



モデル駆動型テレメトリ

- [モデル駆動型テレメトリ \(1 ページ\)](#)

モデル駆動型テレメトリ

モデル駆動型テレメトリは、YANG モデル化されたデータをデータ コレクタにストリーミングするためのメカニズムを提供します。このモジュールでは、モデル駆動型テレメトリについて説明し、テレメトリ リモート プロシージャ コール (RPC) の例を示します。

モデル駆動型テレメトリの前提条件

- テレメトリを使用する際に必要なデータを理解して定義するには、YANG に関する知識が必要です。
- XML、XML 名前空間、および XML XPath の知識。
- IETF テレメトリ仕様で定義されている標準および原則の知識。
- `urn:ietf:params:netconf:capability:notification:1.1` 機能は、hello メッセージでリストする必要があります。この機能は、IETF テレメトリをサポートするデバイスでのみアドバタイズされます。
- NETCONF-YANG がデバイス上で設定済みであり稼働している必要があります。



(注) テレメトリを機能させるには、NETCONF-YANG または gNXI を設定する必要があります。プラットフォームが gNXI をサポートしていない場合は、NETCONF が使用されていない場合でも、NETCONF を設定する必要があります。NETCONF-YANG の設定の詳細については、「NETCONF プロトコル」モジュールを参照してください。gNXI の詳細については、「[gNMI Protocol](#)」[英語]を参照してください。

show platform software yang-management process コマンドを使用して、次のプロセスが実行中であることを確認します。

```
Device# show platform software yang-management process

confd : Running
nesd : Running
syncfd : Running
ncsshd : Running
dmiauthd : Running
nginx : Running
ndbmand : Running
pubd : Running
gnmib : Running
```



- (注) プロセス **pubd** はモデル駆動型テレメトリ プロセスであり、これが実行していない場合にはモデル駆動型テレメトリは機能しません。

次の表に、各デバイス管理インターフェイス (DMI) プロセスの詳細を示します。

表 1: フィールドの説明

デバイス管理インターフェイスプロセス名	主要な役割
confd	コンフィギュレーションデーモン
nesd	ネットワーク要素シンクロナイザデーモン
syncfd	同期デーモン (実行状態と対応するモデル間の同期を維持)
ncsshd	NETCONF セキュアシェル (SSH) デーモン
dmiauthd	DMI 認証デーモン。
nginx	NGINX Web サーバ。RESTCONF の Web サーバとして機能します。
ndbmand	NETCONF データベースマネージャ
pubd	モデル駆動型テレメトリに使用されるパブリケーションマネージャおよびパブリッシャ
gnmib	GNMI プロトコルサーバ。

NETCONF 固有の前提条件

- NETCONF とその使用方法に関する次の知識。

- NETCONF セッションの確立。
- hello および機能メッセージの送受信。
- 確立された NETCONF セッションによる YANG XML RPC の送受信詳細については、『[Configure NETCONF / YANG and Validate Example for Cisco IOS XE 16.x Platforms](#)』を参照してください。

NETCONF の有効化と検証

NETCONF の機能を確認するには、有効なユーザ名とパスワードを使用してデバイスへの SSH 接続を作成し、デバイスの機能を含む hello メッセージを受信します。

```
Device:~ USER1$ ssh -s cisco1@172.16.167.175 -p 830 netconf
cisco1@172.16.167.175's password: cisco1
```

```
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability
.
.
.
</capabilities>
<session-id>2870</session-id></hello>]]>]]>
```

Use < ^C > to exit

hello メッセージに対して正常な応答を受信すると、NETCONF を使用する準備が整います。

RESTCONF 固有の前提条件

- RESTCONF とその使用方法に関する次の知識（RESTCONF を使用してサブスクリプションを作成する場合）。
- RESTCONF がデバイスで設定されている必要があります。
- RESTCONF は、RESTCONF [RFC 8040](#) に準拠した、正しい形式の Uniform Resource Identifier (URI) を送信する必要があります。

RESTCONF の有効化と検証

適切なクレデンシャルと次の URI を使用して、RESTCONF を検証します。

```
Operation: GET
Headers:
" Accept: application/yang-data.collection+json, application/yang-data+json,
application/yang-data.errors+json
" Content-Type: application/yang-data+json
Returned Output (omitted for breverity):
{
```

```

"ietf-restconf:data": {
  "ietf-yang-library:modules-state": {
    "module": [
      {
        "name": "ATM-FORUM-TC-MIB",
        "revision": "",
        "schema":
"https://10.85.116.28:443/restconf/tailf/modules/ATM-FORUM-TC-MIB",
        "namespace": "urn:ietf:params:xml:ns:yang:smiv2:ATM-FORUM-TC-MIB"
      },
      {
        "name": "ATM-MIB",
        "revision": "1998-10-19",
        "schema":
"https://10.85.116.28:443/restconf/tailf/modules/ATM-MIB/1998-10-19",
        "namespace": "urn:ietf:params:xml:ns:yang:smiv2:ATM-MIB"
      },
      {
        "name": "ATM-TC-MIB",
        "revision": "1998-10-19",
        "schema": "https://10.85.116.28:443/restconf/tailf/
..
<snip>
..
}

```

すべてのデバイス機能で前述の応答を受信すると、RESTCONF が正常に検証されます。

gRPC固有の前提条件

- キー値 Google Protocol Buffers (GPB) エンコーディングを理解する gRPC コレクタを設定します。

モデル駆動型テレメトリの制約事項

- yang-push ストリームを使用している場合、選択における自動階層は、変更時サブスクリプション向けにサポートされません。つまり、リストを選択するときに、リストの子リストが自動的に含まれません。たとえば、サブスクリバでは、子リストごとにサブスクリプションを手動で作成する必要があります。
- データアクセス許可のチェックはサポートされていません。サブスクリバによって要求されたすべてのデータが送信されます。
- サブツリーフィルタはサポートされていません。サブツリーフィルタが指定された場合、サブスクリプションは無効としてマークされます。
- サブスクリプションパラメータの中で複数の受信者を定義することはサポートされていません。最初の受信者の宛先だけが試行されます。他の定義済みの受信者は無視されます。

gRPC 固有の制限事項

- デバイスとレシーバ間の Transport Layer Security ベース (TLS ベース) の認証はサポートされていません。

TLS ベースの認証は、Cisco IOS XE Amsterdam 17.1.1 以降のリリースでサポートされています。

yang-push 固有の制限

- サブスクリプションの Quality of Service (QoS) はサポートされていません。

モデル駆動型テレメトリについて

次のセクションでは、モデル駆動型テレメトリのさまざまな側面について説明します。

モデル駆動型テレメトリの概要

テレメトリは、自動の通信プロセスです。これにより、測定およびその他のデータがリモートポイントまたはアクセス不能なポイントで収集され、モニタ用の受信装置に送信されます。モデル駆動型テレメトリは、YANG モデル化されたデータをデータ コレクタにストリーミングするためのメカニズムを提供します。

アプリケーションでは、NETCONF、RESTCONF、または gRPC ネットワーク管理インターフェイス (gNMI) の各プロトコルを介した標準ベースの YANG データ モデルを使用して、必要とする特定のデータ項目をサブスクライブできます。サブスクリプションは CLI を使用して作成することもできます (設定済みサブスクリプションの場合)。

構造化データは、サブスクリプション基準とデータタイプに基づき、定義されたパターンでまたは変更時にパブリッシュされます。

テレメトリ ロール

テレメトリを使用するシステムでは、さまざまなロールが関与します。このドキュメントでは、次のテレメトリ ロールを使用します。

- パブリッシャ：テレメトリ データを送信するネットワーク要素。
- 受信者：テレメトリデータを受信します。コレクタとも呼ばれます。
- コントローラ：サブスクリプションを作成するがテレメトリ データを受信しないネットワーク要素。作成したサブスクリプションに関連付けられたテレメトリデータが受信者に送信されます。管理エージェントまたは管理エンティティとも呼ばれます。
- サブスクライバ：サブスクリプションを作成するネットワーク要素。技術的には、受信者でもある必要はありませんが、このドキュメントではどちらも同じです。

サブスクリプションの概要

サブスクリプションは、テレメトリ ロール間の関連付けを作成する項目であり、ロール間で送信されるデータを定義します。

具体的には、サブスクリプションは、テレメトリデータの一部として要求される一連のデータを定義するために使用されます。たとえば、データがいつ必要か、データの書式設定の方法、また暗黙的でない場合は誰（どの受信者）がデータを受信するかを定義します。

サポートされているサブスクリプションの最大数はプラットフォームによって異なりますが、現在は100個のサブスクリプションがサポートされています。サブスクリプションは、設定済みか動的のいずれかにすることができ、トランスポートプロトコルの任意の組み合わせを使用できます。有効なすべての設定済みサブスクリプションをアクティブにするために同時に多数のサブスクリプションが動作している場合、サブスクリプションの数が多すぎると、アクティブなサブスクリプションを削除したときに、非アクティブであるが有効な設定済みサブスクリプションの1つが試行されます。定期的にトリガーされるサブスクリプション（デフォルトの最小値は100センチ秒）と、変更時にトリガーされるサブスクリプションがサポートされています。

サブスクリプションの設定では、NETCONFやその他のノースバウンドプログラマブルインターフェイス（RESTCONF、gNMIなど）がサポートされています。

Cisco IOS XEシステムのテレメトリでは、ダイナミックサブスクリプションと設定済みサブスクリプションの2種類のサブスクリプションが使用されます。

動的サブスクリプションは、パブリッシャに接続するクライアント（サブスクライバ）によって作成されるため、ダイヤルインと見なされます。設定済みサブスクリプションでは、パブリッシャは受信者への接続を開始し、その結果ダイヤルアウトと見なされます。

ダイヤルインおよびダイヤルアウトのモデル駆動型テレメトリ

モデル駆動型テレメトリには、ダイヤルインとダイヤルアウトの2種類があります。

表 2: ダイヤルインおよびダイヤルアウトのモデル駆動型テレメトリ

ダイヤルイン（動的）	ダイヤルアウト（静的または設定済み）
テレメトリの更新は、イニシエータまたはサブスクライバに送信されます。	テレメトリの更新は、指定された受信者またはコレクタに送信されます。
サブスクリプションの存続期間は、そのサブスクリプションを作成した接続（セッション）に結び付けられ、その存続期間中テレメトリの更新が送信されます。実行コンフィギュレーションでは変更は観察されません。	サブスクリプションは実行コンフィギュレーションの一部として作成されます。これは、コンフィギュレーションが削除されるまでデバイス設定として残ります。
ダイヤルインサブスクリプションはリロード後に再起動する必要があります。これは、確立された接続またはセッションがステートフルスイッチオーバー時にkillされるためです。	ダイヤルアウトサブスクリプションはデバイス設定の一部として作成され、ステートフルスイッチオーバー後に自動的に受信者に再接続します。
サブスクリプションIDは、サブスクリプションの確立が成功したときに動的に生成されます。	サブスクリプションIDは固定であり、設定の一部としてデバイス上で設定されます。

データソースの仕様

サブスクリプション内のテレメトリデータのソースは、ストリームとフィルタを使用して指定されます。ここでのストリームとは、関連する一連のイベントを指します。RFC 5277ではイベントストリームを、いくつかの転送基準に一致する一連のイベント通知として定義していません。

通常は、ストリームからの一連のイベントはフィルタ処理されます。異なるストリームタイプごとに異なるフィルタタイプが使用されます。

Cisco IOS XE は、`yang-push` と `yang-notif-native` の 2 つのストリームをサポートしています。

更新の通知

サブスクリプションの一部として、データが必要になるタイミングを指定できます。ただし、これはストリームによって異なります。ストリーム内で変更が行われたとき、またはイベントが発生した後にのみデータを使用できるようにするストリームもあれば、変更発生時に、または定義済みの時間間隔でデータを使用できるようにするストリームもあります。

