



6-Port Channelized T3 Line Card の設定

この章では、Cisco 10000 シリーズ 6-Port Channelized T3 Line Card（以下、6 ポート チャネライズド T3 ラインカード）の設定手順について説明します。

6 ポート チャネライズド T3 ラインカードを Cisco 10000 シリーズ ルータに装着すると、高密度 T3 サービスの DS3 ポートが 6 つ使用可能になります。

具体的な内容は次のとおりです。

- [ソフトウェア サポート \(p.8-2\)](#)
- [デフォルト値 \(p.8-3\)](#)
- [インターフェイス構文 \(p.8-4\)](#)
- [インターフェイスの設定例 \(p.8-4\)](#)
- [非チャネライズド T3 コマンド \(p.8-7\)](#)
- [チャネライズド T3 コマンド \(p.8-11\)](#)
- [DS0 タイムスロットおよび T1 のチャネルグループ コマンド \(p.8-17\)](#)
- [チャネライズド T1 コマンド \(p.8-19\)](#)

ソフトウェア サポート

表 8-1 に、6 ポート チャネライズド T3 ライン カードをサポートするリリース系統ごとに、最低限必要な Cisco IOS リリースを示します。

表 8-1 6 ポート チャネライズド T3 ライン カードのソフトウェア サポート

必要な PRE	最低限必要な Cisco IOS リリース
PRE1	Cisco IOS Release 12.0(9)SL 以降の Cisco IOS 12.0SL リリース Cisco IOS Release 12.0(17)ST 以降の Cisco IOS 12.0ST リリース Cisco IOS Release 12.0(22)S 以降の Cisco IOS 12.0S リリース Cisco IOS Release 12.2(8)BZ 以降の Cisco IOS 12.2BZ リリース
PRE2	Cisco IOS Release 12.2(15)BX 以降の Cisco IOS 12.2BX リリース Cisco IOS Release 12.3(7)XI 以降の Cisco IOS 12.3 XI リリース Cisco IOS Release 12.2(28)SB 以降の Cisco IOS 12.2SB リリース

ハードウェアとソフトウェアの互換性の確認

Cisco 10000 シリーズ ルータ シャーシに搭載された PRE は、ルータで稼働する Cisco IOS ソフトウェアをサポートしていなければなりません。show version コマンドを使用して、搭載されている PRE のバージョンを確認してください。

機能が Cisco IOS リリースでサポートされているかどうかを確認したり、この機能に関するソフトウェア マニュアルの入手先を調べたり、ルータに搭載されたハードウェアに対する Cisco IOS ソフトウェアの最低限のソフトウェア要件を調べるには、シスコが提供している Cisco.com の Software Advisor ツールを使用してください (<http://www.cisco.com/cgi-bin/Support/CompNav/Index.pl>)。

このツールを使用しても、システム内のライン カードの互換性は検証されません。各ハードウェア ライン カード、モジュール、またはオプションに必要な最低限の Cisco IOS 要件が示されます。

このツールにアクセスするには、Cisco.com に登録する必要があります。

デフォルト値

表 8-2 に、6 ポート チャネライズド T3 ライン カードのデフォルト値、および各回線速度に適用される値を示します。また、デフォルト値を変更するために使用するコマンド、および接続のリモートエンドで値を同じ（または反対）に設定する必要があるかどうかを示します。

表 8-2 6 ポート チャネライズド T3 ライン カードのデフォルト値

コマンド名	デフォルト設定	コマンド構文	リモート エンド の設定	DS 0	T1	Ch T3	Unch T3
コントローラ コンフィギュレーション モード							
cable length	249	<code>cablelength feet</code>	—			X	X
channelized	channelized	<code>[no] channelized</code>	—			X	X
clock source	internal	<code>clock source [line internal]</code>	少なくとも片側を internal に設定			X	X
t1 clock source	internal	<code>t1 t1-number clock source [line internal]</code>	少なくとも片側を internal に設定		X		
framing	auto-detect	<code>framing [c-bit m23 auto-detect]</code>	同じ			X	
t1 framing	esf	<code>t1 t1-number framing [esf sf [hdlc-idle {0x7E 0xFF}]]</code>	同じ		X		
idle pattern	0X7e (flags)	<code>idle pattern [0x0 to 0xFF]</code>	同じ			X	
インターフェイス コンフィギュレーション モード							
crc (cyclic redundancy check)	16	<code>[no] crc [16 32]</code>	同じ	X			X
dsu bandwidth	44,210	<code>dsu bandwidth bandwidth</code>	同じ				X
dsu mode	cisco	<code>dsu mode mode</code>	同じ				X
encapsulation	HDLC	<code>encapsulation [hdlc ppp frame-relay]</code>	同じ	X			X
framing	c-bit	<code>framing [c-bit m13]</code>	同じ				X
idle character	flags (0x7e)	<code>idle character [flags marks]</code>	同じ				X
keepalive	keepalive (10 秒)	<code>keepalive</code>	同じ	X			X
mtu (maximum transmission unit)	4470	<code>mtu size</code>	同じ	X			X
scramble	スクランブルなし	<code>[no] scramble</code>	同じ				X

インターフェイス構文

コンフィギュレーション コマンドでインターフェイス番号を指定するには、表 8-3 の構文を使用して、6 ポート チャネライズド T3 ライン カードのインターフェイスを指定します。

表 8-3 6 ポート チャネライズド T3 ライン カードのインターフェイス構文

インターフェイスのタイプ	スロット	サブスロット	ポート (T3 番号)	T1 番号	チャンネルグループ番号
非チャネライズド	1 ~ 8/	0/	0 ~ 5	—	—
チャネライズド	1 ~ 8/	0/	0 ~ 5/	1 ~ 28	0 ~ 23

