



Cisco IOS XR ソフトウェアでのフレームリレーの設定

ここでは、フレームリレーのカプセル化で設定した Packet-over-SONET/SDH (POS)、マルチリンク、およびシリアル インターフェイスで使用できる、任意設定のフレームリレー パラメータについて説明します。

Cisco IOS XR ソフトウェアのフレームリレー インターフェイス設定の機能履歴

リリース	変更点
リリース 3.4.0	Cisco XR 12000 シリーズ ルータにこの機能が追加されました。
リリース 3.5.0	この機能は、IPv6 をサポートするために更新されました。 Layer 2 Tunnel Protocol Version 3 (L2TPv3) は、フレームリレーのカプセル化を設定したシリアル インターフェイスおよび POS インターフェイスでサポートされていました。
リリース 3.6.0	マルチリンク フレームリレー (FRF.16) およびエンドツーエンド フラグメンテーション (FRF.12) は、Cisco 1 ポート チャネライズド STM-1/OC-3 共有ポート アダプタ (SPA) で導入され、2 ポートおよび 4 ポート チャネライズド T3 SPA は Cisco XR 12000 シリーズ ルータで導入されました。
リリース 3.7.0	変更ありません。
リリース 3.8.0	FRF.16 および FRF.12 は、Cisco XR 12000 シリーズ ルータ の Cisco 1 ポート チャネライズド OC-12 SPA で導入されました。 フレームリレーのサポートは Cisco XR 12000 シリーズ ルータの次のラインカードで導入されました。 <ul style="list-style-type: none">• 1 ポート チャネライズド OC-12/STM-1 ラインカード• 4 ポート チャネライズド OC-12/STM-4 ラインカード

この章の構成

- 「フレームリレー設定の前提条件」 (P.170)
- 「フレームリレー インターフェイスに関する情報」 (P.170)
- 「フレームリレーの設定」 (P.177)
- 「フレームリレーの設定例」 (P.192)
- 「その他の参考資料」 (P.196)

フレームリレー設定の前提条件

フレームリレーを設定する前に、次の条件を満たしていることを確認します。

- この設定作業を行うには、Cisco IOS XR ソフトウェアのシステム管理者が、対応するコマンド タスク ID を含むタスク グループに関連付けられたユーザ グループにユーザを割り当てる必要があります。すべてのコマンド タスク ID は、各コマンド リファレンスおよび『*Cisco IOS XR Task ID Reference Guide*』に記載されています。

タスク グループの割り当てについてサポートが必要な場合は、システム管理者に連絡してください。ユーザ グループおよびタスク ID の詳細については、『*Cisco IOS XR Software System Security Configuration Guide*』の「*Configuring AAA Services on Cisco IOS XR Software*」モジュールを参照してください。

- 使用しているハードウェアが POS インターフェイスまたはシリアル インターフェイスをサポートしている必要があります。
- 対応するモジュールの説明に従って、**encapsulation frame relay** コマンドを使用し、インターフェイスでフレームリレーのカプセル化をイネーブルにしました。
 - マルチリンク バンドル インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このモジュールの「[マルチリンク フレームリレー バンドル インターフェイスの設定 \(P.182\)](#)」を参照してください。
 - POS インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの「[Cisco IOS XR ソフトウェアでの POS インターフェイスの設定](#)」モジュールを参照してください。
 - シリアル インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの「[Cisco IOS XR ソフトウェアでのシリアル インターフェイスの設定](#)」モジュールを参照してください。

フレームリレー インターフェイスに関する情報

ここでは、フレームリレー インターフェイスを設定する際のさまざまな側面について説明します。

- 「[フレームリレーのカプセル化 \(P.170\)](#)」
- 「[マルチリンク フレームリレー \(P.173\)](#)」
- 「[エンドツーエンド フラグメンテーション \(FRF.12\) \(P.177\)](#)」

フレームリレーのカプセル化

Cisco XR 12000 シリーズ ルータでは、フレームリレーは、POS インターフェイス、シリアル メイン インターフェイス、およびそれらのインターフェイスで設定された POV でサポートされます。フレームリレーのカプセル化をインターフェイスでイネーブルにするには、インターフェイス コンフィギュレーション モードで **encapsulation frame-relay** コマンドを使用します。

フレームリレー インターフェイスは、次の 2 種類のカプセル化されたフレームをサポートします。

- Cisco (これがデフォルト値です)
- IETF

PVC で Cisco または IETF のカプセル化を設定するには、フレームリレー PVC コンフィギュレーション モードで **encap** コマンドを使用します。



(注) **encap** コマンドで PVC のカプセル化タイプを設定しない場合、その PVC はメイン インターフェイスからカプセル化タイプを継承します。

encapsulation frame relay コマンドおよび **encap** コマンドについては、次のモジュールを参照してください。

- POS インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの「[Cisco IOS XR ソフトウェアでの POS インターフェイスの設定](#)」モジュールを参照してください。
- シリアル インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの「[Cisco IOS XR ソフトウェアでのシリアル インターフェイスの設定](#)」モジュールを参照してください。

フレームリレーのカプセル化でインターフェイスを設定し、その他の追加コンフィギュレーション コマンドを適用しない場合、表 7 のデフォルト インターフェイス設定が示されます。これらのデフォルト設定は、このモジュールの説明に従って設定で変更できます。

表 7 フレームリレーのカプセル化のデフォルト設定

パラメータ	コンフィギュレーション ファイルのエントリ	デフォルト設定	コマンド モード
PVC Encapsulation	encap [cisco ietf]	cisco (注) encap コマンドを設定しない場合、PVC のカプセル化タイプはフレームリレーのメイン インターフェイスから継承されます。	PVC の設定
Type of support provided by the interface	frame-relay intf-type [dce dte]	dte 。	インターフェイス コンフィギュレーション
LMI type supported on the interface	frame-relay lmi-type {ansi cisco q933a}	DCE の場合、デフォルト設定は cisco です。 DTE の場合、デフォルト設定は DCE でサポートされる LMI タイプに合わせて同期されます。 (注) インターフェイスをデフォルトの LMI タイプに戻すには、 no frame-relay lmi-type {ansi cisco q933a} コマンドを使用します。	インターフェイス コンフィギュレーション
Disable or enable LMI	frame-relay lmi disable	デフォルトでフレームリレー インターフェイスの LMI はイネーブルです。 LMI をディセーブルにした後で改めてイネーブルにするには、 no frame-relay lmi disable コマンドを使用します。	インターフェイス コンフィギュレーション



(注) LMI ポーリング関連のコマンドのデフォルト設定については、表 8 (P.173) と表 9 (P.173) を参照してください。

LMI

Local Management Interface (LMI; ローカル管理インターフェイス) プロトコルは、PVC の追加、削除、およびステータスをモニタリングします。また、フレームリレーの User-Network Interface (UNI; ユーザネットワーク インターフェイス) を構成するリンクの完全性も検証します。

フレームリレー インターフェイスは、UNI で次のタイプの LMI をサポートします。

- ANSI-ANSI T1.617 Annex D
- Q.933-ITU-T Q.933 Annex A
- Cisco

インターフェイスで使用する LMI タイプを設定するには、**frame-relay lmi-type** コマンドを使用します。



(注) 使用する LMI タイプは、メイン インターフェイスに設定されている PVC と対応している必要があります。フレームリレー接続の両エンドの LMI タイプは一致する必要があります。

使用しているルータが別の非フレームリレー ルータに接続するスイッチとして機能する場合、**frame-relay intf-type dce** コマンドを使用して、Data Communication Equipment (DCE; データ通信機器) をサポートする LMI タイプを設定します。

ルータがフレームリレー ネットワークに接続している場合、**frame-relay intf-type dte** コマンドを使用して、Data Terminal Equipment (DTE; データ端末装置) をサポートする LMI タイプを設定します。