この「タイミング」指定の結果は、対象のテレメトリデータを送る一連の更新通知となります。データの送信方法は、パブリッシャと受信者間の接続に使用されるプロトコルによって異なります。

サブスクリプション識別子

サブスクリプションは 32 ビットの正の整数値で識別されます。設定済みサブスクリプションの ID はコントローラによって設定され、動的サブスクリプションの場合はパブリッシャによって設定されます。

コントローラは、パブリッシャで作成された動的サブスクリプションとの競合を避けるために、設定済みサブスクリプションに使用する値を 0 ~ 2147483647 の範囲に制限する必要があります。動的サブスクリプションの ID 空間はグローバルです。つまり、独立して作成された動的サブスクリプションのサブスクリプション ID は重複しません。

サブスクリプション管理

管理操作の任意の形式を使用して、設定済みサブスクリプションの作成、削除、および変更を行うことができます。これには、CLI とネットワークプロトコルの両方の管理操作が含まれます。

すべてのサブスクリプション（設定済みと動的）は、`show` コマンド、およびネットワークプロトコル管理操作を使用して表示できます。

次の表で、サポートされているストリームとエンコーディング、およびサポートされている組み合わせについて説明します。入力としてのストリームは出力としてのプロトコルから独立していることを意図していますが、すべての組み合わせがサポートされているわけではありません。

表 3: サポートされるプロトコルの組み合わせ

トランスポートプロトコル	NETCONF		gRPC		gNMI	
	ダイヤルイン	ダイヤルアウト	ダイヤルイン	ダイヤルアウト	ダイヤルイン	ダイヤルアウト
Stream						
yang-push	対応	非対応	非対応	対応	対応	非対応
yang-notifnative	対応	非対応	非対応	対応	非対応	非対応
Encodings	XML	非対応	非対応	キー値 Google Protocol Buffers (kvGPB)	JSON_IETF	非対応

テレメトリの RPC サポート

確立された NETCONF セッションで、YANG XML リモートプロシージャ コール (RPC) の送受信が行えます。

テレメトリには <establish-subscription> RPC と <delete-subscription> RPC がサポートされています。

<establish-subscription> RPC が送信されると、パブリッシャからの RPC 応答には <rpc-reply> メッセージと、結果ストリングを含む <subscription-result> 要素が含まれます。

次の表は、<rpc-reply> メッセージでの応答と、応答の理由を示しています。

結果文字列	RPC	原因
ok	<establish-subscription> <delete-subscription>	成功
error-no-such-subscription	<delete-subscription>	指定されたテンプレートは存在しません。
error-no-such-option	<establish-subscription>	要求されたサブスクリプションはサポートされていません。
error-insufficient-resources	<establish-subscription>	サブスクリプションは次の理由により作成できません。 <ul style="list-style-type: none"> サブスクリプションが多すぎる。

結果文字列	RPC	原因
		<ul style="list-style-type: none"> 要求されたデータの量が大きすぎる。 定期的なサブスクリプションの間隔が短すぎる。
error-other	<establish-subscription>	その他の何らかのエラーです。

サービス gNMI

gNMI 仕様は、ハイレベル RPC を含む gNMI という名前の単一のトップレベルサービスを識別します。次に、サブスクライブサービス RPC を含むサービス定義を示します。

```
service gNMI{
  .
  .
  .
  rpc Subscribe(stream SubscribeRequest)
    returns (stream SubscribeResponse);
}
```

<subscribe RPC> は、動的サブスクリプションを要求するために管理エージェントによって使用されます。この RPC には一連のメッセージが含まれています。次のセクションでは、<subscribe RPC> でサポートされているメッセージについて説明します。

SubscribeRequest メッセージ

このメッセージは、指定されたパスのセットに対するターゲットからの更新を要求するためにクライアントによって送信されます。次に、メッセージの定義を示します。

```
message SubscribeRequest {
  oneof request {
    SubscriptionList subscribe = 1;
    PollRequest poll = 3;
    AliasList aliases = 4;
  }
  Repeated gNMI_ext.Extensions = 5;
}
```



(注) request.subscribe のみがサポートされます。

SubscribeResponse メッセージ

このメッセージは、確立された <subscribe RPC> を介してターゲットからクライアントに送信されます。次に、メッセージの定義を示します。

```

message SubscribeResponse {
  oneof response {
    Notification update = 1;
    Bool sync_response = 3;
    Error error = 4 [deprecated=true];
  }
}

```



(注) 通知の更新のみがサポートされます。

SubscriptionList メッセージ

このメッセージは、共通のサブスクリプション動作が必要なパスのセットを示すために使用されます。SubscriptionList メッセージの仕様内で、クライアントはモデル内の特定のプレフィックスに対する1つ以上のサブスクリプションを識別できます。次に、SubscriptionList メッセージの定義を示します。

```

message SubscriptionList {
  Path prefix = 1;
  repeated Subscription subscription = 2;
  bool use_aliases = 3;
  QOSMarking qos = 4;
  enum Mode {
    STREAM = 0;
    ONCE = 1;
    POLL = 2;
  }
  Mode mode = 5;
  bool allow_aggregation = 6;
  repeated ModelData use_models = 7;
  Encoding encoding = 8; // only JSON_IETF supported in R16.12
  Bool updates_only = 9;
}

```



(注) Path prefix (明示的な要素名のみ)、Subscription subscription、Mode mode STREAM、および Encoding encoding IETF_JSON がサポートされています。

プレフィックスメッセージ

有効なサブスクリプションリストには、XPath の (要求されたすべてのサブスクリプション間で) 共有部分で構成された入力済みのプレフィックスが含まれている場合と、含まれていない場合があります。

```

message Path {
  repeated string element = 1; [ deprecated ]
  string origin = 2;
  repeated PathElem elem = 3;
}

```

```

    optional string target = 4;
}

```



- (注) origin (サポートされる値は「」と「openconfig」)、elem (サポートされる要素名はプレフィックスなし)、および target がサポートされます。

サブスクリプションメッセージ

このメッセージは、クライアントによってサブスクライブされるデータのセットを一般的に説明しています。通知動作を制御するために使用されるパスと属性が含まれます。次に、サブスクリプションメッセージの定義を示します。

```

message Subscription {
    Path path = 1;
    SubscriptionMode mode = 2;
    uint64 sample_interval = 3;
    bool suppress_redundant = 4;
    uint64 heartbeat_interval = 5;
}

```



- (注) Path path、SubscriptionMode mode、Uint64 sample_interval、および Uint64 heartbeat_interval (値が 0 に設定されている場合のみ) がサポートされます。

パスメッセージ

有効なサブスクリプションには、パスが入力されています。これは、サブスクリプションリストに関連付けられたプレフィックスに追加されると、完全修飾パスを構成します。次に、パスメッセージの定義を示します。

```

message Path {
    repeated string element = 1; [ deprecated ]
    string origin = 2;
    repeated PathElem elem = 3;
    optional string target = 4;
}

```



- (注) origin (サポートされる値は「」と「openconfig」)、elem (サポートされる要素名はプレフィックスなし)、および target がサポートされます。

SubscriptionMode メッセージ

このメッセージは、通知の更新をトリガーする方法をターゲットに通知します。次に、SubscriptionMode メッセージの定義を示します。

```
enum SubscriptionMode {
    TARGET_DEFINED = 0;
    ON_CHANGE     = 1;
    SAMPLE        = 2;
}
```



- (注) SAMPLE と ON_CHANGE (Cisco IOS XE Bengaluru 17.6.1 以降) のみがサポートされています。ON_CHANGE のサポートは、特定のモデルパスに限定されています。パスが ON_CHANGE をサポートしているかどうかを確認するには、Cisco-IOS-XE-MDT-capabilities-oper モデルでパスをクエリします。モデルの詳細については、「[変更時サブスクリプション YANG モデルの表示 \(30 ページ\)](#)」セクションを参照してください。

通知メッセージ

このメッセージは、サブスクリプションターゲットからコレクタにテレメトリデータを配信します。次に、通知メッセージの定義を示します。

```
message Notification {
    int64 timestamp = 1;
    Path prefix = 2;
    string alias = 3;
    repeated Update update = 4;
    repeated Path delete = 5;
    bool atomic = 6;
}
```



- (注) タイムスタンプ、プレフィックス、および更新がサポートされます。

ダイナミックサブスクリプション管理

ここでは、動的サブスクリプションを作成および削除する方法について説明します。

NETCONF ダイアラインの動的サブスクリプションの作成

動的サブスクリプションは、パブリッシャに接続し、その接続内部のメカニズム (通常はRPC) を使用してサブスクリプション作成のための呼び出しを行うサブスクリバによって作成されます。サブスクリプションの存続期間は、サブスクリバとパブリッシャ間の接続の存続期間に制限され、テレメトリデータはそのサブスクリバにのみ送信されます。これらのサブスクリプションは、パブリッシャまたはサブスクリバのいずれかが再起動された場合は存続しません。動的サブスクリプションの作成にはインバンドの <establish-subscription> RPC を使用できます。<establish-subscription> RPC は、IETF テレメトリのサブスクリバからネットワーク

デバイスに送信されます。RPC では、`stream`、`xpath-filter`、および `period` の各フィールドが必須です。

NETCONF による動的サブスクリプションの作成および削除に使用する RPC は、[イベント通知のカスタムサブスクリプション *draft-ietf-netconf-subscribed-notifications-03*](#) および [YANG データストア プッシュ更新のサブスクリプション *draft-ietf-netconf-yang-push-07*](#) で定義されています。

定期的な動的サブスクリプション

次に、ダイヤルインの定期的なサブスクリプションの例を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <establish-subscription
    xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
    xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <stream>yp:yang-push</stream>
    <yp:xpath-filter>/mdt-oper:mdt-oper-data/mdt-subscriptions</yp:xpath-filter>
    <yp:period>1000</yp:period>
  </establish-subscription>
</rpc>
```

変更時動的サブスクリプション

次に、NETCONF を介した変更時動的サブスクリプションの例を示します。

```
<establish-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
  xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <stream>yp:yang-push</stream>

  <yp:xpath-filter>/cdp-ios-xe-oper:cdp-neighbor-details/cdp-neighbor-detail</yp:xpath-filter>

  <yp:dampening-period>0</yp:dampening-period>
</establish-subscription>
```

動的サブスクリプションの削除

動的サブスクリプションを削除するには、インバンドの `<delete subscription>` RPC、**clear telemetry ietf subscription** コマンド、`<kill-subscription>` RPC を使用し、トランスポートセッションを切断します。

gNMI では、`SubscribeRequest.subscribe.subscription` の各サブスクリプションが個別のダイナミックサブスクリプション ID として生成されます。`<kill-subscription>` RPC または CLI のクリアを使用してこれらのサブスクリプション ID のいずれかを強制終了すると、サブスクリプション要求で指定されたすべてのサブスクリプションが強制終了されます。

Cisco IOS XE Gibraltar 16.10.1 で導入された `<delete-subscription>` RPC は、サブスクリプションのみが発行でき、そのサブスクリプションが所有するサブスクリプションのみを削除します。

Cisco IOS XE Gibraltar 16.11.1 以降のリリースでは、**clear telemetry ietf subscription** コマンドを使用してダイナミックサブスクリプションを削除できます。Cisco IOS XE Gibraltar 16.11.1 で導

入された <kill-subscription> RPC は、**clear telemetry ietf subscription** コマンドと同じ方法でダイナミック サブスクリプションを削除します。

親の NETCONF セッションが切断されると、サブスクリプションも削除されます。ネットワーク接続が中断された場合は、SSH または NETCONF セッションがタイムアウトしてその後にサブスクリプションが削除されるまで、多少の時間がかかることがあります。

<kill-subscription> RPC は <delete-subscription> RPC と類似しています。ただし、<kill-subscription> RPC は、*subscription-id* 要素の代わりに、削除するサブスクリプションの ID を含む *identifier* 要素を使用します。ターゲットサブスクリプションで使用されるトランスポートセッションも、<delete-subscription> RPC で使用されているものと異なります。

CLI を使用したサブスクリプションの削除

次の出力例は、使用可能なすべてのサブスクリプションを示しています。