例：

- コントローラ コンフィギュレーション モードで T1 インターフェイス 6 を変更：

```
Router(config)# controller t3 2/0/0
Router(config-controller)# t1 6 command
```
- インターフェイス コンフィギュレーション モードで T1 インターフェイス 6、チャンネルグループ 8 を変更：

```
Router(config)# interface serial 2/0/0/6:8
Router(config-if)
```

インターフェイスの設定例

各 T3 コントローラは単一の T3 インターフェイス (フルまたはサブレート)、28 の T1 インターフェイス、またはそれ以上の個数のフラクショナル T1 として設定できます。次に、フルレートおよびサブレート T3 インターフェイス、および T1 インターフェイスとフラクショナル T1 インターフェイスを作成するための基本手順を示します。

- ステップ 1** インターフェイスを作成します。次の例では、T3 コントローラ (2/0/0 ~ 2/0/3) ごとに異なるタイプのインターフェイスが作成されます。

フルレート T3 インターフェイス

- コントローラ コンフィギュレーション モードを開始します。

```
Router(config)# controller t3 2/0/0
Router(config-controller)#
```
- フルレート T3 インターフェイスを作成するには、**no channelized** コマンドを入力して、T1 インターフェイスを除去します。

```
Router(config-controller)# no channelized
Router(config-controller)# exit
Router(config)#
```
- 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。

```
Router(config)# interface serial 2/0/0
```
- ステップ 2 に進んでください。

サブレート T3 インターフェイス

- a. コントローラ コンフィギュレーション モードを開始します。

```
Router(config)# controller t3 2/0/1
Router(config-controller)#
```

- b. サブレート T3 インターフェイスを作成するには、まずフルレート T3 インターフェイスを作成します。

```
Router(config-controller)# no channelized
Router(config-controller)# exit
Router(config)#
```

- c. 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。このモードでは、**dsu bandwidth** コマンドを使用して、サブレート T3 インターフェイスを作成できます。次の例では、帯域幅が 16,000 kbps のサブレート T3 インターフェイスが作成されます。

```
Router(config)# interface serial 2/0/1
Router(config-if)# dsu bandwidth 16000
```

- d. ステップ 2 に進んでください。

フルレート T1 インターフェイス

- a. コントローラ コンフィギュレーション モードを開始します。

```
Router(config)# controller t3 2/0/2
Router(config-controller)#
```

- b. **t1 channel group** コマンドを使用して、T1 インターフェイスを作成します。次の例では、T1 インターフェイス 1 (28 ある中の 1 つ) が、番号が 20 (0 ~ 23 の任意の番号) の単一チャンネルグループで構成されるように定義されます。このチャンネルグループには、24 のすべての DS0 タイムスロットが含まれます。

```
Router(config-controller)# t1 1 channel-group 20 timeslots 1-24
```

- c. 作成したチャンネルグループに対して、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。

```
Router(config)# interface serial 2/0/2/1:20
```

- d. ステップ 2 に進んでください。

フラクショナル T1 インターフェイス

- a. コントローラ コンフィギュレーション モードを開始します。

```
Router(config)# controller t3 2/0/3
Router(config-controller)#
```

- b. **t1 channel group** コマンドを使用して、フラクショナル T1 インターフェイスを作成します。次の例では、T1 インターフェイス 3 (28 ある中の 1 つ) が、番号 19、20、および 21 (0 ~ 23 の番号が有効) の 3 つのチャンネルグループで構成されるように定義されます。これらのチャンネルグループには、合計 24 の DS0 タイムスロットが含まれています。各チャンネルグループは個別のインターフェイスを表します。

```
Router(config-controller) t1 3 channel-group 19 timeslots 1-6, 10
Router(config-controller) t1 3 channel-group 20 timeslots 7,8,9
Router(config-controller) t1 3 channel-group 21 timeslots 11-24
```

- c. 1 つのチャンネルグループに対して、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。次に例を示します。

```
Router(config)# interface serial 2/0/3/3:19
```

- d. ステップ 2 に進んでください。

■ インターフェイスの設定例

- ステップ 2** カプセル化方式を入力します。次に、フレームリレー カプセル化を使用する場合のコマンドを示します。PPP または HDLC も選択できます。

```
Router(config-if)# encapsulation frame relay
```

- ステップ 3** システム上で IP ルーティングがイネーブルである場合は、IP アドレスおよびサブネット マスクを割り当てます。次に例を示します。

```
Router(config-if)# ip address 172.16.32.49 255.255.0.0
```

- ステップ 4** ルーティング プロトコルをイネーブルにするために必要なコンフィギュレーション サブコマンドをすべて追加し、インターフェイス回線の特性を設定します。

- ステップ 5** シャットダウン ステートをアップに変更して、インターフェイスをイネーブルにします。

```
Router(config-if)# no shutdown
```

- ステップ 6** コンフィギュレーション サブコマンドをすべて入力して、設定を完了したら、**ctrl-z** を押して、コンフィギュレーション モードを終了します。

- ステップ 7** 新しい設定を NVRAM (不揮発性 RAM) に書き込むには、次のように入力します。

```
Router# copy running-config startup-config
```