(注) DTE インターフェイスでは、デフォルトで LMI タイプの自動検知がサポートされています。

システムのフレームリレー インターフェイスに関する統計情報を表示するには、EXEC モードで **show frame-relay lmi** コマンドを使用します (このコマンドの実際の構文は、**show frame-relay lmi interface type interface-path-id** です。type 引数および interface-path-id 引数を指定する場合、メイン インターフェイスの情報を指定する必要があります)。エラーのしきい値、イベント カウント、およびポーリングの検証タイマーを変更し、**show frame-relay lmi** コマンドを使用して、フレームリレー インターフェイスをモニタリングし、発生した問題を解決するときに役立つ情報を収集できます。

LMI タイプが **cisco** (デフォルトの LMI タイプ) の場合、単一のインターフェイスでサポートできる PVC の最大数は、メイン インターフェイスの MTU サイズに関連します。次の式を使用して、カードまたは SPA でサポートされる PVC の最大数を計算します。

$$(MTU - 13) / 8 = \text{PVC の最大数}$$

cisco LMI で設定した POS PVC でサポートされる PVC のデフォルトの数は 557 です。また、**cisco** LMI で設定したシリアル PVC でサポートされる PVC のデフォルトの数は 186 です。

シスコ製ではない LMI タイプの場合、単一のメイン インターフェイスで最大 992 PVC がサポートされます。



(注) 特定の LMI タイプをインターフェイスに設定する場合、**no frame-relay lmi-type {ansi | cisco | q933a}** コマンドを使用して、インターフェイスをデフォルトの LMI タイプに戻します。

表 8 は、DCE 用に設定した PVC で LMI ポーリング オプションを変更するときに使用できるコマンドです。

表 8 DCE の LMI ポーリング コンフィギュレーション コマンド

パラメータ	コンフィギュレーション ファイルのエントリ	デフォルト設定
Sets the error threshold on a DCE interface.	lmi-n392dce threshold	3
Sets the monitored event count.	lmi-n393dce events	4
Sets the polling verification timer on the DCE end.	lmi-t392dce seconds	15

表 9 は、DTE 用に設定した PVC で LMI ポーリング オプションを変更するときに使用できるコマンドです。

表 9 DTE の LMI ポーリング コンフィギュレーション コマンド

パラメータ	コンフィギュレーション ファイルのエントリ	デフォルト設定
Set the number of Line Integrity Verification (LIV) exchanges performed before requesting a full status message.	lmi-n391dte polling-cycles	6
Sets the error threshold.	lmi-n392dte threshold	3
Sets the monitored event count.	lmi-n393dte events	4
Sets the polling interval (in seconds) between each status inquiry from the DTE end.	frame-relay lmi-t391dte seconds	10

マルチリンク フレームリレー

マルチリンク フレームリレーは、次の SPA でのみサポートされます。

- Cisco 1 ポート チャネライズド STM-1/OC-3 SPA
- 2 ポートおよび 4 ポート チャネライズド T3 SPA
- Cisco 1 ポート チャネライズド OC-12 SPA

マルチリンク フレームリレー インターフェイスは、インターフェイスでフレームリレーのカプセル化を可能にするマルチリンク バンドルの一部です。マルチリンク フレームリレー インターフェイスを作成するには、次のコンポーネントを設定します。

- MgmtMultilink コントローラ
- フレームリレーのカプセル化を可能にするマルチリンク バンドル インターフェイス
- バンドル ID 名
- マルチリンク フレームリレー サブインターフェイス
- バンドル インターフェイスの帯域幅クラス
- シリアル インターフェイス

MgmtMultilink コントローラ

次のコマンドを使用して、コントローラのマルチリンク バンドルを設定します。

```
controller MgmtMultilink rack/slot/bay/controller-id
bundle bundleId
```

この設定で、汎用マルチリンク バンドルのコントローラが作成されます。コントローラ ID 番号はコントローラ チップのゼロベース インデックスです。現在、マルチリンク フレームリレーをサポートする SPA には、1 ペイごとに 1 コントローラしかないため、コントローラの ID 番号は常にゼロ (0) です。

マルチリンク バンドル インターフェイス

マルチリンク バンドルを作成した後は、次のコマンドを使用して、フレームリレーのカプセル化を可能にするマルチリンク バンドル インターフェイスを作成します。

```
interface Multilink rack/slot/bay/controller-id/bundleId
encapsulation frame-relay
```

この設定で、マルチリンク バンドル インターフェイスにマルチリンク フレームリレー サブインターフェイスを作成できます。



(注)

マルチリンク バンドル インターフェイス上のカプセル化をフレームリレーに設定した後は、マルチリンク バンドルに関連付けられたメンバー リンクがインターフェイスにある場合、カプセル化は変更できません。

バンドル ID 名



(注)

バンドル ID 名は、Frame Relay Forum 16.1 (FRF 16.1; フレームリレー フォーラム 16.1) でのみ設定できます。

バンドル ID (**bid**) 名は、インターフェイスの両エンドポイントのバンドル インターフェイスを識別します。バンドル ID 名は、一貫したリンクの割り当てを確保するために情報要素で交換されます。

デフォルトで、インターフェイス名 (たとえば **Multilink 0/4/1/0/1**) がバンドル ID 名として使用されます。ただし、オプションで **frame-relay multilink bid** コマンドを使用して名前を作成することもできます。



(注)

デフォルトの名前を使用するか、**frame-relay multilink bid** コマンドを使用して名前を作成するにかかわらず、各バンドルには固有の名前を指定する必要があります。異なるバンドルで同じ名前が使用されると、バンドルは永続的にフラップします。

バンドル ID 名の長さは、ヌルの終端文字を含めて 50 文字までです。バンドル ID 名はバンドル インターフェイス レベルで設定され、各メンバー リンクに適用されます。

バンドル ID 名を設定するには、次のコマンドを使用します。

```
interface Multilink rack/slot/bay/controller-id/bundleId
frame-relay multilink bid bundle-id-name
```

マルチリンク フレームリレー サブインターフェイス

マルチリンク フレームリレー サブインターフェイスを設定するには、次のコマンドを使用します。

interface Multilink rack/slot/bay/controller-id/bundleId.subinterface
[point-to-point | l2transport]

1 つのマルチリンク バンドル インターフェイスには最大 992 サブインターフェイスを設定できます。



(注) サブインターフェイス レベルで特定のフレームリレー インターフェイス機能を設定します。

マルチリンク フレームリレー サブインターフェイス機能

次のコマンドは、マルチリンク フレームリレー バンドル サブインターフェイスで特定の機能を設定するために使用できます。

- **mtu** *MTU size*
- **description**
- **shutdown**
- **bandwidth** *bandwidth*
- **service-policy** {input | output} *polycymap-name*



(注) **service-policy** コマンドを入力すると、ポリシー マップをマルチリンク フレームリレー バンドル サブインターフェイスに付加できるようになりますが、フレームリレー PVC コンフィギュレーション モードで次の操作を実行する必要があります。詳細については、「[マルチリンク フレームリレー バンドル インターフェイスの設定](#)」(P.182) を参照してください。

バンドル インターフェイスの帯域幅クラス

(注) 帯域幅クラスは、マルチリンク バンドル インターフェイスでのみ設定できます。

マルチリンク フレームリレー インターフェイスでは、3 タイプの帯域幅クラスのいずれかを設定できます。

- a - 帯域幅クラス A
- b - 帯域幅クラス B
- c - 帯域幅クラス C

帯域幅クラス A を設定し、1 つまたは複数のメンバー リンクがアップ (PH_ACTIVE) の場合、バンドル インターフェイスもアップで、BL_ACTIVATE がフレームリレー接続にシグナリングされます。すべてのメンバー リンクがダウンの場合、バンドル インターフェイスはダウンで、BL_DEACTIVATE がフレームリレー接続にシグナリングされます。