```
Device# show telemetry ietf subscription all

Telemetry subscription brief

ID              Type          State          Filter type
-----
2147483648      Dynamic       Valid          xpath
2147483649      Dynamic       Valid          xpath
```

次に、ダイナミック サブスクリプションを削除する例を示します。

```
Device# clear telemetry ietf subscription 2147483648
```

NETCONF <delete-Subscription> RPC を使用したサブスクリプションの削除

次に、NETCONF を使用してサブスクリプションを削除する例を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <delete-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
    xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
    <subscription-id>2147483650</subscription-id>
  </delete-subscription>
</rpc>
```

NETCONF <kill-Subscription> RPC を使用したサブスクリプションの削除

次に、<kill-subscription> RPC を使用してサブスクリプションを削除する例を示します。

```
<get>
<filter>
<mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
<mdt-subscriptions/>
</mdt-oper-data>
</filter>
</get>
```

* Enter a NETCONF operation, end with an empty line

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-subscriptions>
        <subscription-id>2147483652</subscription-id>
        <base>
...
          </base>
          <type>sub-type-dynamic</type>
          <state>sub-state-valid</state>
          <comments/>
          <mdt-receivers>
...
        </mdt-receivers>
        <last-state-change-time>2018-12-13T21:16:48.848241+00:00</last-state-change-time>
      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483653</subscription-id>
        <base>
...
          </base>
          <type>sub-type-dynamic</type>
          <state>sub-state-valid</state>
          <comments/>
          <mdt-receivers>
...
        </mdt-receivers>
        <last-state-change-time>2018-12-13T21:16:51.319279+00:00</last-state-change-time>
      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483654</subscription-id>
        <base>
...
          </base>
          <type>sub-type-dynamic</type>
          <state>sub-state-valid</state>
          <comments/>
          <mdt-receivers>
...
        </mdt-receivers>
        <last-state-change-time>2018-12-13T21:16:55.302809+00:00</last-state-change-time>
      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483655</subscription-id>
        <base>
...
          </base>
          <type>sub-type-dynamic</type>
          <state>sub-state-valid</state>
          <comments/>
          <mdt-receivers>
...
        </mdt-receivers>
        <last-state-change-time>2018-12-13T21:16:57.440936+00:00</last-state-change-time>
      </mdt-subscriptions>
    </mdt-oper-data>
  </data>
</rpc-reply>
<kill-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
  xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">

```

```

    <identifier>2147483653</identifier>
  </kill-subscription>

  * Enter a NETCONF operation, end with an empty line

  <?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
    <subscription-result xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
  xmlns:notif-bis="urn:ietf:params:xml:ns:yang:ietf-event-notifications">notif-bis:ok</subscription-result>
  </rpc-reply>
  <get>
  <filter>
  <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
  <mdt-subscriptions/>
  </mdt-oper-data>
  </filter>
  </get>

  * Enter a NETCONF operation, end with an empty line

  <?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
    <data>
      <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
        <mdt-subscriptions>
          <subscription-id>2147483652</subscription-id>
          <base>
            ...
            </base>
            <type>sub-type-dynamic</type>
            <state>sub-state-valid</state>
            <comments/>
            <mdt-receivers>
              ...
            </mdt-receivers>
            <last-state-change-time>2018-12-13T21:16:48.848241+00:00</last-state-change-time>
          </mdt-subscriptions>
          <mdt-subscriptions>
            <subscription-id>2147483654</subscription-id>
            <base>
              ...
              </base>
              <type>sub-type-dynamic</type>
              <state>sub-state-valid</state>
              <comments/>
              <mdt-receivers>
                ...
              </mdt-receivers>
              <last-state-change-time>2018-12-13T21:16:55.302809+00:00</last-state-change-time>
            </mdt-subscriptions>
            <mdt-subscriptions>
              <subscription-id>2147483655</subscription-id>
              <base>
                ...
                </base>
                <type>sub-type-dynamic</type>
                <state>sub-state-valid</state>
                <comments/>
                <mdt-receivers>
                  ...
                </mdt-receivers>
              </mdt-subscriptions>
            </mdt-oper-data>
          </data>
        </rpc-reply>
      </data>
    </rpc-reply>
  </data>

```



```

<last-state-change-time>2018-12-13T21:16:57.440936+00:00</last-state-change-time>

</mdt-subscriptions>
</mdt-oper-data>
</data>
</rpc-reply>

```

設定済みサブスクリプションの管理

ここでは、設定済みサブスクリプションを作成、変更、および削除する方法について説明します。

設定済みサブスクリプションの作成

設定済みサブスクリプションは、コントローラによるパブリッシャでの管理操作によって作成され、サブスクリプションによって定義されたテレメトリデータの受信者の指定が明示的に含まれています。これらのサブスクリプションは、パブリッシャの再起動後も持続します。

設定済みサブスクリプションは複数の受信者を使用して設定できますが、最初の有効な受信者のみが使用されます。受信者がすでに接続されている場合、または接続中の場合は、他の受信者への接続は試行されません。その受信者が削除されると、別の受信者が接続されます。

設定済みダイヤルアウトサブスクリプションは、次の方法でデバイスに設定されます。

- 設定 CLI を使用し、コンソール/VTY を介してデバイス設定に変更を加えます。
- NETCONF/RESTCONF を使用し、目的のサブスクリプションを設定します。

ここでは、設定済みサブスクリプションを作成するための RPC の例を示します。

定期的なサブスクリプション

次の例は、CLI を使用して、設定済みサブスクリプションのトランスポートプロトコルとして gRPC を設定する方法を示しています。

```

telemetry ietf subscription 101
  encoding encode-kvgpb
  filter xpath /memory-ios-xe-oper:memory-statistics/memory-statistic
  stream yang-push
  update-policy periodic 6000
  source-vrf Mgmt-intf
  receiver ip address 10.28.35.45 57555 protocol grpc-tcp

```

次の RPC の例は、NETCONF を使用して定期的なサブスクリプションを作成し、60 秒ごとにテレメトリの更新を受信者に送信する方法を示します。

```

<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><edit-config>
  <target>
    <running/>
  </target>
  <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
      <mdt-subscription>
        <subscription-id>200</subscription-id>
        <base>
          <stream>yang-push</stream>
        </base>
      </mdt-subscription>
    </mdt-config-data>
  </config>
</edit-config>
</rpc>

```

```

    <encoding>encode-kvgpb</encoding>
    <period>6000</period>
    <xpath>/memory-ios-xe-oper:memory-statistics/memory-statistic</xpath>
  </base>
<mdt-receivers>
  <address>10.22.23.48</address>
  <port>57555</port>
  <protocol>grpc-tcp</protocol>
</mdt-receivers>
</mdt-subscription>
</mdt-config-data>
</config>
</edit-config>
</rpc>

```

次に、RESTCONF を使用して定期的なサブスクリプションを作成する RPC の例を示します。

URI:https://10.85.116.28:443/restconf/data/Cisco-IOS-XE-mdt-cfg:mdt-config-data

Headers:

application/yang-data.collection+json, application/yang-data+json,

application/yang-data.errors+json

Content-Type:

application/yang-data+json

BODY:

```

{
  "mdt-config-data": {
    "mdt-subscription": [
      {
        "subscription-id": "102",
        "base": {
          "stream": "yang-push",
          "encoding": "encode-kvgpb",
          "period": "6000",
          "xpath": "/memory-ios-xe-oper:memory-statistics/memory-statistic"
        }
        "mdt-receivers": {
          "address": "10.22.23.48"
          "port": "57555"
        }
      }
    ]
  }
}

```

変更時サブスクリプション

次の RPC の例は、NETCONF を使用して変更時サブスクリプションを作成し、ターゲットデータベースに変更が生じた場合にのみ更新を送信する方法を示します。

```

<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><edit-config>
  <target>
    <running/>
  </target>
  <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
      <mdt-subscription>
        <subscription-id>200</subscription-id>
        <base>
          <stream>yang-push</stream>
          <encoding>encode-kvgpb</encoding>
          <no-synch-on-start>false</no-synch-on-start>
          <xpath>/cdp-ios-xe-oper:cdp-neighbor-details/cdp-neighbor-detail</xpath>
        </base>
      <mdt-receivers>

```

```

        <address>10.22.23.48</address>
        <port>57555</port>
        <protocol>grpc-tcp</protocol>
      </mdt-receivers>
    </mdt-subscription>
  </mdt-config-data>
</config>
</edit-config>
</rpc>

```

次の RPC の例は、RESTCONF を使用して変更時サブスクリプションを作成する方法を示します。

```

URI:
https://10.85.116.28:443/restconf/data/Cisco-IOS-XE-mdt-cfg:mdt-config-data
Headers:
application/yang-data.collection+json, application/yang-data+json,
application/yang-data.errors+json
Content-Type:
application/yang-data+json
BODY:
{
  "mdt-config-data": {
    "mdt-subscription": [
      {
        "subscription-id": "102",
        "base": {
          "stream": "yang-push",
          "encoding": "encode-kvgpb",
          "dampening period": "0",
          "xpath": "/cdp-ios-xe-oper:cdp-neighbor-details/cdp-neighbor-detail "
        }
      }
    ],
    "mdt-receivers": {
      "address": "10.22.23.48"
      "port": "57555"
    }
  }
}

```

gNMI ダイヤルインサブスクリプション

次に、gNMI ダイヤルインサブスクリプションの例を示します。

```

subscribe: <
  prefix: <>
  subscription: <
    path: <
      origin: "openconfig"
      elem: <name: "routing-policy">
    >
    mode: SAMPLE
    sample_interval: 1000000000
  >
  mode: STREAM
  encoding: JSON_IETF
>'

subscribe: <
  prefix: <>
  subscription: <

```

```

path: <
  origin: "legacy"
  elem: <name: "oc-platform:components">
  elem: <
    name: "component"
    key: <
      key: "name"
      value: "PowerSupply8/A"
    >
  >
  elem: <name: "power-supply">
  elem: <name: "state">
>
mode: SAMPLE
sample_interval: 10000000000
>
mode: STREAM
encoding: JSON_IETF
>'

```

設定済みサブスクリプションの変更

設定済みサブスクリプションを変更するには、次の2つの方法があります。

- NETCONF <edit-config> RPC などの管理プロトコル設定操作
- CLI (サブスクリプションの作成と同じ手順)

サブスクリプションの受信者はアドレスとポート番号によって識別されます。受信者を変更することはできません。受信者の特性（プロトコル、プロファイルなど）を変更するには、先に受信者を削除してから新しい受信者を作成する必要があります。

有効なサブスクリプションの有効な受信者設定が切断状態にあり、管理側で受信者への接続のセットアップ時に新しい試行を強制する場合は、同一の特性を持つ受信者を書き換える必要があります。

設定済みサブスクリプションの削除

CLI または管理操作を使用して、設定済みサブスクリプションを削除できます。**no telemetry ietf subscription** コマンドは、設定済みサブスクリプションを削除します。設定されたサブスクリプションは、設定インターフェイスからのみ削除できます。

CLI を使用したサブスクリプションの削除

```

Device# configure terminal
Device(config)# no telemetry ietf subscription 101
Device(config)# end

```

NETCONF を使用したサブスクリプションの削除

次の RPC の例は、設定済みサブスクリプションを削除する方法を示しています。