作成したインターフェイス設定は、該当する Cisco IOS コンフィギュレーション コマンドを使用して、いつでも変更できます。

非チャネライズド T3 コマンド

デフォルトでは、6 ポート チャネライズド T3 ラインカードの T3 インターフェイスは、28 の T1 インターフェイスにチャネル化されます。フルレートまたはサブレート T3 インターフェイスを作成するには、T3 インターフェイスを非チャネル化する必要があります。ここでは、フルレートおよびサブレート T3 インターフェイスを作成、カスタマイズ、およびテストする場合に使用するコマンドについて説明します。内容は次のとおりです。

- T3 のコントローラ コンフィギュレーション モードの開始 (p.8-7)
- T3 インターフェイスの非チャネライズドとしての設定 (p.8-7)
- サブレート T3 の実装 (p.8-8)
- フレーミング タイプの設定 (p.8-8)
- DSU モードの指定 (p.8-8)
- ス克蘭プリングのイネーブル化 (p.8-9)
- アイドル文字の指定 (p.8-9)
- BER テストの実行 (p.8-9)

非チャネライズド T3 を設定する場合は、「チャネライズド T3 コマンド」(p.8-11) に記載されている `cablelength`、`clock source`、`mdl`、`equipment`、および `loopback` コマンドも使用できます。

T3 のコントローラ コンフィギュレーション モードの開始

非チャネライズド T3 インターフェイスを作成するには、まず、設定する T3 コントローラに対して、コントローラ コンフィギュレーション モードを開始する必要があります。

```
controller T3 slot/sub-slot/port
```

次に、コントローラ コンフィギュレーション モードを開始する例を示します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)#
```

T3 インターフェイスの非チャネライズドとしての設定

T3 インターフェイスを非チャネライズド (クリア チャネル) として設定するには、`no channelized` コマンドを入力します。

```
[no] channelized
channelized
```



注意

`no channelized` コマンドを使用すると、チャネライズド T3 インターフェイスからすべてのチャネルグループが削除されます。チャネルグループが設定されている場合は、このコマンドを使用するときに注意してください。

インターフェイスをデフォルトに戻すには、`channelized` コマンドを使用します。T3 インターフェイスのデフォルト値は、チャネライズドです。

次の例では、非チャネライズド T3 インターフェイスが作成されます。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# no channelized
```

サブレート T3 の実装

サブレート T3 を実装するには、非チャネライズド T3 インターフェイスの帯域幅を指定します。そのためには、インターフェイス コンフィギュレーション モードで **dsu bandwidth bandwidth** コマンドを使用します。

```
dsu bandwidth bandwidth
[no] bandwidth bandwidth
```

ここで、*bandwidth* は 0 ~ 44210 kbps の数値です。

デフォルト帯域幅は 44210 kbps です。

デフォルトの帯域幅に戻すには、このコマンドの **no** 形式を使用します。

値を指定すると、現在の DSU (データ サービス ユニット) モードのタイムスロット サイズに基づいて、許容される最も近い値に帯域幅が設定されます。

このコマンドを使用するには、接続のリモート エンドに、**dsu mode** コマンドでサポートされている PA-T3 または PA-2T3 ポート アダプタまたは T3 DSU が搭載された、Cisco 7200 ルータまたは Cisco 7500 ルータを配置する必要があります。

次の例では、帯域幅として 16000 kbps が指定されます。

```
Router(config)# interface serial 1/0/0
Router(config-if)# dsu bandwidth 16000
```

フレーミング タイプの設定

非チャネライズド T3 コントローラのフレーミング タイプを指定するには、**framing** コマンドを使用します。

```
framing [c-bit | m13]
[no] framing [c-bit | m13]
```

デフォルトのフレーミング タイプは C ビットです。

デフォルトのフレーミング タイプに戻すには、このコマンドの **no** 形式を使用します。

次の例では、フレーミングが m13 に設定されます。

```
Router(config)# interface serial 1/0/0
Router(config-if)# framing m13
```

DSU モードの指定

選択した T3 インターフェイスに対して DSU モードを指定するには、インターフェイス コンフィギュレーション モードで **dsu mode** コマンドを使用します。このコマンドは、製造元独自の多重化方式をエミュレートするようにライン カードを設定します。

```
dsu mode [Adtran | cisco | Digital-link | Kentrox | Larscom | verilink-highbit |
verilink-lowbit]
[no] dsu mode
```

デフォルトの DSU モードは **cisco** です。

DSU モードをデフォルトに戻すには、このコマンドの **no** 形式を使用します。

次の例では、DSU モードが **cisco** に設定されます。

```
Router(config)# interface serial 1/0/0
Router(config-if)# dsu mode cisco
```

スクランプリングのイネーブル化

非チャネライズド T3 インターフェイスでスクランプリングを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 **scramble** コマンドを使用します。

```
scramble
[no] scramble
```

このコマンドのデフォルト設定は、**no scramble**（スクランプリングがディセーブル）です。

リンクの両側で、スクランプリング設定を同じにする必要があります。

次の例では、指定された T3 インターフェイスでスクランプリングがイネーブルになります。

```
Router(config)# interface serial 1/0/0
Router(config-if)# scramble
```

アイドル文字の指定

非チャネライズド T3 インターフェイスで、HDLC パケット間に送信する特定の文字を設定するには、インターフェイス コンフィギュレーション モードで **idle character** コマンドを使用します。

```
idle-character [flags | marks]
[no] idle-character [flags | marks]
```

各値は次のとおりです。

- **flags** — アイドル文字に 0x7e を設定します。
- **marks** — アイドル文字に 0xff（すべて）を設定します。

デフォルトのアイドル文字は 0x7e です。

アイドル文字をデフォルトに戻すには、このコマンドの **no** 形式を使用します。

次の例では、アイドル文字が **flags** に設定されます。

```
Router(config)# interface serial 1/0/0
Router(config-if)# idle-character flags
```