帯域幅クラス B を設定し、すべてのメンバー リンクがアップ (PH_ACTIVE) の場合、バンドル インターフェイスはアップで、BL_ACTIVATE がフレームリレー接続にシグナリングされます。いずれかのメンバー リンクがダウンの場合、バンドル インターフェイスはダウンで、BL_ACTIVATE がフレームリレー接続にシグナリングされます。

帯域幅クラス C を設定する場合、バンドルのリンクのしきい値を 1 ~ 255 に設定する必要があります。このしきい値は、バンドル インターフェイスをアップにするため、およびフレームリレー接続に BL_ACTIVATE をシグナリングするために必要な、アップ (PH_ACTIVE) にするリンクの最小数です。アップ状態のリンク数がこのしきい値未満の場合、バンドル インターフェイスがダウンになり、BL_DEACTIVATE がフレームリレー接続にシグナリングされます。しきい値に 1 を入力した場合の動作は、帯域幅クラス A と同じです。アップ状態のメンバー リンク数よりも多いしきい値を入力すると、バンドルはダウンのままです。

フレームリレー マルチリンク バンドル インターフェイスの帯域幅クラスを設定するには、次のコマンドを使用します。

```
interface Multilink rack/slot/bay/controller-id/bundleId
frame-relay multilink bandwidth-class a | b | c threshold
```

デフォルトは a (帯域幅クラス A) です。

シリアル インターフェイス

T3 コントローラと T1 コントローラを設定した後は、マルチリンク フレームリレー バンドル サブインターフェイスにシリアル インターフェイスを追加できます。この場合、シリアル インターフェイスを設定し、マルチリンク フレームリレー (mfr) としてカプセル化し、それをバンドル インターフェイス (マルチリンク グループ番号によって指定されます) に割り当て、リンクの名前を設定します。

マルチリンク フレームリレー シリアル インターフェイスを設定するには、次のコマンドを使用します。

```
interface serial rack/slot/module/port/t1-num:channel-group-number
encapsulation mfr
multilink group group number
frame-relay multilink lid link-id name
```



(注)

MFR バンドルのすべてのシリアル リンクは、マルチリンク インターフェイスの **mtu** コマンドの値を継承します。そのため、MFR バンドルのメンバーとして設定する前に、シリアル インターフェイスで **mtu** コマンドを設定しないでください。デフォルト以外の MTU 値を設定している場合、MFR バンドルのメンバーとしてシリアル インターフェイスを設定しようとすると、Cisco IOS XR ソフトウェアによってブロックされます。また、MFR バンドルのメンバーとして設定されているシリアル インターフェイスの **mtu** コマンド値を変更しようとした場合もブロックされます。

表示コマンド

マルチリンク フレームリレー シリアル インターフェイスの設定を検証するには、次の **show** コマンドを使用します。

```
show frame-relay multilink location
show frame-relay multilink interface serial
```

次の例は、**show frame-relay multilink location** コマンドの表示出力です。

```
RP/0/0/CPU0:router# show frame-relay multilink location 0/4/cpu0
Member interface: Serial0/4/2/0/9:0, ifhandle 0x05007b00
HW state = Up, link state = Up
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800

Bundle interface: Multilink0/4/2/0/2, ifhandle 0x05007800
  Member Links: 4 active, 0 inactive
  State = Up, BW Class = C (threshold 3)
  Member Links:
    Serial0/4/2/0/12:0, HW state = Up, link state = Up
    Serial0/4/2/0/11:0, HW state = Up, link state = Up
    Serial0/4/2/0/10:0, HW state = Up, link state = Up
    Serial0/4/2/0/9:0, HW state = Up, link state = Up

Member interface: Serial0/4/2/0/10:0, ifhandle 0x05007c00
HW state = Up, link state = Up
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800
```

```
Member interface: Serial0/4/2/0/11:0, ifhandle 0x05007d00  
HW state = Up, link state = Up  
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800
```

```
Member interface: Serial0/4/2/0/12:0, ifhandle 0x05007e00  
HW state = Up, link state = Up  
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800
```

次の例は、コマンドの表示出力です。

```
RP/0/0/CPU0:router# show frame-relay multilink interface serial 0/4/2/0/10:0
```

```
Member interface: Serial0/4/2/0/10:0, ifhandle 0x05007c00  
HW state = Up, link state = Up  
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800
```

エンドツーエンド フラグメンテーション (FRF.12)

Data-Link Connection Identifier (DLCI; データリンク接続識別子) を使用して、FRF.12 エンドツーエンドフラグメンテーション接続を設定することができます。ただし、チャネライズドフレームリレーシリアルインターフェイスで設定する必要があります。



(注) **fragment end-to-end** コマンドは、POS インターフェイス、またはマルチリンク フレームリレー バンドルインターフェイスの DLCI では使用できません。

FRF.12 エンドツーエンドフラグメンテーションを DLCI 接続で設定するには、次のコマンドを使用します。

```
fragment end-to-end fragment-size
```

fragment-size 引数には、シリアルインターフェイスのフラグメント サイズをバイト単位で定義します。



(注) DLCI 接続では、優先順位の高いフラグメントと低いフラグメントのインターリーブが発生するように、パケットを優先順位の高低で分類する出力サービス ポリシーを設定することを強くお勧めします。

フレームリレーの設定

次の項では、フレームリレー インターフェイスの設定方法について説明します。

- 「インターフェイスでのデフォルト フレームリレー設定の変更」(P.178)
- 「フレームリレーのカプセル化を設定したインターフェイスでの LMI のディセーブル」(P.180)
- 「マルチリンク フレームリレー バンドル インターフェイスの設定」(P.182)
- 「チャネライズドフレームリレー シリアルインターフェイスでの FRF.12 エンドツーエンドフラグメンテーションの設定」(P.188)

インターフェイスでのデフォルト フレームリレー設定の変更

フレームリレーのカプセル化を設定した POS インターフェイス、マルチリンク インターフェイス、またはシリアル インターフェイスでデフォルトのフレームリレー パラメータを変更するには、次のタスクを実行します。

前提条件

デフォルトのフレームリレー設定を変更する前に、次のモジュールの説明に従ってインターフェイスでフレームリレーをイネーブルにする必要があります。

- POS インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの「[Cisco IOS XR ソフトウェアでの POS インターフェイスの設定](#)」モジュールを参照してください。
- シリアル インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの「[Cisco IOS XR ソフトウェアでのシリアル インターフェイスの設定](#)」モジュールを参照してください。



(注)

POS インターフェイスまたはシリアル インターフェイスでフレームリレーのカプセル化をイネーブルにする前に、そのインターフェイスに IP アドレスを以前に割り当てたことがないことを確認します。IP アドレスが割り当て済みの場合、フレームリレーのカプセル化をイネーブルにすることはできません。フレームリレーの場合、IP アドレスとサブネット マスクはサブインターフェイスで設定します。

制約事項

- LMI タイプは、アクティブにする接続の両エンドで一致する必要があります。
- インターフェイスでフレームリレーのカプセル化を削除し、そのインターフェイスを PPP または HDLC のカプセル化で設定し直す前に、すべてのインターフェイス、サブインターフェイス、LMI、およびそのインターフェイスのフレームリレー設定を削除する必要があります。

手順の概要

1. **configure**
2. **interface type interface-path-id**
3. **frame-relay intf-type [dce | dte]**
4. **frame-relay lmi-type {ansi | cisco | q933a}**
5. **encap [cisco | ietf]**
6. **end**
または
commit
7. **show interfaces type [interface-path-id]**