```

<edit-config>
  <target>
    <running/>

```

```
</target>
<config>
  <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
    <mdt-subscription operation="delete">
      <subscription-id>102</subscription-id>
    </mdt-subscription>
  </mdt-config-data>
</config>
</edit-config>
```

gRPC サブスクリプションの FQDN サポート

gRPC テレメトリ サブスクリプションは設定ベースです。つまり、ユーザーはデバイス設定の一部として受信ホストとその他のサブスクリプションパラメータを指定する必要があります。このレシーバ設定は、テレメトリの更新を送信するための接続の詳細を決定するために使用されます。gRPC サブスクリプション機能の FQDN サポートの導入により、IP アドレスに加え、完全修飾ドメイン名 (FQDN) も gRPC サブスクリプションに使用できます。

テレメトリ サブスクリプションでは、レシーバの詳細をサブスクリプションの一部として指定することも、独立して設定することも可能になりました。ここで、レシーバには名前があり、サブスクリプションを設定するときにこの名前を使用してレシーバを指定します。どちらの場合も、複数のサブスクリプションに同じレシーバ名を指定できます。

この機能をディセーブルにできません。

名前付きレシーバ

FQDN のサポートにより、名前付きレシーバ設定と呼ばれる新しいレシーバ設定方法が導入されました。名前付きレシーバは、サブスクリプションとは無関係に存在できるトップレベルの設定エンティティです。名前付きレシーバは名前で識別されます。この名前は任意の文字列で、システムで名前付きレシーバレコードのインデックスまたはキーとなります。名前付きレシーバ設定には、レシーバに関連付けられているすべての設定が含まれます。これらの設定はサブスクリプションに依存しません。

名前付きレシーバを使用する利点は次のとおりです。

- さまざまなタイプのレシーバをサポートできます。
- より正確な状態および診断の情報を得られます。
- IP アドレスの代わりにホスト名を使用して、プロトコルレシーバのホストを指定できます。
- 複数のサブスクリプションで使用されるレシーバのパラメータを、1つの場所で変更できます。

次のプロトコルタイプ名前付きレシーバだけがサポートされます。

- `cloud-native` : クラウドネイティブプロトコル
- `cntp-tcp` : Civil Network Time Protocol (CNTP) TCP プロトコル
- `cntp-tls` : CNTP TLS プロトコル

- grpc-tcp : gRPC TCP プロトコル
- grpc-tls : gRPC TLS プロトコル
- native : ネイティブプロトコル
- tls-native : ネイティブ TLS プロトコル

名前付きプロトコルレシーバ

名前付きプロトコルレシーバは、プロトコルを使用するテレメトリの転送を指定するために使用されます。名前付きプロトコルレシーバは、レシーバを識別する名前に加えて、ホスト指定も使用します。ホスト指定には、ホスト名または IP アドレス、および宛先ポート番号を指定します。セキュアプロトコル転送もプロファイル文字列を使用します。



- (注) 有効な名前付きプロトコルレシーバが作成されると、レシーバに自動的に接続されません。レシーバへの接続を作成するには、指定されたプロトコルレシーバが、少なくとも1つのサブスクリプションによって要求される必要があります。

CLI または YANG モデルを使用して、名前付きプロトコルレシーバを設定できます。

名前付きプロトコルレシーバの YANG モデルを使用した設定

YANG モデル Cisco-IOS-XE-mdt-cfg には、名前付きプロトコルレシーバが含まれています。最上位の mdt-config-data コンテナ内のコンテナ mdt-named-protocol-rcvrs には、mdt-named-protocol-rcvr 構造体のリストがあります。このグループには、次の 5 つのメンバーがあります。

- Name : リスト内のインデックス
- プロトコル
- Profile
- ホストネーム
- ポート番号

次に、名前付きプロトコルレシーバの作成方法を示す、NETCONF RPC の例を示します。

```
<edit-config>
  <target>
    <running/>
  </target>
  <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
      <mdt-named-protocol-rcvrs>
        <mdt-named-protocol-rcvr>
          <name>receiver1</name>
          <protocol>tls-native</protocol>
          <profile>tls-trustpoint</profile>
          <host>
            <hostname>rcvr.test.com</hostname>
          </host>
        </mdt-named-protocol-rcvr>
      </mdt-named-protocol-rcvrs>
    </mdt-config-data>
  </config>
</edit-config>
```

```

    </host>
    <port>45000</port>
  </mdt-named-protocol-rcvr>
</mdt-named-protocol-rcvrs>
</mdt-config-data>
</config>
</edit-config>

```

名前付きレシーバを使用したサブスクリプションの設定

サブスクリプションで名前付きレシーバを使用するには、レシーバタイプとレシーバ名の両方を指定する必要があります。すべてのレシーバ固有の設定は名前付きレシーバ設定の一部であるため、追加のレシーバ設定は必要ありません。ただし、名前付きプロトコルレシーバは、接続解決プロセスの一部として、サブスクリプションの送信元 VRF および送信元アドレスを引き続き使用します。

サポートされる名前レシーバタイプは **protocol** のみです。

サブスクリプションは名前付きレシーバまたはレガシーレシーバのいずれかを使用できますが、両方を使用することはできません。レガシーレシーバが設定されている場合、サブスクリプションレシーバタイプと名前付きレシーバ名の設定はブロックされます。同様に、サブスクリプションレシーバタイプまたは名前付きレシーバを指定した場合は、レガシーレシーバを設定できません。

複数のレシーバが設定されている場合でも、サブスクリプションは1つのレシーバのみを使用することに注意してください。

レガシーレシーバを使用するサブスクリプションと名前付きレシーバを使用するサブスクリプションは、同じ接続を使用できます。ただし、これは推奨されません。

名前付きレシーバサブスクリプション設定の YANG モデルを使用した設定

名前付きレシーバを使用する場合、**rcvr-type** でサポートされる唯一の値は **rcvr-type-protocol** です。レガシーレシーバを使用する場合、この値はデフォルトの **rcvr-type-undefined** です。

次に、NETCONF RPC の例を示します。名前付きプロトコルレシーバを使用してサブスクリプションを作成する方法を示しています。

```

<edit-config>
  <target>
    <running/>
  </target>
  <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
      <mdt-subscription>
        <subscription-id>1</subscription-id>
        <base>
          <rcvr-type>rcvr-type-protocol</rcvr-type>
        </base>
        <mdt-receiver-names>
          <mdt-receiver-name>
            <name>receiver1</name>
          </mdt-receiver-name>
        </mdt-receiver-names>
      </mdt-subscription>
    </mdt-config-data>
  </config>
</edit-config>

```

```
</config>
</edit-config>
```

名前付きレシーバの動作と動作状態

名前付きレシーバオブジェクトとサブスクリプションレシーバオブジェクト（名前付きレシーバを参照する）には、2つの異なる動作状態があります。その動作状態は有効または無効です。名前付きレシーバが無効になる最も一般的な理由は、設定が不完全であることですが、他の理由も考えられます。名前付きレシーバの動作状態ビューには、レシーバが無効である理由をテキストで説明するフィールドがあります。レシーバの状態が有効な場合、このフィールドは空です。

CLI を使用した名前付きレシーバの状態の表示

すべてのタイプの名前付きレシーバの状態を表示するには、**show telemetry receiver** コマンドを使用します。**all** キーワードは、すべての名前付きレシーバに関する情報を簡単な形式で表示し、**name** キーワードは、指定された名前付きレシーバに関する詳細情報を表示します。

次に、**show telemetry receiver all** コマンドの出力例を示します。

```
Device# show telemetry receiver all
```

```
Telemetry receivers
```

Name	Type	Profile	State	Explanation
receiver1	protocol	tls-trustpoint	Valid	

次に、**show telemetry receiver name** コマンドの出力例を示します。

```
Device# show telemetry receiver name receiver1
```

```
Name: receiver1
Profile: tls-trustpoint
State: Valid
Last State Change: 08/12/20 19:55:54
Explanation:
Type: protocol
Protocol: tls-native
Host: rcvr.test.com
Port: 45000
```

名前付きレシーバの YANG モデルを使用した状態

名前付きレシーバの状態は、Cisco-IOS-XE-mdt-oper-v2 YANG モデルを使用して取得できます。**mdt-oper-v2-data** コンテナには、すべての名前付きレシーバの動作状態を含む **mdt-named-receivers** リストが含まれています。

次に、名前付きレシーバの状態を取得する NETCONF 応答の例を示します。

```
<get>
<filter>
```



```

    <mdt-oper-v2-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper-v2">
      <mdt-named-receivers/>
    </mdt-oper-v2-data>
  </filter>
</get>