(注) marks (0xff) が中断信号として解釈されるシステムがあるため、flags (0x7e) を推奨します。

BER テストの実行

Bit Error Rate (BER; ビット誤り率) テストを実行するように、非チャネライズド T3 インターフェイスを設定できます。BER テストは、現場でケーブルを調べて、信号問題を解決する場合に使用できます。非チャネライズド T3 インターフェイスから BER テスト パターンを送信するには、次のインターフェイス コンフィギュレーション コマンドを使用します。

```
bert [errors number | pattern pattern] interval time
[no] bert
```

各値は次のとおりです。

- **errors number** — 1 ~ 255
- **pattern pattern** — 値は次のとおり
 - 0s — すべてゼロの反復テストパターン (00000...)
 - 1s — すべて 1 の反復テストパターン (11111...)
 - 2^15 — 疑似ランダム O.151 テストパターン (32,768 ビット長)
 - 2^20-O153 — 疑似ランダム O.153 テストパターン (1,048,575 ビット長)
 - QRSS-2^20 — 疑似ランダム QRSS O.151 テストパターン (1,048,575 ビット長)
 - 2^23 — 疑似ランダム O.151 テストパターン (8,388,607 ビット長)
- **interval time** — 1 ~ 1440 分

no bert コマンドを使用すると、いつでも BER テストを終了できます。

詳細については、オンラインで『[Cisco 10000 Series Internet Router Troubleshooting Guide](#)』を参照してください。

例：

- T3 インターフェイス 1/0/0 を介して、BER テスト疑似ランダムパターン 2^20 を 5 分間送信します。


```
Router(config)# interface serial 1/0/0
Router(config-if)# bert pattern 2^20 interval 5
```
- T3 インターフェイス 1/0/0 を介して、すべて 1 の反復パターンを 1440 分間送信します。


```
Router(config)# interface serial 1/0/0
Router(config-if)# bert pattern 1s interval 1440
```

チャネライズド T3 コマンド

デフォルトでは、6 ポート チャネライズド T3 ラインカードの T3 インターフェイスは、28 の T1 インターフェイスにチャネル化されます。ここでは、チャネライズド T3 インターフェイスをカスタマイズして、テストする場合に使用するコマンドについて説明します。内容は次のとおりです。

- [T3 のコントローラ コンフィギュレーションモードの開始 \(p.8-11\)](#)
- [T3 インターフェイスのチャネライズドとしての設定 \(p.8-12\)](#)
- [ケーブル長の指定 \(p.8-12\)](#)
- [フレーミングタイプの設定 \(p.8-13\)](#)
- [MDL メッセージの入力 \(p.8-13\)](#)
- [アイドルパターンの指定 \(p.8-14\)](#)
- [クロックソースの設定 \(p.8-15\)](#)
- [ループバックモードの設定 \(p.8-15\)](#)
- [機器ループバックの実行 \(p.8-16\)](#)

加入者トラフィックにチャネルを使用するには、T1 および DS0 コンポーネントを設定する必要があります。詳細については、「[DS0 タイムスロットおよび T1 のチャネルグループ コマンド](#)」(p.8-17) を参照してください。

図 8-1 に、チャネル化によって実現される、設定可能なインターフェイス帯域幅レベルを示します。

T3 のコントローラ コンフィギュレーションモードの開始

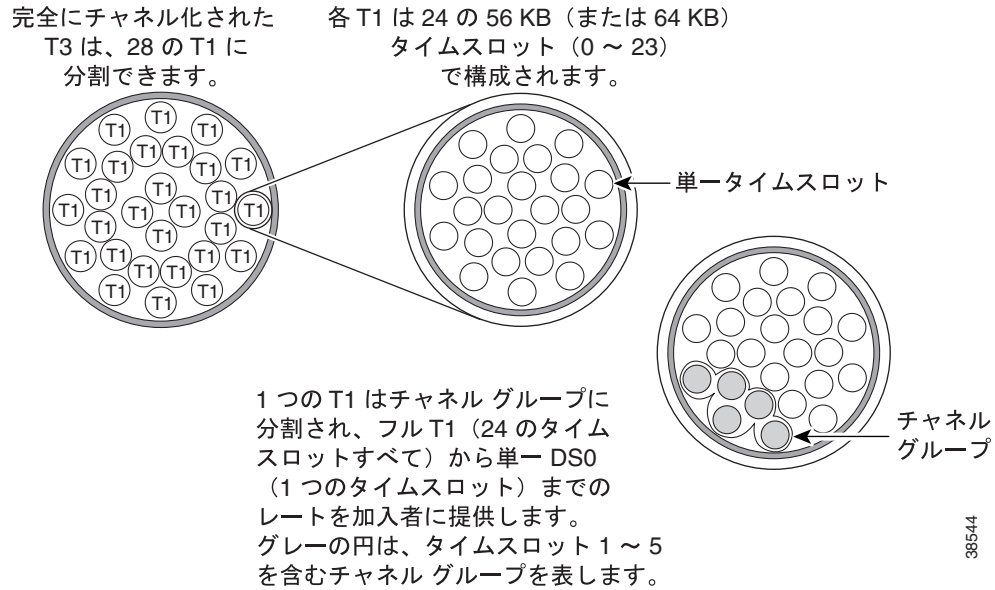
T3 インターフェイスに関連するアクションを実行するために、コントローラ コンフィギュレーションモードを開始しなければならないことがあります。たとえば、非チャネライズド T3 インターフェイスをチャネライズドに戻したり、Maintenance Data Link (MDL) メッセージを設定したりするには、このモードを開始します。

```
controller T3 slot/sub-slot/port
```