詳細手順

コマンドまたはアクション	目的
ステップ 1 <code>configure</code> 例: RP/0/0/CPU0:router# configure	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>interface type interface-path-id</code> 例: RP/0/0/CPU0:router(config)# interface pos 0/4/0/1	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 3 <code>frame-relay intf-type [dce dte]</code> 例: RP/0/0/CPU0:router(config-if)# frame-relay intf-type dce	インターフェイスで提供するサポートのタイプを設定します。 <ul style="list-style-type: none"> • 使用しているルータが別のルータに接続するスイッチとして機能する場合、frame-relay intf-type dce コマンドを使用して、DCE をサポートする LMI タイプを設定します。 • ルータがフレームリレー ネットワークに接続している場合、frame-relay intf-type dte コマンドを使用して、DTE をサポートする LMI タイプを設定します。 (注) デフォルトのインターフェイス タイプは DTE です。
ステップ 4 <code>frame-relay lmi-type {ansi q933a cisco}</code> 例: RP/0/0/CPU0:router(config-if)# frame-relay lmi-type ansi	インターフェイスでサポートする LMI タイプを選択します。 <ul style="list-style-type: none"> • ANSI T1.617a-1994 Annex D の定義に従って LMI を使用するには、frame-relay lmi-type ansi コマンドを入力します。 • Cisco の定義（標準ではありません）に従って LMI を使用するには、frame-relay lmi-type cisco コマンドを使用します。 • ITU-T Q.933 (02/2003) Annex A の定義に従って LMI を使用するには、frame-relay lmi-type q933a コマンドを使用します。 (注) デフォルトの LMI タイプは Cisco です。
ステップ 5 <code>encap [cisco ietf]</code> 例: RP/0/0/CPU0:router (config-subif)# encap ietf	フレームリレー PVC のカプセル化を設定します。 (注) PVC のカプセル化タイプを明示的に設定しない場合、その PVC はメイン インターフェイスからカプセル化タイプを継承します。

■ フレームリレーの設定

コマンドまたはアクション	目的
<p>ステップ 6 <code>end</code> または <code>commit</code></p> <p>例: RP/0/0/CPU0:router(config-if)# <code>end</code> または RP/0/0/CPU0:router(config-if)# <code>commit</code></p>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> • end コマンドを発行すると、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設定変更が保存され、コンフィギュレーションセッションが終了し、ルータが EXEC モードに戻ります。 - no と入力すると、設定変更をコミットせずにコンフィギュレーションセッションが終了し、ルータが EXEC モードに戻ります。 - cancel と入力すると、コンフィギュレーションセッションの終了や設定変更のコミットは行われず、ルータでは現在のコンフィギュレーションセッションが継続されます。 • 設定変更を実行コンフィギュレーションファイルに保存し、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。
<p>ステップ 7 <code>show interfaces type [interface-path-id]</code></p> <p>例: RP/0/0/CPU0:router# <code>show interface pos 0/4/0/1</code></p>	<p>(任意) 指定したインターフェイスの設定を検証します。</p>

フレームリレーのカプセル化を設定したインターフェイスでの LMI のディセーブル

フレームリレーのカプセル化が設定されたインターフェイスで LMI をディセーブルにするには、次のタスクを実行します。



(注)

フレームリレーのカプセル化がイネーブルなインターフェイスでは、デフォルトで LMI がイネーブルです。インターフェイスの LMI をディセーブルにした後で改めてイネーブルにするには、インターフェイス コンフィギュレーション モードで **no frame-relay lmi disable** コマンドを使用します。

手順の概要

1. **configure**
2. **interface type interface-path-id**
3. **frame-relay lmi disable**
4. **end**
または
commit
5. **show interfaces type [interface-path-id]**

詳細手順

	コマンドまたはアクション	目的
ステップ 1	configure 例: RP/0/0/CPU0:router# configure	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	interface type interface-path-id 例: RP/0/0/CPU0:router(config)# interface POS 0/4/0/1	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 3	frame-relay lmi disable 例: RP/0/0/CPU0:router(config-if)# frame-relay lmi disable	指定したインターフェイスで LMI をディセーブルにします。
ステップ 4	end または commit 例: RP/0/0/CPU0:router(config-if)# end または RP/0/0/CPU0:router(config-if)# commit	設定変更を保存します。 <ul style="list-style-type: none"> • end コマンドを発行すると、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]: <ul style="list-style-type: none"> – yes と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – no と入力すると、設定変更をコミットせずにコンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – cancel と入力すると、コンフィギュレーション セッションの終了や設定変更のコミットは行われず、ルータでは現在のコンフィギュレーション セッションが継続されます。 • 設定変更を実行コンフィギュレーション ファイルに保存し、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。
ステップ 5	show interfaces type interface-path-id 例: RP/0/0/CPU0:router# show interfaces POS 0/1/0/0	(任意) 指定したインターフェイスで LMI がディセーブルになっていることを確認します。

マルチリンク フレームリレー バンドル インターフェイスの設定

マルチリンク フレームリレー バンドル インターフェイスとそのサブインターフェイスを設定するには、次の手順で操作します。

制約事項

- マルチリンク フレームリレー バンドル インターフェイスのすべてのメンバー リンクは、同じタイプにする必要があります。メンバー リンクは、ポイントツーポイントなど、同じフレーム構成タイプにし、同じ帯域幅クラスにする必要があります。
- すべてのリンクは、遠端でも同じラインカードまたは SPA に接続している必要があります。
- MFR バンドルのすべてのシリアル リンクは、マルチリンク インターフェイスの **mtu** コマンドの値を継承します。そのため、MFR バンドルのメンバーとして設定する前に、シリアル インターフェイスで **mtu** コマンドを設定しないでください。Cisco IOS XR ソフトウェアは次の処理をブロックします。
 - インターフェイスがデフォルト以外の MTU 値で設定されている場合、MFR バンドルのメンバーとしてシリアル インターフェイスを設定しようとする処理。
 - MFR バンドルのメンバーとして設定されているシリアル インターフェイスの **mtu** コマンド値を変更しようとする処理。

手順の概要

1. **config**
2. **controller MgmtMultilink rack/slot/bay/controller-id**
3. **exit**
4. **controller t3 interface-path-id**
5. **mode type**
6. **clock source {internal | line}**
7. **exit**
8. **controller t1 interface-path-id**
9. **channel-group channel-group-number**
10. **timeslots range**
11. **exit**
12. **exit**
13. **interface Multilink rack/slot/bay/controller-id/bundleId**
14. **encapsulation frame-relay**
15. **frame-relay multilink bid bundle-id-name**
16. **frame-relay multilink bandwidth-class a | b | c threshold**
17. **multilink fragment-size size**
18. **exit**
19. **interface Multilink rack/slot/bay/controller-id/bundleId.**
subinterface [point-to-point | l2transport]

20. `ipv4 address ip-address`
21. `pvc dlc`
22. `service-policy {input | output} policy-map`
23. `exit`
24. `exit`
25. `interface serial interface-path-id`
26. `encapsulation mfr`
27. `multilink group group-id`
28. `frame-relay multilink lid link-id name`
29. `exit`
30. `end`
または
`commit`
31. `exit`
32. `show frame-relay interface multilink rack/slot/bay/controller-id/bundleId`

詳細手順

	コマンドまたはアクション	目的
ステップ 1	<code>config</code> 例： RP/0/0/CPU0:router# config	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>controller MgmtMultilink rack/slot/bay/controller-id</code> 例： RP/0/0/CPU0:router(config)# controller MgmtMultilink 0/1/0/0	<code>rack/slot/bay/controller-id</code> 表記で汎用マルチリンクバンドルのコントローラを作成し、マルチリンク管理コンフィギュレーションモードを開始します。コントローラ ID 番号はコントローラチップのゼロベース インデックスです。現在、マルチリンクフレームリレーをサポートする SPA には、1 ベイごとに 1 コントローラしかないため、コントローラの ID 番号は常にゼロ (0) です。
ステップ 3	<code>exit</code> 例： RP/0/0/CPU0:router(config-mgmtmultilink)# exit	マルチリンク管理コンフィギュレーションモードを終了します。
ステップ 4	<code>controller t3 interface-path-id</code> 例： RP/0/0/CPU0:router(config)# controller t3 0/1/0/0	<code>rack/slot/module/port</code> 表記で T3 コントローラ名を指定し、T3 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	<code>mode type</code> 例： RP/0/0/CPU0:router(config-t3)# mode t1	チャネライズするマルチリンクのタイプを設定します (たとえば、28 T1)。