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <data>
    <mdt-oper-v2-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper-v2">
      <mdt-named-receivers>
        <name>receiver1</name>
        <profile>tls-trustpoint</profile>
        <params>
          <protocol>tls-native</protocol>
        </params>
      <host>
        <hostname>rcvr.test.com </hostname>
      </host>
      <port>45000</port>
    </params>
    <state>named-rcvr-state-valid</state>
    <last-state-change-time>2020-...:00</last-state-change-time>
  </mdt-named-receivers>
</mdt-oper-v2-data>
</data>
</rpc-reply>

```

サブスクリプションレシーバの動作と動作状態

サブスクリプションレシーバは、実際のサブスクリプションレシーバまたはコレクタに接続するサブスクリプション関連のオブジェクトです。コレクタに到達するために必要なメカニズムはレシーバタイプに固有ですが、接続は、サブスクリプションがレシーバまたはコレクタに到達できるようにするために使用されるエンティティです。

サブスクリプションレシーバの状態は、レシーバへの接続を要求および使用する機能に基づいており、複数の状態があります。それぞれの状態は、サブスクリプションがレシーバまたはコレクタに更新を送信できるようにするために必要な他のリソースの制御に関連付けられています。

サブスクリプションレシーバの状態

サブスクリプションレシーバの動作状態は、設定された名前（接続のインデックス）、レシーバの状態、状態に関する説明またはメモ、および最後の状態変更の時刻で構成されます。説明文字列は常に使用されるわけではありません。

サブスクリプションレシーバの考えられる状態を次の表に示します。

表 4: サブスクリプションレシーバの状態

サブスクリプションレシーバの状態		説明
CLI 値	YANG 値	レシーバは切断され、再接続は試行されません。
Disconnected（切断）	rcvr-state-disconnected	

サブスクリプションレシーバの状態		説明
Resolving	rcvr-state-resolving	レシーバに到達するために必要な接続パラメータを解決しています。
Transport requested	rcvr-state-transport-requested	レシーバに到達するための接続要求が、 Resolving 状態から決定された接続パラメータを使用していました。
接続中 (Connecting)	rcvr-state-connecting	サブスクリプションをレシーバに接続するために必要なリソースが割り当てられています。
接続されている状態	rcvr-state-connecting	サブスクリプションがレシーバに接続され、更新をレシーバに送信できるようになりました。
Disconnecting	rcvr-state-disconnected	接続で使用されているリソースが再割り当てされています。

YANG 値 `rcvr-state-invalid` は、レガシーレシーバでのみ使用されます。無効なサブスクリプションレシーバは接続できないため、サブスクリプションレシーバの状態が無効な場合は `disconnected` に設定されます。説明文字列で、`invalid` のサブスクリプションレシーバと `disconnected` のサブスクリプションレシーバを区別できます。

サブスクリプションレシーバは、次の理由で切断されることがあります。

- サブスクリプションの別のレシーバが切断されていない。
- 接続のセットアップが永続的に失敗した。
- 名前付きレシーバが存在しない。
- 名前付きレシーバがサブスクリプションで指定されたタイプではない。
- 名前付きレシーバが無効である。
- サブスクリプションが無効である。
- 要求された接続が別のレシーバによって使用されている。

サブスクリプションレシーバの接続

このセクションでは、サブスクリプションレシーバが接続を使用する方法について説明します。

テレメトリ接続

テレメトリ接続は、サブスクリプションがレシーバに到達するために使用するトランスポートインスタンスを表し、単に動作状態を表します。テレメトリ接続は、整数のインデックス値で識別されます。接続に関するその他の情報は、サブスクリプションが使用するように設定されているレシーバのタイプに基づく接続のタイプに固有です。

セキュアなシスコ独自のトランスポートでは、設定された名前付きレシーバのホスト部分は、接続のセットアップ時にレシーバが提供する証明書の識別名 (DN) と一致する必要があります。このため、同じ接続を使用する複数のレシーバを持つことはできません。

この項で説明するすべての状態は、すべてのタイプの接続で該当しますが、すべてが使用されるわけではありません。

表 5: テレメトリの接続状態

接続状態		説明
CLI 値	YANG 値	
Pending	con-state-pending	接続が作成されましたが、まだ開始されていません。
Connecting	con-state-connecting	接続をセットアップする要求が進行中です。
Active	con-state-active	接続が確立されており、サブスクリプションレシーバで使用できます。
Disconnecting	con-state -disconnecting	接続が切断され、サブスクリプションレシーバによる解放を待機しています。

接続に関連付けられているその他の動作状態には、リモートレシーバ（使用可能な場合はピア）の ID、および最後の状態変更の時刻などがあります。

テレメトリプロトコル接続

ここでは、プロトコルタイプ接続と、これらが名前付きプロトコルレシーバに割り当てられたサブスクリプションレシーバによりどのように使用されるかについて説明します。

表 6: プロトコルタイプ接続のパラメータ

パラメータ	発信元	注
Destination IP address	名前付きレシーバホスト	ホストはドメイン名を使用するため、ドメイン名の解決が必要になる場合があります。

パラメータ	発信元	注
Destination port number	名前付きレシーバポート	明示的に設定する必要があります。
Source VRF	サブスクリプション（指定されている場合）	VRF が指定されていない場合、デフォルトの VRF が使用されます。それ以外の場合、VRF 名は内部識別子に解決されます。
Source IP address	サブスクリプション（指定されている場合）	指定しない場合、送信元 IP アドレスは VRF および宛先 IP アドレスに基づいて決定されます。

これらのパラメータの一部は、サブスクリプションレシーバの親サブスクリプションの設定に基づいています。

設定から接続パラメータを解決するときに、順序が指定されていない場合は、最初に VRF が決定され、次に宛先 IP アドレス、最後に送信元 IP アドレスが決定されます。解決の特定のステップが非永続的に失敗する場合、5 秒間隔で無限に再試行されます。

接続は、要求されるとすぐにインスタンス化されます。つまり、最初のサブスクリプションレシーバが **resolving** 状態から **transport requested** 状態に移行するとすぐに、サブスクリプションレシーバによって解決されたパラメータを持つ接続インスタンスが作成されます。

要求された接続が正常に設定され、テレメトリで使用されると、接続状態は **connected** に変わります。これは、Cisco IOS XE デバイスとレシーバデバイス間に接続が存在することを意味します。レシーバが使用するリソースを再割り当てするために、リソースを使用するサブスクリプションレシーバに、接続が設定されたことが通知されます。これらのサブスクリプションレシーバの状態は、**connecting** に移行して、サブスクリプションをレシーバに接続するために必要なリソースを設定します。これらのリソースが確保されると、サブスクリプションレシーバの状態が **connected** に変更され、レシーバが更新通知を受信します。

テレメトリ接続がアクティブになることができない理由の一部を次に示します。

- 接続先に到達できない。
- リモートホストポートにリスナーが存在しない。
- リモートホストポートのリスナーのタイプが正しくない。
- Authentication failures（認証エラー）：



(注) 接続セットアップが進行中のときは、接続セットアップを開始するために必要なパラメータが正常に解決されているため、この接続を使用するサブスクリプションレシーバはすべて **connecting** 状態になります。

接続セットアップが失敗したときに実行されるアクションは、プロトコルによって異なります。次の表に、単一のセットアップ要求内の接続の再試行動作と、接続セットアップ要求が失敗した場合の再解決要求の再試行動作を示します。この動作は、レガシーレシーバによって要求された接続でも同じです。

表 7: プロトコル固有の再試行間隔

Protocol	接続の再試行回数	再解決要求
<ul style="list-style-type: none"> • grpc-tcp • grpc-tls 	1、3、4、7 秒の間隔で 5 回再試行	制限なし。接続の再試行が失敗すると、継続的に再解決を要求します（試行ごとに 14 秒）。
<ul style="list-style-type: none"> • cloud-native • cntp-tcp • cntp-tls • native • tls-native 		5、10、15、20、25、および 30 秒。

名前付きレシーバ接続のトラブルシューティング

サブスクリプションが設定されている場合、一般的な問題の 1 つは、テレメトリ更新メッセージが受信されないことです。原因として、送信するイベントがないか、サブスクリプションが無効であることが考えられます。ここでは、名前付きレシーバ接続で発生する一般的な問題のトラブルシューティング方法について説明します。

テレメトリプロセスからのログ、および一部の **show** コマンドの出力には、名前付きレシーバ設定のトラブルシューティングに使用できる情報が含まれています。

表 8: 名前付きレシーバ接続のトラブルシューティング

問題	確認方法/症状	対処法
サブスクリプション構成が無効。	show telemetry ietf subscription iddetails	サブスクリプションの設定を修正します。
サブスクリプションレシーバが無効。	show telemetry ietf subscription idreceiver	名前付きレシーバの設定を修正します。
サブスクリプションレシーバの接続パラメータを解決できない。	show telemetry ietf subscription idreceiver サブスクリプションレシーバの状態が、解決中の状態から変わらない。	レシーバ、ネットワーク設定、またはインターフェイスの状態を確認します。

問題	確認方法/症状	対処法
サブスクリプションレシーバ接続がアップ状態にならない。	show telemetry ietf subscription idreceiver サブスクリプションレシーバの状態が、解決中から接続中に何度も変化する。	解決された接続が有効であり、レシーバまたはコレクタが到達可能で、指定されたトランスポートを使用してインバウンド接続を受け入れることができることを確認します。
サブスクリプションレシーバの接続が拒否される。	show telemetry ietf subscription idreceiver サブスクリプションレシーバの状態が、切断を除くすべての状態に変化し続ける。	コレクタのタイプが正しいこと、および設定されている認証と許可が有効であることを確認します。
サブスクリプションレシーバが接続されているが、更新が受信されない。	show telemetry internal subscription idstats メッセージドロップカウントが増加しているが、送信されたレコードが増加しない。	コレクタが更新通知のフローに追従できていることを確認します。
サブスクリプションレシーバが接続されているが、更新が受信されない。	show telemetry internal subscription カウントが変化しない。	変更時のサブスクリプションの場合は、実際にイベントが発生していないことを確認します。 定期的なサブスクリプションの場合は、更新期間が短く、時間は 100 分の 1 秒単位で指定されていることを確認します。

show telemetry internal connection : このコマンドは、オプションの接続インデックス値を取ります。インデックスが指定されていない場合は、使用されているすべての接続の基本的な接続パラメータ情報が表示されます。コマンドで接続インデックスを指定すると、接続に関する詳細が表示されます。コマンド出力はトランスポート固有であり、すべてのトランスポートで使用できるとは限りません。このコマンドの出力は変更される場合があります。

show telemetry internal diagnostics : このコマンドは、すべてのテレメトリログと動作状態をダンプしようとします。問題を報告するときは、可能な限り問題の発生時のすぐ後にこのコマンドを使用し、**show running-config | section telemetry** コマンドの出力も提供すると、解決に役立ちます。

変更時サブスクリプション YANG モデルの表示

Cisco-IOS-XE-mdt-capabilities-oper.YANG モデルをクエリして、変更時サブスクリプションおよびそのトランスポートでサポートされるモデルについての情報を表示できます。

サブスクリプションのモニタリング

CLIおよび管理プロトコル操作を使用して、すべてのタイプのサブスクリプションを監視できます。

CLI

テレメトリのサブスクリプションに関する情報を表示するには、**show telemetry ietf subscription** コマンドを使用します。コマンドからの出力例を、次に示します。

```
Device# show telemetry ietf subscription 2147483667 detail
```

```
Telemetry subscription detail:
```

```
Subscription ID: 2147483667
State: Valid
Stream: yang-push
Encoding: encode-xml
Filter:
  Filter type: xpath
  XPath: /mdt-oper:mdt-oper-data/mdt-subscriptions
Update policy:
  Update Trigger: periodic
  Period: 1000
Notes:
```

NETCONF

次に、テレメトリのサブスクリプションに関する情報を表示する NETCONF メッセージの例を示します。

```
<get>
<filter>
<mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
<mdt-subscriptions/>
</mdt-oper-data>
</filter>
</get>

* Enter a NETCONF operation, end with an empty line
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-subscriptions>
        <subscription-id>101</subscription-id>
        <base>
          <stream>yang-push</stream>
          <encoding>encode-kvgpb</encoding>
          <source-vrf>RED</source-vrf>
          <period>10000</period>
          <xpath>/ios:native/interface/Loopback[name="1"]</xpath>
        </base>
        <type>sub-type-static</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
          <address>5.22.22.45</address>
          <port>57500</port>
```

```

        <protocol>grpc-tcp</protocol>
        <state>rcvr-state-connecting</state>
        <comments/>
        <profile/>
        <last-state-change-time>1970-01-01T00:00:00+00:00</last-state-change-time>
    </mdt-receivers>
    <last-state-change-time>1970-01-01T00:00:00+00:00</last-state-change-time>
</mdt-subscriptions>
<mdt-subscriptions>
  <subscription-id>2147483648</subscription-id>
  <base>
    <stream>yang-push</stream>
    <encoding>encode-xml</encoding>
    <source-vrf/>
    <period>1000</period>
  </base>
</mdt-subscriptions>
<xpath>/if:interfaces-state/interface[name="GigabitEthernet0/0"]/oper-status</xpath>
</base>
<type>sub-type-dynamic</type>
<state>sub-state-valid</state>
<comments/>
<mdt-receivers>
  <address>5.22.22.45</address>
  <port>51259</port>
  <protocol>netconf</protocol>
  <state>rcvr-state-connected</state>
  <comments/>
  <profile/>
  <last-state-change-time>1970-01-01T00:00:00+00:00</last-state-change-time>
</mdt-receivers>
<last-state-change-time>1970-01-01T00:00:00+00:00</last-state-change-time>
</mdt-subscriptions>
</mdt-oper-data>
</data>
</rpc-reply>

```

ストリーム

ストリームは、サブスクライブ可能な一連のイベントを定義します。ほぼすべてのイベントがこの一連のイベントとして有効です。ただし、各ストリームの定義に従い、すべてのイベントの候補は何らかの形で関連しています。ここでは、サポートされているストリームについて説明します。

サポートされているストリームのセットを表示するには、管理プロトコル操作を使用して、*mdt-streams* コンテナにある Cisco-*IOS-XE-mdt-oper* モジュール（YANG モデル Cisco-*IOS-XE-mdt-oper.yang* からのもの）から *streams* テーブルを取得します。

次に、NETCONF を使用して、サポートされているストリームを取得する例を示します。

```

<get>
<filter>
<mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
<mdt-streams/>
</mdt-oper-data>
</filter>
</get>

```

* Enter a NETCONF operation, end with an empty line