次に、コントローラ コンフィギュレーションモードを開始する例を示します。

```
Router(config)# controller T3 1/0/0  
Router(config-controller)#
```

図 8-1 T3 のチャネル化



T3 インターフェイスのチャネライズドとしての設定

T3 インターフェイスはデフォルトでチャネル化されます。非チャネル化されたインターフェイスの設定を変更する場合は、**channelized** コマンドを使用します。

```
[no] channelized
channelized
```



注意

no channelized コマンドを使用すると、チャネライズド T3 インターフェイスからすべてのチャネルグループが削除されます。チャネルグループが設定されている場合は、このコマンドを使用するときに注意してください。

次に、チャネライズド T3 インターフェイスを作成する例を示します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# channelized
```

ケーブル長の指定

T3 コントローラのケーブル長を指定するには、**cablelength** コマンドを使用します。

```
cablelength feet
[no] cablelength feet
```

ここで、*feet* は 0 ~ 450 の値です。

デフォルト値は 249 です。

デフォルトのケーブル長に戻すには、このコマンドの **no** 形式を使用します。

次の例では、ケーブル長が 40 フィートに設定されます。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# cablelength 40
```



(注)

このコマンドを使用すると、インピーダンス照合やパルスシェーピングについて、2 つの設定（0 ～ 249 フィートのケーブル長に対する設定と、250 フィートを超えるケーブル長に対する設定）のいずれかが使用されます。入力した正確な値は、コンフィギュレーションファイルに保存されます。

フレーミング タイプの設定

チャネライズド T3 コントローラのフレーミング タイプを指定するには、**framing** コマンドを使用します。

```
framing [c-bit | m23 | auto-detect]
[no] framing [c-bit | m23 | auto-detect]
```

デフォルトは **auto-detect** です。

フレーミング タイプをデフォルトに戻すには、このコマンドの **no** 形式を使用します。

次のように、遠端からフレーミング タイプを検出して、同じフレーミング タイプを送信するように、6 ポート チャネライズド T3 ラインカードに指示することができます。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# framing auto-detect
```

MDL メッセージの入力

チャネライズド T3 インターフェイスに MDL メッセージ（ANSI T1.107a-1990 仕様で定義）を設定できます。



(注)

MDL メッセージがサポートされるのは、T3 フレーミングが C ビット パリティに設定されている場合のみです（「[フレーミング タイプの設定](#)」 [p.8-13] を参照）。

MDL メッセージを設定するには、**mdl** コマンドを使用します。

```
mdl {transmit {path | idle-signal | test-signal} | string {eic | lic | fic | unit | pfi |
port | generator} id_string}
[no] mdl {transmit {path | idle-signal | test-signal} | string {eic | lic | fic | unit |
pfi | port | generator} id_string}
```

各値は次のとおりです。

- **transmit path** — MDL パス メッセージの送信をイネーブルにします。
- **transmit idle-signal** — MDL アイドル信号メッセージの送信をイネーブルにします。
- **eic** — 機器 ID コード（最大 10 文字）
- **lic** — 場所 ID コード（最大 11 文字）
- **fic** — フレーム ID コード（最大 10 文字）
- **unit** — ユニット ID コード（最大 6 文字）
- **pfi** — MDL パス メッセージに含まれるファシリティ ID コード（最大 38 文字）

- **port** — MDL アイドル信号メッセージに含まれる（アイドル信号の開始元となる）機器のポート（最大 38 文字）
- **generator** — MDL テスト信号メッセージに含まれるジェネレータ番号（最大 38 文字）

デフォルトでは、MDL メッセージは設定されません。

MDL メッセージを削除するには、このコマンドの **no** 形式を使用します。

次に、MDL メッセージの設定例を示します。

- MDL パス メッセージの送信をイネーブルにします。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# mdl transmit path
```
- MDL アイドル信号メッセージの送信をイネーブルにします。

```
Router(config-controller)# mdl transmit idle-signal
```
- MDL テスト信号メッセージの送信をイネーブルにします。

```
Router(config-controller)# mdl transmit test-signal
```
- 機器 ID コードを入力します。

```
Router(config-controller)# mdl string eic router A
```
- 場所 ID コードを入力します。

```
Router(config-controller)# mdl string lic test network
```
- フレーム ID コードを入力します。

```
Router(config-controller)# mdl string fic building b
```
- ユニット ID コードを入力します。

```
Router(config-controller)# mdl string unit abc
```
- ファシリティ ID コードを入力します。

```
Router(config-controller)# mdl string pfi string
```
- MDL アイドル信号メッセージに格納して送信するポート番号を入力します。

```
Router(config-controller)# mdl string port string
```
- MDL テスト信号メッセージに格納して送信するジェネレータ番号を入力します。

```
Router(config-controller)# mdl string generator string
```

アイドルパターンの指定

チャネライズド T3 インターフェイスに属する未設定のすべてのタイムスロットについて、HDLC パケット間に送信される特定のパターンを設定できます。そのためには、**idle pattern** コマンドを使用します。

```
idle pattern patterns
[no] idle pattern patterns
```

ここで、*patterns* は 0x0 ~ 0xff（16 進数）または 0 ~ 255（10 進数）の値です。この値は 16 進数または 10 進数で入力できます。0 ~ 254 の値を指定すると、アイドルパターンは HDLC flags（0x7e）に設定され、255 の値を指定すると、0xff（すべて 1）に設定されます。