■ フレームリレーの設定

コマンドまたはアクション	目的
ステップ 6 <code>clock source {internal line}</code> 例: RP/0/0/CPU0:router(config-t3)# clock source internal	(任意) 個々の E3 リンクのクロッキングを設定します。 (注) デフォルトのクロック ソースは internal です。 (注) シリアルリンクでクロッキングを設定する場合、一方のエンドを internal にし、もう一方を line にする必要があります。接続の両エンドに internal クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに line クロッキングを設定すると、ラインはアップ状態になりません。
ステップ 7 <code>exit</code> 例: RP/0/0/CPU0:router(config-t3)# exit	T3/E3 コントローラ コンフィギュレーション モードを終了します。
ステップ 8 <code>controller t1 interface-path-id</code> 例: RP/0/0/CPU0:router(config)# controller t1 0/1/0/0/0	T1 コンフィギュレーション モードを開始します。
ステップ 9 <code>channel-group channel-group-number</code> 例: RP/0/0/CPU0:router(config-t1)# channel-group 0	T1 チャンネル グループを作成し、そのチャンネル グループのチャンネル グループ コンフィギュレーション モードを開始します。
ステップ 10 <code>timeslots range</code> 例: RP/0/0/CPU0:router(config-t1-channel_group)# timeslots 1-24	1 つまたは複数の DS0 タイムスロットをチャンネル グループに関連付け、関連付けたシリアル サブインターフェイスをそのチャンネル グループに作成します。 <ul style="list-style-type: none"> • 範囲は 1 ~ 24 タイムスロットです。 • 24 タイムスロットすべてを単一のチャンネル グループに割り当てることも、タイムスロットを複数のチャンネル グループに分割することもできます。 (注) 個々の T1 コントローラは、24 DS0 タイムスロットの合計をサポートします。
ステップ 11 <code>exit</code> 例: RP/0/0/CPU0:router(config-t1-channel_group)# exit	チャンネル グループ コンフィギュレーション モードを終了します。
ステップ 12 <code>exit</code> 例: RP/0/0/CPU0:router(config-t1)# exit	T1 コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 13	<pre>interface Multilink rack/slot/bay/controller-id/bundleId</pre> <p>例:</p> <pre>RP/0/0/CPU0:router(config)# interface Multilink 0/1/0/0/100</pre>	バンドルのフレームリレーのカプセル化を指定するマルチリンク バンドル インターフェイスを作成します。マルチリンク バンドル インターフェイスにマルチリンク フレームリレー サブインターフェイスを作成します。
ステップ 14	<pre>encapsulation frame-relay</pre> <p>例:</p> <pre>Router(config-if)# encapsulation frame-relay</pre>	フレームリレーのカプセル化タイプを指定します。
ステップ 15	<pre>frame-relay multilink bid bundle-id-name</pre> <p>例:</p> <pre>Router(config-if)# frame-relay multilink bid MFRBundle</pre>	<p>(任意) デフォルトで、インターフェイス名 (たとえば Multilink 0/4/1/0/1) がバンドル ID 名として使用されます。ただし、オプションで frame-relay multilink bid コマンドを使用して名前を作成することもできます。</p> <p>(注) デフォルトの名前を使用するか、frame-relay multilink bid コマンドを使用して名前を作成するかにかかわらず、各バンドルには固有の名前を指定する必要があります。異なるバンドルで同じ名前が使用されると、バンドルは永続的にフラップします。</p>
ステップ 16	<pre>frame-relay multilink bandwidth-class a b c threshold</pre> <p>例:</p> <pre>Router(config-if)# frame-relay multilink bandwidth-class a</pre>	<p>マルチリンク フレームリレー インターフェイスでは、3 タイプの帯域幅クラスのいずれかを設定します。</p> <ul style="list-style-type: none"> • a - 帯域幅クラス A • b - 帯域幅クラス B • c — 帯域幅クラス C <p>デフォルトは a (帯域幅クラス A) です。</p>
ステップ 17	<pre>multilink fragment-size size</pre> <p>例:</p> <pre>RP/0/0/CPU0:router(config-if)# multilink fragment-size 256</pre>	(任意) マルチリンク フラグメントのサイズを指定します。デフォルトは no fragments です。
ステップ 18	<pre>exit</pre> <p>例:</p> <pre>RP/0/0/CPU0:router(config-if)# exit</pre>	インターフェイス コンフィギュレーション モードを終了します。

■ フレームリレーの設定

コマンドまたはアクション	目的
<p>ステップ 19 <code>interface Multilink</code> <code>rack/slot/bay/controller-id/bundleId.</code> <code>subinterface [point-to-point l2transport]</code></p> <p>例: RP/0/0/CPU0:router(config)# interface Multilink 0/1/0/0/100.16 point-to-point</p>	<p><code>rack/slot/bay/controller-id bundleId.subinterface</code> [point-to-point l2transport] 表記でマルチリンクサブインターフェイスを作成し、サブインターフェイス コンフィギュレーション モードを開始します。</p> <ul style="list-style-type: none"> • l2transport - 接続回路として扱います。 • point-to-point - ポイントツーポイント リンクとして扱います。 <p>1 つのマルチリンク バンドル インターフェイスには最大 992 サブインターフェイスを設定できます。DLCI は 16 ~ 1007 です。</p>
<p>ステップ 20 <code>ipv4 address ip-address</code></p> <p>例: RP/0/0/CPU0:router(config-subif)# ipv4 address 3.1.100.16 255.255.255.0</p>	<p>次の形式でインターフェイスに IP アドレスおよびサブネット マスクを割り当てます。</p> <p><code>A.B.C.D/prefix</code> or <code>A.B.C.D/mask</code></p>
<p>ステップ 21 <code>pvc dlci</code></p> <p>例: RP/0/0/CPU0:router (config-subif)# pvc 16</p>	<p>POS 相手先固定接続 (PVC) を作成し、フレームリレー PVC コンフィギュレーション サブモードを開始します。</p> <p><code>dlci</code> を 16 から 1007 の範囲の PVC ID に置き換えます。</p> <p>(注) 各サブインターフェイスに設定できる PVC は 1 つだけです。</p>
<p>ステップ 22 <code>service-policy {input output} policy-map</code></p> <p>例: RP/0/0/CPU0:router(config-fr-vc) # service-policy output policy-mapA</p>	<p>ポリシー マップを入力サブインターフェイスまたは出力サブインターフェイスに付加します。付加すると、そのサブインターフェイスのサービス ポリシーとしてポリシー マップが使用されます。</p> <p>(注) ポリシー マップの作成と設定については、『Cisco IOS XR Modular Quality of Service Configuration Guide』を参照してください。</p>
<p>ステップ 23 <code>exit</code></p> <p>例: RP/0/0/CPU0:router(config-fr-vc) # exit</p>	<p>フレームリレー仮想回線モードを終了します。</p>
<p>ステップ 24 <code>exit</code></p> <p>例: RP/0/0/CPU0:router(config-subif) # exit</p>	<p>サブインターフェイス コンフィギュレーションモードを終了します。</p>
<p>ステップ 25 <code>interface serial interface-path-id</code></p> <p>例: RP/0/0/CPU0:router(config)# interface serial 0/1/0/0/0/0:0</p>	<p>完全なインターフェイス番号を <code>rack/slot/module/port/T3Num/T1num:instance</code> 表記で指定します。</p>
<p>ステップ 26 <code>encapsulation mfr</code></p> <p>例: RP/0/0/CPU0:router(config) # encapsulation mfr</p>	<p>シリアル インターフェイスでマルチリンク フレームリレーをイネーブルにします。</p>

