時刻要求と NTP 制御クエリーを許可し、システムがアクセス リストの基準を満たすアドレスを持つ別のシステムに同期することを許可します。
2. **serve**：時刻要求と NTP 制御クエリーを許可しますが、システムがアクセス リストの基準を満たすアドレスを持つ別のシステムに同期することは許可しません。
3. **serve-only**：アクセス リストの条件を満たすアドレスを持つシステムからの時刻要求のみを許可します。
4. **query-only**：アクセス リストの基準を満たすアドレスを持つ別のシステムからの NTP 制御クエリーのみを許可します。

複数のアクセス タイプについて送信元 IP アドレスがアクセス リストに一致する場合は、最初のタイプが認可されます。アクセス グループが指定されていない場合は、すべてのデバイスに対してすべてのアクセス タイプが認可されます。いずれかのアクセス グループが指定されている場合は、指定されたアクセス タイプだけが認可されます。

NTP 制御クエリーの詳細については、RFC 1305（NTP バージョン 3）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	**configure**	
ステップ 2	**ntp** **例**： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**access-group** {**peer** | **query-only** | **serve** | **serve-only**} *access-list-name* **例**： `RP/0/RP0/CPU0:router(config-ntp)# access-group peer access1`	アクセス グループを作成して、基本的な IPv4 または IPv6 アクセス リストを適用します。

	コマンドまたはアクション	目的

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

NTP 認証の設定

ここでは、NTP 認証の設定方法について説明します。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

信頼できる形式のアクセス コントロールが必要な場合は、暗号化された NTP 認証方式を使用する必要があります。IP アドレスに基づくアクセス リストベースの制約方式とは異なり、暗号化認証方式では認証キーと認証プロセスを使用して、ローカル ネットワーク上の指定されたピアまたはサーバによって送信された NTP 同期パケットが信頼できると見なすかどうかを、一緒に送信された時刻情報を受け入れる前に判断します。

認証プロセスは、NTP パケットが作成されるとすぐに開始されます。MD5 メッセージ ダイジェスト アルゴリズムを使用してメッセージ認証コード（MAC）が計算され、その MAC が NTP 同期パケットに埋め込まれます。NTP 同期パケットは、埋め込まれた MAC およびキー番号とともに受信側クライアントに送信されます。認証がイネーブルであり、キーが信頼できれば、受信側クライアントは同じ方法で MAC を計算します。計算された MAC と埋め込まれた MAC が一致すると、システムはパケットでこのキーを使用するサーバとの同期を許可されます。

NTP 認証が適切に設定されると、ネットワーキング デバイスは信頼できる時刻源と同期し、信頼できる時刻源だけに同期を提供します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** **例**： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**authenticate** **例**： `RP/0/RP0/CPU0:router(config-ntp)# authenticate`	NTP 認証機能をイネーブルにします。
ステップ 4	**authentication-key** *key-number* **md5** [**clear** | **encrypted**] *key-name* **例**： `RP/0/RP0/CPU0:router(config-ntp)# authentication-key 42 md5 clear key1`	認証キーを定義します。 • 各キーにはキー番号、タイプ、値が設定されており、オプションで名前が設定されます。現在サポートされているキー タイプは **md5** だけです。
ステップ 5	**trusted-key** *key-number* **例**： `RP/0/RP0/CPU0:router(config-ntp)# trusted-key 42`	`Defines trusted authentication keys.` • キーが信頼できる場合、このルータは NTP パケットでこのキーを使用するシステムとのみ同期します。
ステップ 6	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

特定のインターフェイス上の NTP サービスのディセーブル化

NTPサービスは、デフォルトではすべてのインターフェイスでディセーブルになっています。

なんらかの NTP コマンドを入力すると、NTP がグローバルにイネーブルになります。特定のインターフェイス上の NTP をオフにすることによって、NTP パケットが特定のインターフェイス経由で受信されることを選択的に防止できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **no interface** *type interface-path-id*
	• **interface** *type interface-path-id* **disable**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# no interface pos 0/0/0/1`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# interface POS 0/0/0/1 disable`
		指定されたインターフェイスでNTPサービスをディセーブルにします。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュ

	コマンドまたはアクション	目的
		レーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

NTP パケットの送信元 IP アドレスの設定

デフォルトでは、ルータが送信する NTP パケットの送信元 IP アドレスは、その NTP パケットが送信されたインターフェイスのアドレスです。別の送信元アドレスを設定するには、この手順を使用します。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**source** *type interface-path-id* 例：	IP 送信元アドレスの取得元のインターフェイスを設定します。

	コマンドまたはアクション	目的
	`RP/0/RP0/CPU0:router(config-ntp)# source POS 0/0/0/1`	（注） このインターフェイスは、すべての宛先に送信されるすべてのパケットの送信元アドレスに使用されます。特定のアソシエーションに送信元アドレスを使用する場合は、Poll-Based アソシエーションの設定 （4 ページ） に示す **peer** コマンドまたは **server** コマンドで **source** キーワードを使用します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続

	コマンドまたはアクション	目的
		するには、**commit** コマンドを使用します。

正規の NTP サーバとしてのシステムの設定

システムが外部の時刻源に同期化されていない場合でも、ルータが正規の NTP サーバとして動作するように設定することができます。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**master** *stratum* 例： `RP/0/RP0/CPU0:router(config-ntp)# master 9`	`Makes the router an authoritative NTP server.` （注） **master** コマンドの使用には注意が必要です。このコマンドを使用すると、有効な時刻源が容易に上書きされてしまいます。低いストラタム番号を設定する際には、特に注意が必要です。**master** コマンドを使用して同じネットワーク内の複数のマシンを設定した場合は、それらのマシンの時刻が一致していないと、時刻管理が不安定になることがあります。
ステップ 4	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

NTP-PTP インターワーキングの設定

PTP を時刻源として NTP が使用するように設定するには、このタスクを使用します。

始める前に

NTP-PTP インターワーキングを設定するには、ルータで PTP がサポートされていてイネーブルにされている必要があります。PTPがイネーブルでない場合、設定をコミットしようとすると、次のようなエラー メッセージが表示されます。

```
RP/0/RP0/CPU0:router(config)# ntp master primary-reference-clock
RP/0/RP0/CPU0:router(config)# commit

% Failed to commit one or more configuration items. Please issue
'show configuration failed' from this session to view the errors

RP/0/RP0/CPU0:router(config)# show configuration failed
[:::]
ntp
 master primary-reference-clock
!!% 'ip-ntp' detected the 'fatal' condition 'PTP is not supported on this platform'
!
end
```

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**master primary-reference-clock** 例： `RP/0/RP0/CPU0:router(config-ntp)# master primary-reference-clock`	PTP を NTP の時刻源に指定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモー

	コマンドまたはアクション	目的
		ドに戻ります。変更はコミットされません。 • **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

ハードウェア クロックの更新

ハードウェア クロック（システム カレンダー）が搭載されたデバイスでは、ハードウェア クロックを、ソフトウェア クロックから定期的に更新されるように設定できます。ソフトウェア クロック（NTP を使用して設定）の時刻と日付がハードウェア クロックよりも正確であるため、これは、NTP を使用するデバイスで推奨されます。ハードウェア クロックの時刻設定は、時間の経過とともにわずかにずれる可能性があります。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** **例**： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**update-calendar** **例**： `RP/0/RP0/CPU0:router(config-ntp)# update-calendar`	ルータがシステム カレンダーをソフトウェア クロックから定期間隔で更新するように設定します。

	コマンドまたはアクション	目的

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

外部基準クロックのステータスの確認

ここでは、NTP コンポーネントのステータスの確認方法について説明します。

（注） コマンドは任意の順序で入力できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**show ntp associations** [**detail**] [**location** *node-id*] 例： `RP/0/RP0/CPU0:router# show ntp associations`	NTP アソシエーションのステータスを表示します。
ステップ 2	**show ntp status** [**location** *node-id*] 例： `RP/0/RP0/CPU0:router# show ntp status`	NTP のステータスを表示します。