(注) marks (0xff) が中断信号として解釈されるシステムがあるため、flags (0x7e) を推奨します。

デフォルトのアイドルパターンは 0x7e です。

アイドルパターンをデフォルト値に戻すには、このコマンドの **no** 形式を使用します。

例 :

- 16 進数のアイドルパターンを設定します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# idle pattern 0x10
```
- 10 進数のアイドルパターンを設定します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# idle pattern 23
```

クロック ソースの設定

選択した T3 コントローラに内部クロック ソースまたは回線クロック ソースを設定するには、プロンプトで **clock source** コマンドを使用します。このコマンドは、コントローラ コンフィギュレーション モードで設定します。

```
clock source {internal | line}
```

各値は次のとおりです。

- **internal** — 内部クロック ソースを使用するように指定します。
- **line** — ネットワーク クロック ソースを使用するように指定します。

デフォルトは内部クロック ソースです。

次の例では、回線クロック ソースを使用するように T3 コントローラに指示します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# clock source line
```

ループバック モードの設定

T3 コントローラにループバック モードを設定するには、**loopback** コマンドを使用します。

```
loopback [local | network | remote]
[no] loopback [local | network | remote]
```

ローカルループバックとネットワークループバックは同じです。

ループバックを取り消すには、このコマンドの **no** 形式を使用します。

loopback コマンドの詳細については、オンラインで『[Cisco 10000 Series Internet Router Troubleshooting Guide](#)』を参照してください。

例 :

- T3 コントローラにローカルループバックを設定します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# loopback local
```

- T3 コントローラにリモート ループバックを設定します。
Router(config)# controller T3 1/0/0
Router(config-controller)# loopback remote

機器ループバックの実行

リモート機器と組み合わせてループバックを実行するには、**equipment loopback** コマンドを使用します。

```
equipment [customer | network] loopback  
[no] equipment [customer | network] loopback
```

各値は次のとおりです。

- **customer** — リモート T3 機器からリモート T3 ループバック コマンドに応答できるように、ラインカードを設定します。
- **network** — リモート T3 ループバック コマンドを無視するように、ラインカードを設定します。

ループバックを終了する場合は、このコマンドの **no** 形式を使用します。

loopback コマンドの詳細については、オンラインで『[Cisco 10000 Series Internet Router Troubleshooting Guide](#)』を参照してください。

次の例では、機器ネットワーク ループバックが設定されます。

```
Router(config)# controller T3 1/0/0  
Router(config-controller)# equipment network loopback
```


DS0 タイムスロットおよび T1 のチャネルグループコマンド

各チャネライズド T3 は、それぞれ 24 のタイムスロット (DS0) に分割される、28 個の T1 で構成されます。加入者トラフィックをサポートするインターフェイスを作成するには、タイムスロットをチャネルグループに結合する必要があります。チャネルグループを作成するには、**t1 channel-group** コマンドを使用します。

チャネルグループには次の特性があります。

- チャネルグループはインターフェイスです。
- チャネルグループのサイズは、タイムスロット 1～24 個分に設定できます。
 - 24 のタイムスロットで構成されるチャネルグループは、T1 インターフェイスです。
 - 24 未満のタイムスロットで構成されるチャネルグループは、フラクショナル T1 インターフェイスとして記述できます。
- 24 のタイムスロットからなる各グループは、複数のフラクショナル T1 インターフェイスに分割できます。
- 1 つのチャネルグループを複数の T1 に含めることはできません。

図 8-1 に、6 ポート チャネライズド T3 ラインカードのチャネル化の仕組みを示します。

次のコマンドを使用して、論理チャネルグループを作成します。

```
t1 t1-number channel-group Channel-group-number timeslots list-of-timeslots [speed {56 | 64}]
[no] t1 t1-number channel-group channel-group-number
```

各値は次のとおりです。

- **t1-number** — 1～28 の T1 インターフェイス番号
- **channel-group channel-group-number** — 0～23 の任意の番号を持つチャネルグループを指定します。
- **timeslots list-of-timeslots** — 1～24、または 1～24 のサブ範囲の組み合わせを指定できます。範囲を指定するには、ハイフン、カンマ、または両方の組み合わせを使用します。1 つのタイムスロットが 1 つの DS0 に相当します。次の例を参照してください。
- **speed {56 | 64}** — タイムスロットの速度を 56 kbps または 64 kbps として指定するオプションの引数。デフォルトは 64 です (56 kbps のタイムスロットは SF [スーパーフレーム] フレーミングに対応し、通常、B8ZS をサポートしない古い T1 機器で使用します)。

論理チャネルグループを削除するには、このコマンドの **no** 形式を使用します。

次に、**t1 channel-group** コマンドの使用例を示します。

- 次の例では、T1 インターフェイス 3 にチャネルグループ 20 が含まれ、9 つのチャネライズドタイムスロットで構成されています。

```
Router(config)# controller t3 1/0/0
Router(config-controller)# t1 3 channel-group 20 timeslots 1-8, 10
```

このチャネルグループに対してインターフェイス コンフィギュレーション モードを開始するには、次のように入力します。

```
Router(config)# interface serial 1/0/0/3:20
```

- 次の例では、T1 インターフェイス 4 にチャネルグループ 18 が含まれ、24 のすべてのタイムスロットで構成されていて、フル T1 インターフェイスを形成しています。

```
Router(config)# controller t3 1/0/0
Router(config-controller)# t1 4 channel-group 18 timeslots 1-24
```

■ DS0 タイムスロットおよび T1 のチャンネルグループコマンド

このチャンネルグループに対してインターフェイス コンフィギュレーション モードを開始するには、次のように入力します。

```
Router(config)# interface serial 1/0/0/4:18
```