コマンドまたはアクション	目的
ステップ 27 <code>multilink group group-id</code> 例: RP/0/0/CPU0:router(config-if)# multilink group 100	このインターフェイスのマルチリンク グループ ID を指定します。
ステップ 28 <code>frame-relay multilink lid link-id name</code> 例: RP/0/0/CPU0:router(config-if)# frame-relay multilink lid sj1	フレームリレー マルチリンク バンドル リンクの名前を設定します。 (注) バンドル内の各リンクには固有の名前を指定する必要があります。同じバンドル内の異なるリンクに同じ名前が使用されている場合、バンドルは永続的にフラップします。
ステップ 29 <code>exit</code> 例: RP/0/0/CPU0:router(config-if)# exit	インターフェイス コンフィギュレーション モードを終了します。
ステップ 30 <code>end</code> または commit 例: RP/0/0/CPU0:router(config-if)# end または RP/0/0/CPU0:router(config-if)# commit	設定変更を保存します。 <ul style="list-style-type: none"> • end コマンドを発行すると、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 - no と入力すると、設定変更をコミットせずにコンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 - cancel と入力すると、コンフィギュレーション セッションの終了や設定変更のコミットは行われず、ルータでは現在のコンフィギュレーション セッションが継続されます。 • 設定変更を実行コンフィギュレーション ファイルに保存し、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。
ステップ 31 <code>exit</code> 例: RP/0/0/CPU0:router(config)# exit	グローバル コンフィギュレーション モードを終了します。
ステップ 32 <code>show frame-relay interface multilink rack/slot/bay/controller-id/bundleId</code> 例: RP/0/0/CPU0:router# show frame-relay interface Multilink 0/5/1/0/1	バンドル固有の情報やフレームリレー情報など、 Interface Description Block (IDB; インターフェイス デスクリプション ブロック) から取得した情報を表示します。

チャネライズド フレームリレー シリアル インターフェイスでの FRF.12 エンドツーエンド フラグメンテーションの設定

チャネライズド フレームリレー シリアル インターフェイスで FRF.12 エンドツーエンド フラグメンテーションを設定するには、次の手順で操作します。

手順の概要

1. **config**
2. **controller t3** *interface-path-id*
3. **mode** *type*
4. **clock source** {**internal** | **line**}
5. **exit**
6. **controller t1** *interface-path-id*
7. **channel-group** *channel-group-number*
8. **timeslots** *range*
9. **exit**
10. **exit**
11. **interface serial** *interface-path-id*
12. **encapsulation** **frame-relay**
13. **exit**
14. **interface serial** *interface-path-id*
15. **ipv4 address** *ip-address*
16. **pvc** *dci*
17. **service-policy** {**input** | **output**} *policy-map*
18. **fragment end-to-end** *fragment-size*
19. **exit**
20. **exit**
21. **exit**
22. **end**
または
commit
23. **exit**
24. **show frame-relay pvc** [**dci** | **interface** | **location**]

詳細手順

	コマンドまたはアクション	目的
ステップ 1	<code>config</code> 例: RP/0/0/CPU0:router# config	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>controller t3 interface-path-id</code> 例: RP/0/0/CPU0:router(config)# controller t3 0/1/0/0	<code>rack/slot/module/port</code> 表記で T3 コントローラ名を指定し、T3 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	<code>mode type</code> 例: RP/0/0/CPU0:router(config-t3)# mode t1	チャネライズするマルチリンクのタイプを設定します (たとえば、28 T1)。
ステップ 4	<code>clock source {internal line}</code> 例: RP/0/0/CPU0:router(config-t3)# clock source internal	(任意) 個々の E3 リンクのクロッキングを設定します。 (注) デフォルトのクロック ソースは internal です。 (注) シリアルリンクでクロッキングを設定する場合、一方のエン드를 internal にし、もう一方を line にする必要があります。接続の両エン드에 internal クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エン드에 line クロッキングを設定すると、ラインはアップ状態になりません。
ステップ 5	<code>exit</code> 例: RP/0/0/CPU0:router(config-t3)# exit	T3/E3 または T1/E1 コントローラ コンフィギュレーション モードを終了します。
ステップ 6	<code>controller t1 interface-path-id</code> 例: RP/0/0/CPU0:router(config)# controller t1 0/1/0/0/0	T1 コンフィギュレーション モードを開始します。
ステップ 7	<code>channel-group channel-group-number</code> 例: RP/0/0/CPU0:router(config-t1)# channel-group 0	T1 チャネル グループを作成し、そのチャネルグループのチャネルグループ コンフィギュレーション モードを開始します。

■ フレームリレーの設定

コマンドまたはアクション	目的
ステップ 8 <code>timeslots range</code> 例: RP/0/0/CPU0:router(config-t1-channel_group)# timeslots 1-24	1 つまたは複数の DS0 タイムスロットをチャンネルグループに関連付け、関連付けたシリアル サブインターフェイスをそのチャンネルグループに作成します。 <ul style="list-style-type: none"> • 範囲は 1 ~ 24 タイムスロットです。 • 24 タイムスロットすべてを単一のチャンネルグループに割り当てることも、タイムスロットを複数のチャンネルグループに分割することもできます。 (注) 個々の T1 コントローラは、24 DS0 タイムスロットの合計をサポートします。
ステップ 9 <code>exit</code> 例: RP/0/0/CPU0:router(config-t1-channel_group)# exit	チャンネルグループ コンフィギュレーション モードを終了します。
ステップ 10 <code>exit</code> 例: RP/0/0/CPU0:router(config-t1)# exit	T1 コンフィギュレーション モードを終了します。
ステップ 11 <code>interface serial interface-path-id</code> 例: RP/0/0/CPU0:router(config)# interface serial 0/1/0/0/0/0:0	完全なインターフェイス番号を <code>rack/slot/module/port/T3Num/T1num:instance</code> 表記で指定します。
ステップ 12 <code>encapsulation frame-relay</code> 例: RP/0/0/CPU0:Router(config-if)# encapsulation frame-relay	フレームリレーのカプセル化タイプを指定します。
ステップ 13 <code>exit</code> 例: RP/0/0/CPU0:router(config-if)# exit	インターフェイス コンフィギュレーション モードを終了します。
ステップ 14 <code>interface serial interface-path-id</code> 例: RP/0/0/CPU0:router(config)# interface serial 1/0/0/0/0:0.1	<code>rack/slot/module/port[/channel-num:channel-group-number].subinterface</code> 表記で、完全なサブインターフェイス番号を指定します。
ステップ 15 <code>ipv4 address ip-address</code> 例: RP/0/0/CPU0:router(config-subif)# ipv4 address 3.1.100.16 255.255.255.0	次の形式でインターフェイスに IP アドレスおよびサブネット マスクを割り当てます。 A.B.C.D/prefix or A.B.C.D/mask

コマンドまたはアクション	目的
ステップ 16 <code>pvc dldci</code> 例: RP/0/0/CPU0:router (config-subif)# pvc 100	POS PVC を作成し、フレームリレー PVC コンフィギュレーション サブモードを開始します。 <i>dldci</i> を 16 から 1007 の範囲の PVC ID に置き換えます。 (注) 各サブインターフェイスに設定できる PVC は 1 つだけです。
ステップ 17 <code>service-policy {input output} policy-map</code> 例: RP/0/0/CPU0:router (config-fr-vc)# service-policy output policy-mapA	ポリシー マップを入力サブインターフェイスまたは出力サブインターフェイスに付加します。付加すると、そのサブインターフェイスのサービス ポリシーとしてポリシー マップが使用されます。 (注) 効率的な FRF.12 機能（具体的にはインターリーブ）のためには、出力サービス ポリシーに優先順位を設定します。 (注) ポリシー マップの作成と設定については、『Cisco IOS XR Modular Quality of Service Configuration Guide』を参照してください。
ステップ 18 <code>fragment end-to-end fragment-size</code> 例: RP/0/0/CPU0:router (config-fr-vc)# fragment end-to-end 100	インターフェイスでフレームリレー フレームのフラグメンテーションをイネーブルにします。 <i>fragment-size</i> を、発信元フレームリレー フレームのペイロード バイト数に置き換えます。これが各フラグメントのバイト数になります。この数値には、元のフレームのフレームリレー ヘッダーは含まれません。 有効な値は 16 ~ 1600 です。デフォルト値は 53 です。
ステップ 19 <code>exit</code> 例: RP/0/0/CPU0:router (config-fr-vc)# exit	フレームリレー仮想回線モードを終了します。
ステップ 20 <code>exit</code> 例: RP/0/0/CPU0:router (config-subif)# exit	サブインターフェイス コンフィギュレーション モードを終了します。
ステップ 21 <code>exit</code> 例: RP/0/0/CPU0:router (config-if)# exit	インターフェイス コンフィギュレーション モードを終了します。