例

次に、 **show ntp associations** コマンドの出力例を示します。

```
RP/0/RP0/CPU0:router# show ntp associations

Tue Oct  7 11:22:46.839 JST

      address         ref clock     st  when  poll reach  delay  offset    disp
*~192.168.128.5    10.81.254.131     2     1    64  377    7.98  -0.560   0.108
+~dead:beef::2 vrf testAA
                   171.68.10.80      3    20    64  377    6.00  -2.832   0.046
* sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured


RP/0/RP0/CPU0:router# show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
+~127.127.1.1      127.127.1.1       5     5  1024   37     0.0    0.00   438.3
*~172.19.69.1      172.24.114.33     3    13  1024    1     2.0   67.16     0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

次に、 **show ntp status** コマンドの出力例を示します。

```
RP/0/RP0/CPU0:router# show ntp status

Tue Oct  7 11:22:54.023 JST

Clock is synchronized, stratum 3, reference is 192.168.128.5
nominal freq is 1000.0000 Hz, actual freq is 1000.2725 Hz, precision is 2**24
reference time is CC95463C.9B964367 (11:21:48.607 JST Tue Oct  7 2008)
clock offset is -1.738 msec, root delay is 186.050 msec
root dispersion is 53.86 msec, peer dispersion is 0.09 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.0002724105 s/s
system poll interval is 64, last update was 66 sec ago


RP/0/RP0/CPU0:router# show ntp status
```

```
Clock is synchronized, stratum 4, reference is 172.19.69.1
nominal freq is 1000.0000 Hz, actual freq is 999.9988 Hz, precision is 2**26
reference time is C54C131B.9EECF6CA (07:26:19.620 UTC Mon Nov 24 2008)
clock offset is 66.3685 msec, root delay is 7.80 msec
root dispersion is 950.04 msec, peer dispersion is 3.38 msec
```

NTP の実装の設定例

Poll-Based アソシエーションの設定：例

次に、ルータのシステム クロックが IP アドレス 192.168.22.33 のタイム サーバ ホストとのピア アソシエーションを形成し、IP アドレス 10.0.2.1 および 172.19.69.1 のタイム サーバ ホストによって同期されるように設定する、NTP の設定例を示します。

```
ntp
  server 10.0.2.1 minpoll 5 maxpoll 7
  peer 192.168.22.33

  server 172.19.69.1
```

Broadcast-Based アソシエーションの設定：例

次に、インターフェイス 0/11/0/0 が NTP ブロードキャスト パケットを受信するように設定し、NTP クライアントと NTP ブロードキャスト サーバ間の推定ラウンドトリップ遅延を 2 マイクロ秒に設定する、NTP クライアントの設定例を示します。

```
ntp
  interface tengige 0/11/0/0
    broadcast client
    exit
  broadcastdelay 2
```

次に、インターフェイス 0/11/0/2 がブロードキャスト サーバになるように設定する、NTP サーバの設定例を示します。

```
ntp
  interface tengige 0/11/0/2
    broadcast
```

Multicast-Based アソシエーションの設定：例

次に、10 ギガビット イーサネット インターフェイス 0/11/0/0 がマルチキャスト クライアントになり、デフォルトのマルチキャスト グループ（IPv4 アドレス 224.0.1.1）に参加するように設定する、NTP マルチキャスト クライアントの設定例を示します。

```
ntp interface TenGigE 0/11/0/0
  multicast client
```

次に、10 ギガビット イーサネット インターフェイス 0/11/0/0 がマルチキャスト サーバになるように設定する、NTP マルチキャスト サーバの設定例を示します。

```
ntp interface TenGigE 0/11/0/0
  multicast destination 224.0.1.1
```

NTP アクセス グループの設定：例

次に、以下のアクセス グループの制約事項が適用される NTP アクセス グループの設定例を示します。

- peer の制約事項は、peer-acl というアクセス リストの条件を満たす IP アドレスに適用されます。
- serve の制約事項は、serve-acl というアクセス リストの条件を満たす IP アドレスに適用されます。
- serve-only の制約事項は、serve-only-acl というアクセス リストの条件を満たす IP アドレスに適用されます。
- query-only の制約事項は、query-only-acl というアクセス リストの条件を満たす IP アドレスに適用されます。