- 次の例では、T1 インターフェイス 5 に 3 つのチャンネルグループ、および合計 11 のタイムスロットが含まれています。

```
Router(config)# controller t3 1/0/0
Router(config-controller)# t1 5 channel-group 19 timeslots 1-6
Router(config-controller)# t1 5 channel-group 20 timeslots 10
Router(config-controller)# t1 5 channel-group 21 timeslots 7-9, 24
```

- 次の例では、チャンネルグループ 20 が削除されます。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# no t1 1 channel-group 20
```

チャネライズド T1 コマンド

T3 インターフェイスのコントローラ コンフィギュレーション モードから、T1 インターフェイスの特性を変更するコマンドを入力できます。ここで示されるコマンドの用途は、次のとおりです。

- フレーミング フォーマットの設定 (p.8-19)
- イエロー アラームの制御 (p.8-19)
- クロック ソースの設定 (p.8-20)
- FDL の設定 (p.8-20)
- BER テストの設定 (p.8-21)
- ループバック モードの設定 (p.8-22)

T1 インターフェイスを設定したら、インターフェイス コンフィギュレーション モードを開始して、カプセル化、ルーティング、およびその他の命令を追加できます。次に例を示します。

```
Router(config)# interface serial 1/0/0/4:18
```

フレーミング フォーマットの設定

T1 インターフェイス フレーミング フォーマットを指定するには、次のコマンドを使用します。

```
t1 t1-number framing {esf | sf [hdlc-idle {0x7E | 0xFF}]}
[no] t1 t1-number framing {esf | sf [hdlc-idle {0x7E | 0xFF}]}
```

各値は次のとおりです。

- **t1-number** — 1 ～ 28 の T1 インターフェイス番号
- **framing** — 拡張スーパーフレーム (Extended Super Frame) または SF。SF の hdlc-idle は 0x7E または 0xFF に設定できます。
- **hdlc-idle** オプションを使用すると、T1 インターフェイスのアイドル パターンを 0x7e (デフォルト) または 0xff に設定できます。

デフォルトのフレーミング フォーマットは拡張スーパーフレームです。

フレーミングをデフォルト値に戻すには、このコマンドの **no** 形式を使用します。

例：

- T1 インターフェイス 6 に SF フレーミング フォーマットを設定します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# t1 6 framing sf
```

- T1 インターフェイス 16 に拡張スーパーフレーム フレーミング フォーマットを設定します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# t1 16 framing esf
```

イエロー アラームの制御

イエロー アラームの検出または生成をオンまたはオフにするには、**t1 yellow** コマンドを使用します。

```
t1 t1-number yellow {detection | generation}
[no] t1 t1-number yellow {detection | generation}
```

各値は次のとおりです。

- **t1-number** — 1 ～ 28 の T1 インターフェイス

■ チャネライズド T1 コマンド

- **detection** — 障害がある場合は、リモート デバイスからインターフェイスに通知され、IOS によってコンソールにメッセージが送信されます。
- **generation** — 障害がある場合は、インターフェイスからリモート デバイスに通知され、IOS によってコンソールにメッセージが送信されます。

デフォルト速度に 64 を使用している フル T1 インターフェイス (24 のタイムスロット) に SF フレーミングを選択した場合は、**no t1-number yellow detection** コマンドを使用して、イエロー アラーム検出をオフにすることを検討してください。SF フレーミングの場合、イエロー アラームが誤って検出されることがあります。

次の例では、T1 インターフェイス 1 にイエロー検出が設定されます。

```
Router(config)# controller t3 1/0/0
Router(config-controller)# t1 1 yellow detection
```

クロック ソースの設定

T1 インターフェイスに内部クロック ソースまたは回線 (ネットワーク) クロック ソースを設定するには、**controller** コマンドを使用します。

```
t1 t1-number clock source {internal | line}
```

各値は次のとおりです。

- **t1-number** — 1 ~ 28 の T1 インターフェイス
- **internal** — 内部クロック ソースを使用するように指定します。
- **line** — ネットワーク クロック ソースを使用するように指定します。

デフォルトは内部クロック ソースです。

次の例では、回線からクロック ソースを取得するようにインターフェイスに指示します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# t1 1 clock source line
```

FDL の設定

ANSI T1.403 仕様に従って、T1 接続の両端で Facility Data Link (FDL; ファシリティ データ リンク) を介してパフォーマンス レポートを 1 秒間送信するように設定できます。そのためには、次のコマンドを使用します。

```
t1 t1-number fdl ansi
[no] t1 t1-number fdl ansi
```

ここで、**t1-number** は 1 ~ 28 の T1 インターフェイスです。

この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。



(注)

このコマンドを使用できるのは、T1 フレーミングが拡張スーパーフレームの場合のみです。

次の例では、FDL がイネーブル化されます。

```
Router(config)# controller t3 1/0/0
Router(config-controller)# t1 2 fdl ansi
```

BER テストの設定

独立した BER テストを実行するように、各 T1 インターフェイスを設定できます。BER テストは、現場でケーブルを調べて、信号問題を解決する場合に使用します。T1 インターフェイスから BER テストパターンを送信するには、次のコマンドを使用します。

```
t1 t1-number bert pattern pattern interval time [unframed]
[no] t1 t1-number bert
```