■ フレームリレーの設定例

コマンドまたはアクション	目的
<p>ステップ 22 <code>end</code> または <code>commit</code></p> <p>例: RP/0/0/CPU0:router(config-if)# <code>end</code> または RP/0/0/CPU0:router(config-if)# <code>commit</code></p>	<p>設定変更を保存します。</p> <ul style="list-style-type: none"> • end コマンドを発行すると、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設定変更が保存され、コンフィギュレーションセッションが終了し、ルータが EXEC モードに戻ります。 – no と入力すると、設定変更をコミットせずにコンフィギュレーションセッションが終了し、ルータが EXEC モードに戻ります。 – cancel と入力すると、コンフィギュレーションセッションの終了や設定変更のコミットは行われず、ルータでは現在のコンフィギュレーションセッションが継続されます。 • 設定変更を実行コンフィギュレーションファイルに保存し、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。
<p>ステップ 23 <code>exit</code></p> <p>例: RP/0/0/CPU0:router(config)# <code>exit</code></p>	<p>グローバル コンフィギュレーション モードを終了します。</p>
<p>ステップ 24 <code>show frame-relay pvc [dlci interface location]</code></p> <p>例: RP/0/0/CPU0:router# <code>show frame-relay pvc 100</code></p>	<p>指定した PVC DLCI、インターフェイス、または場所の情報を表示します。</p>

フレームリレーの設定例

ここでは、次の設定例について説明します。

- 「オプションのフレームリレー パラメータ : 例」 (P.193)
- 「マルチリンク フレームリレー : 例」 (P.195)
- 「エンドツーエンド フラグメンテーション : 例」 (P.196)

オプションのフレームリレー パラメータ : 例

次の例は、フレームリレーのカプセル化を設定した POS インターフェイスを始動および設定する方法です。この例では、インターフェイスが DCE で ANSI T1.617a-1994 Annex D LMI をサポートするように、デフォルトのフレームリレー設定を変更します。

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router (config)# interface POS 0/3/0/0
RP/0/0/CPU0:router (config-if)# encapsulation frame-relay IETF
RP/0/0/CPU0:router (config-if)# frame-relay intf-type dce
RP/0/0/CPU0:router (config-if)# frame-relay lmi-type ansi
RP/0/0/CPU0:router (config-if)# no shutdown
RP/0/0/CPU0:router (config-if)# end
```

```
Uncommitted changes found, commit them? [yes]: yes
```

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router (config)# interface pos 0/3/0/0.10 point-to-point
RP/0/0/CPU0:router (config-subif)# ipv4 address 10.46.8.6/24
RP/0/0/CPU0:router (config-subif)# pvc 20
RP/0/0/CPU0:router (config-fr-vc)# encap ietf
RP/0/0/CPU0:router (config-subif)# commit
```

次の例は、フレームリレーのカプセル化を設定したシリアル インターフェイスで LMI をディセーブルにする方法です。

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router (config)# interface
RP/0/0/CPU0:router (config)# interface serial 0/3/0/0/0:0
RP/0/0/CPU0:router (config-if)# frame-relay lmi disable
RP/0/0/CPU0:router (config-if)# end
```

```
Uncommitted changes found, commit them? [yes]: yes
```

次の例は、シリアル インターフェイスで LMI を再度イネーブルにする方法です。

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router (config)# interface
RP/0/0/CPU0:router (config)# interface serial 0/3/0/0/0:0
RP/0/0/CPU0:router (config-if)# no frame-relay lmi disable
RP/0/0/CPU0:router (config-if)# end
```

```
Uncommitted changes found, commit them? [yes]: yes
```

次の例は、POS インターフェイス上の LMI に関するフレームリレーの統計情報を表示する方法です。

```
RP/0/0/CPU0:router# show frame-relay lmi

LMI Statistics for interface POS0/1/0/0/ (Frame Relay DCE) LMI TYPE = ANSI
Invalid Unnumbered Info 0          Invalid Prot Disc 0
Invalid Dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 9
Invalid Information ID 0           Invalid Report IE Len 0
Invalid Report Request 0           Invalid Keep IE Len 0
Num Status Enq. Rcvd 9444          Num Status Msgs Sent 9444
Num Full Status Sent 1578          Num St Enq. Timeouts 41
Num Link Timeouts 7

LMI Statistics for interface POS0/1/0/1/ (Frame Relay DCE) LMI TYPE = CISCO
Invalid Unnumbered Info 0          Invalid Prot Disc 0
Invalid Dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0           Invalid Report IE Len 0
```

■ フレームリレーの設定例

```

Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Rcvd 9481        Num Status Msgs Sent 9481
Num Full Status Sent 1588        Num St Enq. Timeouts 16
Num Link Timeouts 4

```

次の例は、メイン シリアル インターフェイス上の PVC でシリアル サブインターフェイスを作成する方法です。

```

RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# interface serial 0/3/0/0:0.10 point-to-point
RP/0/0/CPU0:router (config-subif)# ipv4 address 10.46.8.6/24
RP/0/0/CPU0:router (config-subif)# pvc 20
RP/0/0/CPU0:router (config-fr-vc)# encapsulation ietf
RP/0/0/CPU0:router(config-subif)# commit

```

次の例は、システムに設定されているすべての PVC に関する情報を表示する方法です。

```
RP/0/0/CPU0router# show frame-relay pvc
```

```
PVC Statistics for interface Serial0/3/2/0 (Frame Relay DCE)
```

	Active	Inactive	Deleted	Static
Local	4	0	0	0
Switched	0	0	0	0
Dynamic	0	0	0	0

```

DLCI = 612, DLCI USAGE = LOCAL, ENCAP = CISCO, INHERIT = TRUE, PVC STATUS = ACTI
VE, INTERFACE = Serial0/3/2/0.1

```

```

input pkts 0          output pkts 0          in bytes 0
out bytes 0          dropped pkts 0          in FECN packets 0
in BECN pkts 0        out FECN pkts 0          out BECN pkts 0
in DE pkts 0          out DE pkts 0
out bcast pkts 0      out bcast bytes 0
pvc create time 00:00:00    last time pvc status changed 00:00:00

```

```

DLCI = 613, DLCI USAGE = LOCAL, ENCAP = CISCO, INHERIT = TRUE, PVC STATUS = ACTI
VE, INTERFACE = Serial0/3/2/0.2

```

```

input pkts 0          output pkts 0          in bytes 0
out bytes 0          dropped pkts 0          in FECN packets 0
in BECN pkts 0        out FECN pkts 0          out BECN pkts 0
in DE pkts 0          out DE pkts 0
out bcast pkts 0      out bcast bytes 0
pvc create time 00:00:00    last time pvc status changed 00:00:00

```

```

DLCI = 614, DLCI USAGE = LOCAL, ENCAP = CISCO, INHERIT = TRUE, PVC STATUS = ACTI
VE, INTERFACE = Serial0/3/2/0.3