```
<?xml version="1.0" encoding="UTF-8"?>
```



```

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-streams>
        <stream>native</stream>
        <stream>yang-notif-native</stream>
        <stream>yang-push</stream>
      </mdt-streams>
    </mdt-oper-data>
  </data>
</rpc-reply>

```

この例は、native、yang-notif-native、yang-push の3つのストリームがサポートされていることを示しています。ストリーム native は汎用としては使用できず、無視できます。



(注) 現在のところ、サポートされているストリームのリストを返す CLI はありません。

YANG-push ストリーム

yang-push ストリームは、サポートされている YANG モデルにより記述される、構成データベース内と運用データベース内のデータです。このストリームは、ストリームの中で対象とするデータを指定するための XPath フィルタをサポートしており、XPath 式は対象のデータを定義する YANG モデルに基づきます。

このストリームの更新通知は、対象のサブスクリプションについて、データの変更時または固定間隔で送信される場合がありますが、両方に対応して送信されることはありません。現在存在しないデータのサブスクリプションは許可され、通常のサブスクリプションとして実行されます。

サポートされている唯一のターゲットデータベースは「実行中」です。

変更時機能の決定

現在のところ、変更時サブスクリプションを使用し、サブスクライブ可能なデータのタイプについて YANG モデルの中で指定する手段はありません。変更時サブスクリプションを使用し、サブスクライブができないデータにサブスクライブしようとする、失敗（動的）となるか、無効なサブスクリプション（設定済み）となります。On-Change パブリケーションの詳細については、「*On-Change Publication for yang-push*」の項を参照してください。

IETF ドラフトへの準拠

yang-push ストリームを使用するテレメトリは、テレメトリの IETF NETCONF ワーキンググループの初期ドラフトに基づいています。これらを次に示します。

- [イベント通知のカスタム サブスクリプション、バージョン 03](#)
- [YANG データストア プッシュ更新のサブスクライブ、バージョン 07](#)



(注) 対応するドラフトに記載されている次の機能はサポートされていません。

- サブツリー フィルタ
- アウトオブバンドの通知
- サポート対象として明示的に記載されていないすべてのサブスクリプション パラメータ

YANG-push の XPath フィルタ

サブスクライブ先の *yang-push* ストリーム内のデータセットは、XPath フィルタを使用して指定する必要があります。XPath 式には次のガイドラインが適用されます。

- XPath 式では、リストまたはコンテナに 1 つのエントリを指定するためのキーを持たせることができます。サポートされているキー指定の構文は次のとおりです。

```
[{key name}={key value}]
```

XPath 式の例を次に示します。

```
filter xpath
/rt:routing-state/routing-instance[name="default"]/ribs/rib[name="ipv4-default"]/routes/route

# VALID!
```

複合キーを使用するには、複数のキー指定を使用します。キーの名前と値は正確である必要があります。範囲やワイルドカードによる値はサポートされていません。

- XPath 式で、キーの間に [] を使用して複数のキーを選択し、". で文字列をカプセル化します。XPath 式の例を次に示します。

```
filter xpath
/environment-ios-xe-oper:environment-sensors/environment-sensor[location="\Switch\1\"]
[name="\Inlet\ Temp\ Sens\"]/current-reading
```

- XPath 式では、単一のサブスクリプションで複数のオブジェクトをサポートできるように、結合演算子 (|) を使用できます。ユニオン演算子は NETCONF トランスポートでのみ機能し、gRPC では機能しません。

Cisco Catalyst 9800 ワイヤレスコントローラでサポートされる XPath 式

Cisco IOS XE Bengaluru、17.4.1 では、次の OpenConfig XPath 式のセットが Cisco Catalyst 9800 シリーズ ワイヤレスコントローラでサポートされています。

テレメトリ サブスクリプションを有効にするには、NETCONF、RESTCONF、gNMI プロトコルなどのプログラマビリティ インターフェイスを使用して、次の RPC を実行します。

```
<rpc xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <target>
        <running/>
      </target>
    </target>
    <config>
      <provision-aps xmlns="http://openconfig.net/yang/wifi/ap-manager">
```

```

    <provision-ap>
      <mac>eth_mac_of_the_AP</mac>
      <config>
        <mac>eth_mac_of_the_AP</mac>
        <hostname>AP_NAME</hostname>
      </config>
    </provision-ap>
  </provision-aps>
</config>
</edit-config>
</rpc>

```

次に示す XPath 式はすべて、openconfig-access-points YANG モデルの一部です。ただし、最後の式だけは openconfig-ap-manager YANG モデルの一部です。テレメトリ操作が正しく機能するように、OpenConfig モデルに基づいて設定が行われていることを確認します。

- /access-points/access-point/radios/radio/state
- /access-points/access-point/radios/radio/neighbors/neighbor
- /access-points/access-point/radios/radio/neighbors/neighbor/state
- /access-points/access-point/ssids/ssid/bssids/bssid/state/counters
- /access-points/access-point/ssids/ssid/clients/client/state/counters
- /access-points/access-point/ssids/ssid/clients/client/client-rf/state
- /access-points/access-point/ssids/ssid/clients/client/client-connection/state
- /access-points/access-point/system/aaa/server-groups/server-group/servers/server/radius/state
- /joined-aps/joined-ap/state/opstate

XPath をサブスクライブすると、サブスクライブされた XPath とその階層内のすべての XPath のデータを受信します。たとえば、/access-points/access-point/radios/radio/state へサブスクライブすると、関連付けられているすべてのリーフとその下のサブコンテナのデータが配信されません。

情報のサブセットのみが必要な場合は、XPath 式でフィルタを設定して更新を制限します。特定のアクセスポイント (AP) のデータをフィルタリングするには、ノードの後にキーを使用します。たとえば、ホスト名が 'my_hostname' である AP のデータを受信するには、サブスクリプション XPath: access-point[hostname='my_hostname'] を使用します。データ更新には、定義済みの限定されたサブセットだけでなく、すべてのリーフからのデータオブジェクトが含まれることに注意してください。

拡張性に関する情報

次の表に、3 つの異なるスケールシナリオにおけるそれぞれの収集ポイントの最小推奨間隔を示します。

シナリオ 1 : 4 つの SSID によるフルスケール

表 9: 設定

AP	2,000
----	-------

クライアント	30,000
AP ごとの SSID	4
AP ごとの BSSID	8
AP ごとの物理ネイバー	12
AP ごとのネイバー	96

表 10: 推奨間隔

収集ポイント	レコード	推奨間隔 (秒)	
		コレクタ X1	コレクタ X2
参加	2000	30	60
AAA	2000	30	60
無線	4000	30	60
クライアント RF	30,000	30	60
クライアント CNTR	30,000	30	60
クライアント CONN	30,000	60	120
BSSID	16,000	90	180
Neighbor	192,000	180	360

シナリオ 2: 6つの SSID によるフルスケール

表 11: 設定

AP	2,000
クライアント	30,000
AP ごとの SSID	6
AP ごとの BSSID	12
AP ごとの物理ネイバー	12
AP ごとのネイバー	144

表 12: 推奨間隔

収集ポイント	レコード	推奨間隔 (秒)	推奨間隔 (秒)
		コレクタ X1	コレクタ X2
参加	2000	30	60
AAA	2000	30	60
無線	4000	30	60
クライアント RF	30,000	30	60
クライアント CNTR	30,000	30	60
クライアント CONN	30,000	60	120
BSSID	24000	120	240
Neighbor	288,000	240	420

シナリオ 3 : 6つの SSID による減少スケール

表 13: 設定

AP	1,000
クライアント	15,000
AP ごとの SSID	6
AP ごとの BSSID	12
AP ごとの物理ネイバー	12
AP ごとのネイバー	144

表 14: 推奨間隔

収集ポイント	レコード	推奨間隔 (秒)	推奨間隔 (秒)
		コレクタ X1	コレクタ X2
参加	1000	NA	30
AAA	1000	NA	30

収集ポイント	レコード	推奨間隔 (秒) コレクタ X1	推奨間隔 (秒) コレクタ X2
無線	2000	NA	30
クライアント RF	15,000	NA	30
クライアント CNTR	15,000	NA	30
クライアント CONN	15,000	NA	30
BSSID	12,000	NA	120
Neighbor	144,000	該当なし	180

Cisco Catalyst 9800 ワイヤレスコントローラの XPath 値と対応するレート

Cisco-IOS-XE-wireless-mesh-rpc の XPath /exec-linktest-ap/data-rate-idx で許容されている値と対応するレートを次に示します。