各値は次のとおりです。

- *t1-number* — 1 ～ 28 の T1 インターフェイス番号
- *time* — 1 ～ 14,400 分
- *pattern* — 値は次のとおり
 - 0s — すべてゼロの反復テストパターン (00000...)
 - 1s — すべて 1 の反復テストパターン (11111...)
 - 2¹¹ — 疑似ランダムテストパターン (2048 ビット長)
 - 2¹⁵ — 疑似ランダム O.151 テストパターン (32,768 ビット長)
 - 2²⁰-O153 — 疑似ランダム O.153 テストパターン (1,048,575 ビット長)
 - 2²⁰-QRSS — 疑似ランダム QRSS O.151 テストパターン (1,048,575 ビット長)
 - 2²³ — 疑似ランダム O.151 テストパターン (8,388,607 ビット長)
 - alt-0-1 — ゼロ (0) および 1 が交互に現れる反復テストパターン (01010101 など)
- **unframed** — T1 フレーミングおよびペイロード ビットを含む T1 帯域幅全体を BER テストパターンで使用します。**unframed** を省略すると、T1 フレーミング コマンドの設定に応じて T1 は SF または拡張スーパーフレーム フレーミングになり、BER テストパターンは T1 ペイロード ビットのみを占有します。



(注) 各 T3 で実行できる BER テストは、一度に 1 つのみです。

このコマンドの **no** 形式を使用すると、いつでも BER テストを終了できます。

詳細については、オンラインで『[Cisco 10000 Series Internet Router Troubleshooting Guide](#)』を参照してください。

例：

- T1 インターフェイス 10 を介して、BER テスト疑似ランダムパターン 2²⁰ を 5 分間送信します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# t1 10 bert pattern 2^20 interval 5 unframed
```

- T1 インターフェイス 10 を介して、すべて 1 の反復パターンを 14,400 分間 (240 時間) 送信します。

```
Router(config)# controller T3 1/0/0
Router(config-controller)# t1 10 bert pattern 1s interval 14400 unframed
```

ループバック モードの設定

T1 インターフェイスを設定するときに問題が発生した場合は、コントローラ コンフィギュレーション モードで次のコマンドを使用して、ライン カードのトラブルシューティングを実行できます。

```
t1 t1-number loopback [local | network {line | payload} | remote [line [fdl {ansi | bellcore} | inband] | payload [fdl | ansi]]]
[no] t1 t1-number loopback
```

各値は次のとおりです。

- **t1-number** — 1 ~ 28 の T1 インターフェイス
- **local** — T1 フレーマでルータ出力データをルータ方向にループバックさせて、Alarm Indication Signal (AIS; アラーム表示信号) をネットワークに送信します。
- **network {line | payload}** — データをネットワーク方向にループバックさせて、HDLC コントローラにローカルループバックを自動設定するか (**line**)、またはペイロードデータをネットワーク方向にループバックさせて、HDLC コントローラにローカルループバックを自動設定します (**payload**)。
- **remote line fdl {ansi | bellcore}** — 反復する 16 ビットの拡張スーパーフレーム データ リンク キーワードをリモートエンドに送信して、リモートエンドでネットワーク ラインループバックを開始するように要求します。ANSI またはベルコア キーワードを指定できます。



(注) ループバック コードは ANSI T1.404-1989 Table 3 (Assigned Bit-Oriented ESF Data-Link Messages) で定義されています。

リモート回線 FDL ANSI ループバックを実行する場合、6 ポート チャネライズド T3 ラインカードは次のコードを送信します。

- 回線ループバック アクティブ化コード : 0 000111 01111111
- 回線ループバック非アクティブ化コード : 0 010010 01111111

リモート回線 FDL ベルコア (SmartJack) ループバックを実行する場合、6 ポート チャネライズド T3 ラインカードは次のコードを送信します。

- ネットワーク使用 (ループバック アクティブ化) コード : 0 001001 01111111
- ネットワーク使用 (ループバック非アクティブ化) コード : 0 010010 01111111

- **remote line inband** — 反復する帯域内パターンをリモートエンドに送信して、リモートエンドでネットワーク回線ループバックを開始するように要求します。帯域内ループバック要求は、T1 内のすべてのデータをループ要求パターンで上書きします。リモートエンドがこのコードに応答するのは、パターンが 5 秒間以上継続して受信された場合のみです。
 - 反復する帯域内ループバック アクティブ化コード : 00001
 - 反復する帯域内ループバック非アクティブ化コード : 001
- **remote payload [fdl] [ansi]** — 反復する 16 ビット拡張スーパーフレーム データ リンク コードワードをリモートエンドに送信して、リモートエンドでネットワーク ペイロードループバックを開始するように要求します。**fdl** および **ansi** を使用すると、T1 チャンネルでリモート ペイロード FDL ANSI ビットループバックがイネーブルになります。



(注) ループバック コードは ANSI T1.404-1989 Table 3 (Assigned Bit-Oriented ESF Data-Link Messages) で定義されています。

リモートペイロード FDL ANSI ループバックを実行する場合、6 ポートチャネライズド T3 ラインカードは次のコードを送信します。

- ペイロードループバック アクティブ化コード : 0 001010 011111111
- ペイロードループバック非アクティブ化コード : 0 011001 011111111

ループバックを終了する場合は、このコマンドの **no** 形式を使用します。

このコマンドの詳細については、オンラインで『[Cisco 10000 Series Internet Router Troubleshooting Guide](#)』を参照してください。

例 :

- T3 コントローラに T1 インターフェイス 1 でのローカルループバックを設定します。

```
Router(config)# controller t3 1/0/0
Router(config-controller)# t1 1 loopback local
```
- T3 コントローラに、T1 インターフェイス 1 での FDL ANSI ループバックを設定します。

```
Router(config)# controller t3 1/0/0
Router(config-controller)# t1 1 loopback remote line fdl ansi
```

■ チャネライズド T1 コマンド