```

```

input pkts 0          output pkts 0          in bytes 0
out bytes 0          dropped pkts 0          in FECN packets 0
in BECN pkts 0        out FECN pkts 0          out BECN pkts 0
in DE pkts 0          out DE pkts 0
out bcast pkts 0      out bcast bytes 0
pvc create time 00:00:00    last time pvc status changed 00:00:00

```

```

DLCI = 615, DLCI USAGE = LOCAL, ENCAP = CISCO, INHERIT = TRUE, PVC STATUS = ACTI
VE, INTERFACE = Serial0/3/2/0.4

```

```

input pkts 0          output pkts 0          in bytes 0
out bytes 0          dropped pkts 0          in FECN packets 0
in BECN pkts 0        out FECN pkts 0          out BECN pkts 0
in DE pkts 0          out DE pkts 0
out bcast pkts 0      out bcast bytes 0
pvc create time 00:00:00    last time pvc status changed 00:00:00

```

次の例は、DTE 用に設定した PVC で LMI ポーリング オプションを変更し、**show frame-relay lmi** コマンドを使用して、モニタリングしてインターフェイスの問題を解決するための情報を表示する方法です。

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# interface serial 0/3/0/0/0
RP/0/0/CPU0:router(config-if)# frame-relay lmi-n391dte 10
RP/0/0/CPU0:router(config-if)# frame-relay lmi-n391dte 5
RP/0/0/CPU0:router(config-if)# frame-relay lmi-t391dte 15
RP/0/0/CPU0:router(config-subif)# commit

RP/0/0/CPU0:router# show frame-relay lmi

LMI Statistics for interface serial 0/3/0/0/0 (Frame Relay DTE) LMI TYPE = ANSI
Invalid Unnumbered Info 0          Invalid Prot Disc 0
Invalid Dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 9
Invalid Information ID 0           Invalid Report IE Len 0
Invalid Report Request 0           Invalid Keep IE Len 0
Num Status Enq. Rcvd 9444          Num Status Msgs Sent 9444
Num Full Status Sent 1578          Num St Enq. Timeouts 41
Num Link Timeouts 7
```

マルチリンク フレームリレー : 例

次の例は、シリアル インターフェイスでマルチリンク フレームリレーを設定する方法です。

```
RP/0/0/CPU0:router# config
RP/0/0/CPU0:router(config)# controller MgmtMultilink 0/3/1/0
RP/0/0/CPU0:router(config-mgmtmultilink)# bundle 100
RP/0/0/CPU0:router(config-mgmtmultilink)# exit

RP/0/0/CPU0:router(config)# controller T3 0/3/1/0
RP/0/0/CPU0:router(config-t3)# mode tl
RP/0/0/CPU0:router(config-t3)# clock source internal
RP/0/0/CPU0:router(config-t3)# exit

RP/0/0/CPU0:router(config)# controller T1 0/3/1/0/0
RP/0/0/CPU0:router(config-t1)# channel-group 0
RP/0/0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/0/CPU0:router(config-t1-channel_group)# exit
RP/0/0/CPU0:router(config-t1-channel_group)# exit
RP/0/0/CPU0:router(config-t1)# exit

RP/0/0/CPU0:router(config)# interface Multilink 0/3/1/0/100
RP/0/0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/0/CPU0:router(config-if)# exit

RP/0/0/CPU0:router(config)# interface Multilink 0/3/1/0/100.16 point-to-point
RP/0/0/CPU0:router(config-subif)# ipv4 address 3.1.100.16 255.255.255.0
RP/0/0/CPU0:router(config-subif)# pvc 16
RP/0/0/CPU0:router(config-fr-vc)# service-policy output policy-mapA
RP/0/0/CPU0:router(config-fr-vc)# exit
RP/0/0/CPU0:router(config-subif)# exit

RP/0/0/CPU0:router(config)# interface Serial 0/3/1/0/0:0
RP/0/0/CPU0:router(config-if)# encapsulation mfr
RP/0/0/CPU0:router(config-if)# multilink group 100
RP/0/0/CPU0:router(config-if)# frame-relay multilink lid sj1
RP/0/0/CPU0:router(config-if)# exit
RP/0/0/CPU0:router(config)#
```

エンドツーエンド フラグメンテーション：例

次の例は、チャネライズド フレームリレー シリアル インターフェイスで FRF.12 エンドツーエンド フラグメンテーションを設定する方法です。

```
RP/0/0/CPU0:router# config
RP/0/0/CPU0:router(config)# controller T30/3/1/0
RP/0/0/CPU0:router(config-t3)# mode t1
RP/0/0/CPU0:router(config-t3)# clock source internal
RP/0/0/CPU0:router(config-t3)# exit
RP/0/0/CPU0:router(config-t3)# controller T10/3/1/0/0
RP/0/0/CPU0:router(config-t1)# channel-group 0
RP/0/0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/0/CPU0:router(config-t1-channel_group)# exit
RP/0/0/CPU0:router(config-t1-channel_group)# interface Serial 0/3/1/0/0:0
RP/0/0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/0/CPU0:router(config-if)# exit
RP/0/0/CPU0:router(config-if)# interface Serial 0/3/1/0/0:0.100 point-to-point
RP/0/0/CPU0:router(config-subif)# ipv4 address 3.1.1.1 255.255.255.0
RP/0/0/CPU0:router(config-subif)# pvc 100
RP/0/0/CPU0:router(config-fr-vc)# service-policy output LFI
RP/0/0/CPU0:router(config-fr-vc)# fragment end-to-end 256
```

その他の参考資料

フレームリレーに関連する参考資料を示します。

関連資料

内容	参照先
Cisco IOS XR マスター コマンド リファレンス	『Cisco IOS XR Master Commands List』
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	『Cisco IOS XR Interface and Hardware Component Command Reference』
Cisco IOS XR ソフトウェアを使用した初期システム ブートアップとルータの設定情報	『Cisco IOS XR Getting Started Guide』
Cisco IOS XR AAA サービス構成情報	『Cisco IOS XR System Security Configuration Guide』 および 『Cisco IOS XR System Security Command Reference』
リモートの Craft Works Interface (CWI) クライアント管理アプリケーションからの、Cisco CRS-1 ルータ上のインターフェイスとその他のコンポーネントの設定に関する情報	『Cisco Craft Works Interface Configuration Guide』

規格

規格	タイトル
FRF.12	フレームリレー フォーラム .12
FRF.16	フレームリレー フォーラム .16

規格	タイトル
ANSI T1.617 Annex D	American National Standards Institute T1.617 Annex D
ITU Q.933 Annex A	International Telecommunication Union Q.933 Annex A

MIB

MIB	MIB リンク
FRF.16 MIB	Cisco IOS XR ソフトウェアを使用して MIB を検索およびダウンロードするには、 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。

RFC

RFC	タイトル
RFC 1294	Multiprotocol Interconnect Over Frame Relay
RFC 1315	Management Information Base for Frame Relay DTEs
RFC 1490	Multiprotocol Interconnect Over Frame Relay
RFC 1586	Guidelines for Running OSPF Over Frame Relay Networks
RFC 1604	Definitions of Managed Objects for Frame Relay Service
RFC 2115	Management Information Base for Frame Relay DTEs Using SMIv2
RFC 2390	Inverse Address Resolution Protocol
RFC 2427	Multiprotocol Interconnect Over Frame Relay
RFC 2954	Definitions of Managed Objects for Frame Relay Service
RFC 3020	RFC for FRF.16 MIB

シスコのテクニカル サポート

説明	リンク
シスコのテクニカル サポート Web サイトでは、製品、テクノロジー、ソリューション、テクニカル ヒント、ツールへのリンクなど、さまざまな技術的コンテンツを検索可能な形で提供しています。Cisco.com に登録されている場合は、次のページからログインしてさらに多くのコンテンツにアクセスできます。	http://www.cisco.com/techsupport