```

ewlc-mesh-linktest-rate-idx-1 1 Mbps
ewlc-mesh-linktest-rate-idx-2 2 Mbps
ewlc-mesh-linktest-rate-idx-3 5 Mbps
ewlc-mesh-linktest-rate-idx-4 6 Mbps
ewlc-mesh-linktest-rate-idx-5 9 Mbps
ewlc-mesh-linktest-rate-idx-6 11 Mbps
ewlc-mesh-linktest-rate-idx-7 12 Mbps
ewlc-mesh-linktest-rate-idx-8 18 Mbps
ewlc-mesh-linktest-rate-idx-9 24 Mbps
ewlc-mesh-linktest-rate-idx-10 36 Mbps
ewlc-mesh-linktest-rate-idx-11 48 Mbps
ewlc-mesh-linktest-rate-idx-12 54 Mbps
ewlc-mesh-linktest-rate-idx-13 108 Mbps
ewlc-mesh-linktest-rate-idx-14 m0
ewlc-mesh-linktest-rate-idx-15 m1
ewlc-mesh-linktest-rate-idx-16 m2
ewlc-mesh-linktest-rate-idx-17 m3
ewlc-mesh-linktest-rate-idx-18 m4
ewlc-mesh-linktest-rate-idx-19 m5
ewlc-mesh-linktest-rate-idx-20 m6
ewlc-mesh-linktest-rate-idx-21 m7
ewlc-mesh-linktest-rate-idx-22 m8
ewlc-mesh-linktest-rate-idx-23 m9
ewlc-mesh-linktest-rate-idx-24 m10
ewlc-mesh-linktest-rate-idx-25 m11
ewlc-mesh-linktest-rate-idx-26 m12
ewlc-mesh-linktest-rate-idx-27 m13
ewlc-mesh-linktest-rate-idx-28 m14
ewlc-mesh-linktest-rate-idx-295 m15

```

YANG-push の定期パブリケーション

定期的なサブスクリプションでは、サブスクリプション対象情報による最初のプッシュ更新は即時に送信されます。ただしデバイスがビジー状態であったりネットワークが混雑していたりする

と遅延することがあります。次に更新は、設定された定期タイマーの満了時に送信されます。たとえば、期間を 10 分と設定すると、サブスクリプションの作成直後に最初の更新が送信され、その後は 10 分おきに送信されます。

期間は、定期的なプッシュ更新間のセンチ秒 (1/100 秒) 単位の時間です。期間が 1000 であれば、サブスクライブ対象情報の更新は 10 秒ごとになります。設定できる最小の期間間隔は 100 (つまり 1 秒) です。デフォルト値はありません。この値は、動的サブスクリプションの場合は <establish-subscription> RPC で明示的に設定する必要がある、設定済みサブスクリプションの場合は設定で明示的に設定する必要があります。

定期的な更新には、サポートされているすべてのトランスポートプロトコルに関連するサブスクライブ対象のデータ要素またはテーブルのフルコピーが含まれています。

定期的なサブスクリプションを使用して空のデータをサブスクライブすると、要求された期間で空の更新通知が送信されます。データが存在するようになると、次の期間の値が通常の更新通知として送信されます。

YANG-push の変更時パブリケーション

変更時サブスクリプションを作成する場合は、ダンプニング期間がないことを示すためにダンプニング期間を 0 に設定する必要があります。その他の値はサポートされていません。

変更時サブスクリプションでは、最初のプッシュ更新は、サブスクライブされたデータのセット全体です (IETF の文書で定義されている初期同期)。これは制御できません。以降の更新は、データが変更され、変更後のデータのみで構成されている場合に送信されます。ただし、変更とみなされる最小のデータ分解能は行です。したがって、変更時サブスクリプションが行内のリーフに対するものである場合、その行のいずれかの項目が変更されると、更新通知が送信されます。更新通知の正確な内容はトランスポートプロトコルによって異なります。

また、変更時サブスクリプションは階層状ではありません。つまり、子コンテナを持つコンテナにサブスクライブしても、子コンテナ内の変更はサブスクリプションには認識されません。

現在存在しないデータのサブスクリプションは許可され、通常のサブスクリプションとして実行されます。初期同期更新通知は空であり、データが利用可能になるまでそれ以上更新されません。

XPath 式は単一のオブジェクトを指定する必要があります。このオブジェクトには、コンテナ、リーフ、リーフリスト、またはリストを使用できます。

yang-notif-native ストリーム

yang-notif-native ストリームは、パブリッシャ内の任意の YANG 通知であり、通知の元のイベントソースで Cisco IOS XE のネイティブのテクノロジーが使用されています。このストリームは、対象となる通知を指定する XPath フィルタもサポートしています。このストリームの更新通知は、通知の対象になるイベントが発生した場合にのみ送信されます。

このストリームは変更時サブスクリプションのみをサポートしているため、ダンプニング間隔として値 0 を指定する必要があります。

yang-notif-native の XPath フィルタ

サブスクライブ先の yang-notif-native ストリーム内のデータセットは、XPath フィルタを使用し指定します。次のガイドラインが XPath 式に適用されます。

- XPath 式は YANG 通知全体を指定する必要があります。属性のフィルタ処理はサポートされていません。
- ユニオン演算子 (|) はサポートされていません。

TLDP 変更時の通知

Targeted Label Discovery Protocol (T-LDP) は、直接接続されていないラベルスイッチドルーター (LSR) 間の LDP セッションです。TLDP 変更時の通知機能は、TLDP セッションが起動または停止したとき、および TLDP が設定またはディセーブルになったときにユーザに通知します。通知を機能させるには、TLDP を有効にする必要があります。

イベントベースの通知は、次の 2 つのシナリオで生成されます。

- 設定されたイベントは、TLDP が設定され、デバイスから削除されたときに生成されます。通知は、TLDP セッションがアップまたはダウンしたときにも生成されます。
- 通知は、TLDP セッションがアップまたはダウンしたときにも生成されます。

トランスポート プロトコル

データの送信方法は、パブリッシャと受信者間の接続に使用されるプロトコルによって決まります。このプロトコルはトランスポートプロトコルと呼ばれ、設定済みサブスクリプションの管理プロトコルからは独立しています。トランスポートプロトコルは、データのエンコーディング (XML、Google Protocol Buffers (GPB) など) と更新通知自体の形式に影響を与えます。



(注) また、選択したストリームも更新通知の形式に影響を与える場合があります。

サポートされているトランスポートプロトコルは、gNMI、gRPC、NETCONF です。

NETCONF プロトコル

NETCONF プロトコルは、動的サブスクリプションのトランスポートにのみ使用でき、yang-push ストリームと yang-notif-native ストリームで使用できます。

NETCONF をトランスポートプロトコルとして使用する場合は、次の 3 つの更新通知形式が使用されます。

- サブスクリプションで yang-push ストリームが使用されていて、定期的な場合、または、初期同期更新通知が変更時サブスクリプションで送信される場合。
- サブスクリプションで yang-push ストリームが使用されていて、初期同期更新通知以外の変更時サブスクリプションの場合。

- サブスクリプションで `yang-notif-native` ストリームが使用されている場合。

yang-push 形式

`yangpush` ソースストリームが NETCONF を介して XML エンコーディングのトランスポートとして送信される場合、2つの更新通知形式が定義されます。これらの更新通知形式は、*draft-ietf-netconf-yang-push-07*に基づいています。詳細については、IETF ドラフトの 3.7 項を参照してください。

yang-notif-native 形式

ソース ストリームが `yang-notif-native` の場合、NETCONF を介して XML でエンコードされるときの更新通知の形式は RFC 7950 によって定義されています。詳細については、RFC の 7.16.2 項を参照してください。

`yang-push` ストリームの形式とは異なり、サブスクリプション ID は更新通知にはありません。

gRPC プロトコル

gRPC プロトコルは、設定されたサブスクリプションのトランスポートにのみ使用でき、`yang-push` ストリームと `yang-notif-native` ストリームで使用できます。gRPC トランスポートプロトコルでは kvGPB エンコーディングのみがサポートされています

gRPC プロトコルに基づく受信者の接続の再試行（指数バックオフ）がサポートされています。

proto ファイルで定義されたテレメトリメッセージについては、`mdt_grpc_dialout.proto` および <https://github.com/cisco-ie/cisco-proto/blob/9cc3967cb1cabbb3e9f92f2c46ed96edf8a0a78b/proto/x/telemetry.proto> を参照してください。

gRPC テレメトリの相互認証

gRPC は、テレメトリデータの送信に使用されるサポートされているダイヤルアウトプロトコルの 1 つです。ダイヤルアウトプロトコルでは、デバイスはクライアントと見なされ、コネクタはサーバーと見なされます。gRPC は、暗号化されていない TCP 接続と暗号化された TLS ベースの接続の両方をサポートします。

相互認証にクライアント ID 証明書を使用できるように、トラストポイントのペアを含む新しい gRPC-TLS プロファイルがテレメトリ構成に追加されました。プロファイルには 2 つのトラストポイントが含まれています。1 つはサーバー検証用の認証局 (CA) 証明書で、もう 1 つはクライアント検証用の ID 証明書です。

デバイスがレシーバに初めて接続するとき、サーバーの設定に基づいて、クライアント認証または相互認証が必要になる場合があります。デバイスはレシーバの ID 証明書を受信すると、その証明書が、レシーバプロファイルで設定されたトラストポイントに関連付けられた証明書で識別された CA によって署名されているかどうかを検証します。次に、受信者がデバイスの証明書 ID を要求すると、デバイスは、プロファイルの ID トラストポイントフィールドに以前にインストールされたクライアント ID 証明書を送信します。

相互認証を要求するようにサーバーが設定されていて、プロファイルにクライアント ID トラストポイントがない場合、クライアント認証は行われず、接続も成功しません。

同じトラストポイントラベルを複数のプロファイルに設定でき、同じプロファイルを複数のレシーバに設定できます。



- (注) クライアント ID を持つトラストポイントは、gRPC over TLS では相互認証が要求されないため、プロファイル設定では必須ではありません。以前のリリースと同様に、gRPC over TLS はサーバー検証でのみ設定できます。

クライアント ID トラストポイントを追加するには、**telemetry protocol grpc profile <name>** コマンドを使用します。

この機能は無効にできません。ただし、レシーバでgRPC-TLSプロトコルを使用するようにを設定しないか、レシーバ設定のクライアント ID トラストポイントフィールドを削除するか設定しないことで、未使用のままにすることができます。

テレメトリにおけるハイアベイラビリティ

テレメトリの動的な接続は、アクティブなスイッチかスイッチスタック内のメンバーへのSSH、またはハイアベイラビリティ対応デバイスでのアクティブなルートプロセッサへのSSHを介してNETCONFセッションで確立されます。切り替え後は、テレメトリのサブスクリプションを送信するNETCONFセッションを含め、暗号を使用するすべてのセッションを破棄し、再確立する必要があります。また、スイッチオーバー後にすべてのダイナミックサブスクリプションを再作成する必要があります。gNMI ダイアルイン サブスクリプションも、SSH を介したNETCONFセッションと同様に機能します。

gRPCダイアルアウトサブスクリプションは、アクティブなスイッチまたはスタックメンバの実行コンフィギュレーションの一部としてデバイスに設定されます。スイッチオーバーが発生すると、テレメトリ受信者への既存の接続が切断され、再接続されます（受信者へのルートが残っている限り）。サブスクリプションを再設定する必要はありません。



- (注) デバイスのリロード時には、サブスクリプションの設定をデバイスのスタートアップコンフィギュレーションに同期させる必要があります。これにより、デバイスの再起動後もサブスクリプション設定がデバイス上にそのまま残ります。必要なプロセスが起動して実行されると、デバイスはテレメトリ受信者への接続を試行し、通常動作を再開します。

Pubd の再起動可能性

Cisco IOS XE Cupertino 17.9.1 では、すべてのプラットフォームで pubd プロセスを再起動できます。このリリースより前は、pubd は特定のプラットフォームでのみ再起動可能でした。他のプラットフォームで pubd プロセスを再起動するには、デバイス全体を再起動する必要がありました。

pubd は、必要に応じて NETCONF-YANG または gNXI 設定を削除して再度追加することで、再起動できます。これにより、他の NETCONF-YANG や gNXI プロセスも再起動されることに注意してください。

サンプルのモデル駆動型テレメトリ RPC

次のセクションでは、RPCの例のリストを示し、サブスクリプションの設定方法について説明します。

設定済みサブスクリプションの管理



(注) 現在のところ、設定済みサブスクリプションの管理に使用できるのは gRPC プロトコルのみです。

手順の概要

1. **enable**
2. **configure terminal**
3. **telemetry ietf subscription *id***
4. **stream yang-push**
5. **filter xpath *path***
6. **update-policy {on-change | periodic} *period***
7. **encoding encode-kvgpb**
8. **source-vrf *vrf-id***
9. **source-address *source-address***
10. **receiver ip address *ip-address* *receiver-port* **protocol** *protocol* **profile** *name***
11. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	telemetry ietf subscription <i>id</i> 例： Device(config)# telemetry ietf subscription 101	テレメトリのサブスクリプションを作成し、テレメトリサブスクリプションモードを開始します。
ステップ 4	stream yang-push 例： Device(config-mdt-subs)# stream yang-push	サブスクリプションのストリームを設定します。

	コマンドまたはアクション	目的
ステップ 5	filter xpath path 例： Device(config-mdt-subs)# filter xpath /memory-ios-xe-oper:memory-statistics/memory-statistic	サブスクリプションの XPath フィルタを指定します。
ステップ 6	update-policy {on-change periodic} period 例： Device(config-mdt-subs)# update-policy periodic 6000	サブスクリプションの定期的な更新ポリシーを設定します。
ステップ 7	encoding encode-kvgpb 例： Device(config-mdt-subs)# encoding encode-kvgpb	kvGPB エンコードを指定します。
ステップ 8	source-vrf vrf-id 例： Device(config-mdt-subs)# source-address Mgmt-intf	ソースの VRF インスタンスを設定します。
ステップ 9	source-address source-address 例： Device(config-mdt-subs)# source-vrf 192.0.2.1	送信元アドレスを設定します。
ステップ 10	receiver ip address ip-address receiver-port protocol protocol profile name 例： Device(config-mdt-subs)# receiver ip address 10.28.35.45 57555 protocol grpc-tcp	通知の受信者の IP アドレス、プロトコル、およびプロファイルを設定します。
ステップ 11	end 例： Device(config-mdt-subs)# end	テレメトリサブスクリプションのコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

gRPC の変更時サブスクリプションの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **telemetry ietf subscription id**
4. **stream yang-push**
5. **filter xpath path**
6. **update-policy {on-change | periodic period}**
7. **encoding encode-kvgpb**
8. **receiver ip address ip-address receiver-port protocol protocol profile name**

9. end

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	telemetry ietf subscription id 例： Device(config)# telemetry ietf subscription 8	テレメトリのサブスクリプションを作成し、テレメトリサブスクリプションモードを開始します。
ステップ 4	stream yang-push 例： Device(config-mdt-subs)# stream yang-push	サブスクリプションのストリームを設定します。
ステップ 5	filter xpath path 例： Device(config-mdt-subs)# filter xpath /iosxe-oper:ios-oper-db/hwidb-table	サブスクリプションの XPath フィルタを指定します。
ステップ 6	update-policy {on-change periodic period} 例： Device(config-mdt-subs)# update-policy on-change	サブスクリプションの変更時更新ポリシーを設定します。
ステップ 7	encoding encode-kvgpb 例： Device(config-mdt-subs)# encoding encode-kvgpb	kvGPB エンコードを指定します。
ステップ 8	receiver ip address ip-address receiver-port protocol protocol profile name 例： Device(config-mdt-subs)# receiver ip address 10.22.22.45 45000 protocol grpc_tls profile secure_profile	通知の受信者の IP アドレス、プロトコル、およびプロファイルを設定します。
ステップ 9	end 例： Device(config-mdt-subs)# end	テレメトリサブスクリプションのコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

応答コードの受信

サブスクリプションが正常に作成されると、デバイスはサブスクリプション結果 `notif-bis:ok` およびサブスクリプション ID で応答します。次に、動的サブスクリプションの応答 RPC メッセージの例を示します。

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
<subscription-result xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
xmlns:notif-bis="urn:ietf:params:xml:ns:yang:ietf-event-notifications">notif-bis:
ok</subscription-result>
<subscription-id
xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications">2147484201</subscription-id>
</rpc-reply>
```