```
ntp
  peer 10.1.1.1
  peer 10.1.1.1
  peer 10.2.2.2
  peer 10.3.3.3
  peer 10.4.4.4
  peer 10.5.5.5
  peer 10.6.6.6
  peer 10.7.7.7
  peer 10.8.8.8
  access-group peer peer-acl
  access-group serve serve-acl
  access-group serve-only serve-only-acl
  access-group query-only query-only-acl
  exit
ipv4 access-list peer-acl
  10 permit ip host 10.1.1.1 any
  20 permit ip host 10.8.8.8 any
  exit
ipv4 access-list serve-acl
  10 permit ip host 10.4.4.4 any
  20 permit ip host 10.5.5.5 any
  exit
ipv4 access-list query-only-acl
  10 permit ip host 10.2.2.2 any
  20 permit ip host 10.3.3.3 any
  exit
ipv4 access-list serve-only-acl
  10 permit ip host 10.6.6.6 any
  20 permit ip host 10.7.7.7 any
```

```
  exit
```

NTP 認証の設定：例

次に、NTP 認証の設定例を示します。この例では、次のように設定されます。

- NTP 認証がイネーブルになります。
- 2 つの認証キーが設定されます（キー 2 およびキー 3）。
- ルータは、ソフトウェア クロックが、認証キー 2 を使用する IP アドレス 10.3.32.154 のピアのクロックと（またはその逆に）同期することを許可するように設定されます。
- ルータは、ソフトウェア クロックが、認証キー 3 を使用する IP アドレス 10.32.154.145 のデバイスのクロックと同期することを許可するように設定されます。
- ルータは、NTP パケットに認証キー 3 を提供するシステムのみと同期するように設定されます。

```
ntp
  authenticate
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
  authentication-key 3 md5 encrypted 1311121E074110232621
  trusted-key 3
  server 10.3.32.154 key 3
  peer 10.32.154.145 key 2
```

インターフェイスでの NTP のディセーブル化：例

次に、0/11/0/0 インターフェイスをディセーブルにする NTP の設定例を示します。

```
ntp
  interface tengige 0/11/0/0
    disable
    exit
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
  authentication-key 3 md5 encrypted 1311121E074110232621
  authenticate
  trusted-key 3
  server 10.3.32.154 key 3
  peer 10.32.154.145 key 2
```

NTP パケット用の送信元 IP アドレスの設定：例

次に、イーサネット管理インターフェイス 0/RP0/CPU0/0 が NTP パケットの送信元アドレスとして設定される、NTP の設定例を示します。

```
ntp
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
```

```
authentication-key 3 md5 encrypted 1311121E074110232621
authenticate
trusted-key 3
server 10.3.32.154 key 3
peer 10.32.154.145 key 2
source MgmtEth0/0/CPU0/0
```

正規の NTP サーバとしてのシステムの設定：例

次に、外部の NTP ソースが使用不可になったときに、独自の NTP マスター クロックを使用してピアと同期するように in which the router を設定する、NTP の設定例を示します。

```
ntp
 master 6
```

ハードウェア クロックの更新：例

次に、NTP の設定例を示します。 the router is configured to update its hardware clock from the software clock at periodic intervals:

```
ntp
 server 10.3.32.154
 update-calendar
```

VRF インターフェイス内での NTP サーバの設定

ここでは、VRF インターフェイス内に NTP サーバを設定する方法について説明します。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP 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 3	**vrf** *vrf-name* 例： `RP/0/RP0/CPU0:router(config)# ntp vrf Customer_A`	設定する VRF（VPN ルーティングおよび転送）インスタンスの名前を指定します。
ステップ 4	**source** *interface-type interface-instance* 例： `RP/0/RP0/CPU0:router(config)# ntp vrf Customer_A source bvi 70`	IP 送信元アドレスの取得元のインターフェイスを設定します。これにより、IOS-XR は VRF インターフェイス上の NTP クエリに応答できます。この場合、送信元は BVI です。 （注） このインターフェイスは、すべての宛先に送信されるすべてのパケットの送信元アドレスに使用されます。特定のアソシエーションに送信元アドレスを使用する場合は、Poll-Based アソシエーションの設定 （4 ページ） に示す **peer** コマンドまたは **server** コマンドで **source** キーワードを使用します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。