NETCONF ダイアラインのサブスクリプションプッシュ更新の受信

デバイスからプッシュされるサブスクリプション更新は XML RPC 形式であり、それらが作成された同じ NETCONF セッションにより送信されます。サブスクライブ対象情報の要素またはツリーは `datastore-contents-xml` タグ内で返されます。次に示すのは、サブスクライブ対象情報を提供するサンプル RPC メッセージです。

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-05-09T21:34:51.74Z</eventTime>
<push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
<subscription-id>2147483650</subscription-id>
<datastore-contents-xml>
<cpu-usage
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-process-cpu-oper"><cpu-utilization>
<five-minutes>5</five-minutes></cpu-utilization></cpu-usage>
</datastore-contents-xml>
</push-update>
</notification>
```

サブスクリプションが行われる情報要素が空である場合、またはそれが動的（名前付きアクセスリストなど）であり存在しない場合、定期更新は空になり、自己終結 `datastore-contents-xml` タグを持つこととなります。次に示すのは、定期更新が空である RPC メッセージの例です。

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-05-09T21:34:09.74Z</eventTime>
<push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
<subscription-id>2147483649</subscription-id>
<datastore-contents-xml />
</push-update>
</notification>
```

サブスクリプションの詳細の取得

現在のサブスクリプションの一覧を取得するには、`<get>` RPC を `Cisco-IOS-XE-mdt-oper` モデルに送信します。現在のサブスクリプションの一覧を表示するには、`show telemetry ietf subscription` コマンドも使用できます。

次に、`<get>` RPC メッセージの例を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
        <mdt-subscriptions/>
      </mdt-oper-data>
    </filter>
  </get>
</rpc>
```

次に、RPC 応答の例を示します。

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-subscriptions>
        <subscription-id>2147485164</subscription-id>
        <base>
          <stream>yang-push</stream>
          <encoding>encode-xml</encoding>
          <period>100</period>
          <xpath>/ios:native/router/ios-rip:rip/ios-rip:version</xpath>
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <updates-in>0</updates-in>
        <updates-dampened>0</updates-dampened>
        <updates-dropped>0</updates-dropped>
      </mdt-subscriptions>
    </mdt-oper-data>
  </data>
</rpc-reply>
```

次に、`show telemetry ietf subscription dynamic brief` コマンドの出力例を示します。

```
Device# show telemetry ietf subscription dynamic brief

Telemetry subscription brief

  ID                Type      State      Filter type
  -----
  2147483667         Dynamic   Valid      xpath
  2147483668         Dynamic   Valid      xpath
  2147483669         Dynamic   Valid      xpath
```

次に、`show telemetry ietf subscription subscription-IDdetail` コマンドの出力例を示します。

```
Device# show telemetry ietf subscription 2147483667 detail

Telemetry subscription detail:

  Subscription ID: 2147483667
```

```

State: Valid
Stream: yang-push
Encoding: encode-xml
Filter:
  Filter type: xpath
  XPath: /mdt-oper:mdt-oper-data/mdt-subscriptions
Update policy:
  Update Trigger: periodic
  Period: 1000
Notes:

```

次に、**show telemetry ietf subscription all detail** コマンドの出力例を示します。

```
Device# show telemetry ietf subscription all detail
```

```
Telemetry subscription detail:
```

```

Subscription ID: 101
Type: Configured
State: Valid
Stream: yang-push
Encoding: encode-kvgpb
Filter:
  Filter type: xpath
  XPath: /iosxe-oper:ios-oper-db/hwidb-table
Update policy:
  Update Trigger: on-change
  Synch on start: Yes
  Dampening period: 0
Notes:

```

次の RPC の例は、RESTCONF を使用してサブスクリプションの詳細を取得する方法を示します。

Subscription details can also be retrieved through a RESTCONF GET request to the Cisco-IOS-XE-mdt-oper database:

URI:

```
https://10.85.116.28:443/restconf/data/Cisco-IOS-XE-mdt-oper:mdt-oper-data/mdt-subscriptions
```

Headers:

```
application/yang-data.collection+json, application/yang-data+json,
application/yang-data.errors+json
```

Content-Type:

```
application/yang-data+json
```

Returned output:

```

{
  "Cisco-IOS-XE-mdt-oper:mdt-subscriptions": [
    {
      "subscription-id": 101,
      "base": {
        "stream": "yang-push",
        "encoding": "encode-kvgpb",
        "source-vrf": "",
        "no-synch-on-start": false,
        "xpath": "/iosxe-oper:ios-oper-db/hwidb-table"
      },
      "type": "sub-type-static",
      "state": "sub-state-valid",
      "comments": "",
      "updates-in": "0",
      "updates-dampened": "0",
    }
  ]
}

```



```

    "updates-dropped": "0",
    "mdt-receivers": [
      {
        "address": "5.28.35.35",
        "port": 57555,
        "protocol": "grpc-tcp",
        "state": "rcvr-state-connecting",
        "comments": "Connection retries in progress",
        "profile": ""
      }
    ]
  }
}

```

CLI を使用した名前付きプロトコルレシーバの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **telemetry receiver protocol** *receiver-name*
4. **protocol** {**cloud-native** | **cntp-tcp** | **cntp-tls profile** *profile-name* | **grpc-tcp** | **grpc-tls profile** *profile-name* | **native** | **tls-native profile** *profile-name*}
5. **host** {**ip** *ip-address* | **name** *hostname*} *receiver-port*
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	telemetry receiver protocol <i>receiver-name</i> 例： Device(config)# telemetry receiver protocol receiver1	名前付きプロトコルレシーバを設定し、テレメトリプロトコルレシーバ コンフィギュレーション モードを開始します。
ステップ 4	protocol { cloud-native cntp-tcp cntp-tls profile <i>profile-name</i> grpc-tcp grpc-tls profile <i>profile-name</i> native tls-native profile <i>profile-name</i> } 例： Device(config-mdt-protocol-receiver)# protocol grpc-tcp	名前付きプロトコルレシーバ接続のプロトコルを設定します。

名前付きレシーバを使用したサブスクリプションの設定（CLIを使用）

	コマンドまたはアクション	目的
ステップ 5	host { <i>ip ip-address</i> <i>name hostname</i> } <i>receiver-port</i> 例： Device(config-mdt-protocol-receiver)# host name rcvr.test.com 45000	名前付きプロトコルレシーバのホスト名を設定します。
ステップ 6	end 例： Device(config-mdt-protocol-receiver)# end	テレメトリプロトコルレシーバコンフィギュレーションモードを終了し、特権EXECモードに戻ります。

名前付きレシーバを使用したサブスクリプションの設定（CLIを使用）

手順の概要

1. **enable**
2. **configure terminal**
3. **telemetry ietf subscription *id***
4. **receiver-type protocol }**
5. **receiver name *name***
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバルコンフィギュレーションモードを開始します。
ステップ 3	telemetry ietf subscription <i>id</i> 例： Device(config)# telemetry ietf subscription 101	テレメトリのサブスクリプションを作成し、テレメトリサブスクリプションモードを開始します。
ステップ 4	receiver-type protocol } 例： Device(config-mdt-subs)# receiver-type protocol	プロトコルタイプレシーバを設定します。
ステップ 5	receiver name <i>name</i> 例： Device(config-mdt-subs)# receiver name receiver1	通知ためのレシーバの名前を設定します。

	コマンドまたはアクション	目的
ステップ 6	end 例 : Device(config-mdt-sub)# end	テレメトリのテレメトリサブスクリプションモードを終了し、特権 EXEC モードに戻ります。

モデル駆動型テレメトリに関するその他の参考資料

関連資料

関連項目	マニュアルタイトル
YANG エクスプローラ	https://github.com/CiscoDevNet/yang-explorer

標準および RFC

標準/RFC	タイトル
イベント通知のカスタム サブスクリプション <i>draft-ietf-netconf-subscribed-notifications-03</i>	https://tools.ietf.org/id/draft-ietf-netconf-subscribed-notifications-03.txt
イベント通知の <i>NETCONF</i> サポート	draft-ietf-netconf-netconf-event-notifications-01
<i>RFC 5277</i>	NETCONF イベント通知
<i>RFC 6241</i>	ネットワーク設定プロトコル (NETCONF)
<i>RFC 7950</i>	YANG 1.1 データ モデリング言語
<i>RFC 8040</i>	RESTCONF プロトコル
イベント通知への登録	draft-ietf-netconf-rfc5277bis-01
YANG データストア プッシュのサブスクリプション	draft-ietf-netconf-yang-push-04
YANG データストア プッシュ更新のサブスクリプション <i>draft-ietf-netconf-yang-push-07</i>	https://tools.ietf.org/id/draft-ietf-netconf-yang-push-07.txt

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	<p>http://www.cisco.com/support</p>

モデル駆動型テレメトリの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 15: モデル駆動型テレメトリの機能情報

機能名	リリース	機能情報
モデル駆動型テレメトリ NETCONF ダイアルイン	Cisco IOS XE Everest 16.6.1	<p>モデル駆動型テレメトリでは、ネットワーク デバイスからサブスクライバに、リアルタイムの設定や運用状態の情報を継続的にストリーミングすることができます。</p> <ul style="list-style-type: none"> • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ
	Cisco IOS XE Everest 16.6.2	<ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ
	Cisco IOS XE Fuji 16.7.1	<ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ (ASR1001-HX、ASR1001-X、ASR1002-HX、ASR1002-X)
	Cisco IOS XE Fuji 16.8.1	<ul style="list-style-type: none"> • Cisco 1000 シリーズ サービス統合型ルータ • Cisco ASR 1000 RP2 および RP3 シリーズ アグリゲーション サービス ルータ
	Cisco IOS XE Fuji 16.8.1a	<ul style="list-style-type: none"> • Cisco Catalyst 9500 ハイパフォーマンス シリーズ スイッチ

機能名	リリース	機能情報
	Cisco IOS XE Fuji 16.9.1	<ul style="list-style-type: none"> • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Gibraltar 16.9.2	<ul style="list-style-type: none"> • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU
	Cisco IOS XE Gibraltar 16.10.1	<ul style="list-style-type: none"> • Cisco クラウド サービス ルータ 1000v • Cisco Network Convergence System 520 シリーズ
	Cisco IOS XE Gibraltar 16.11.1	<ul style="list-style-type: none"> • Cisco Catalyst 9600 シリーズ スイッチ

機能名	リリース	機能情報
モデル駆動型テレメトリ gNMI ダイヤルイン	Cisco IOS XE Gibraltar 16.12.1	<p>イニシエータ/サブスクリイバに送信されるテレメトリの更新は、ダイヤルと呼ばれます。</p> <p>この機能は、次のプラットフォームに実装されていました。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ
	Cisco IOS XE Amsterdam 17.1.1	<ul style="list-style-type: none"> • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Amsterdam 17.2.1	Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ

機能名	リリース	機能情報
モデル駆動型テレメトリ gRPC ダイヤルアウト	Cisco IOS XE Gibraltar 16.10.1	

機能名	リリース	機能情報
		<p>設置済みサブスクリプションでは、パブリッシャが受信者への接続を開始し、それらの接続はダイヤルアウトと見なされます。</p> <p>この機能は、次のプラットフォームに実装されていました。</p> <ul style="list-style-type: none"> • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンドルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ

機能名	リリース	機能情報
		<ul style="list-style-type: none"> • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Gibraltar 16.11.1	<ul style="list-style-type: none"> • Cisco Catalyst 9600 シリーズ スイッチ
モデル駆動型テレメトリ : サブスクリプションの kill	Cisco IOS XE Gibraltar 16.11.1	<p>動的サブスクリプションを削除するには、CLI および kill-subscription RPC を使用できます。</p> <ul style="list-style-type: none"> • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ (RSP2) • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ

機能名	リリース	機能情報
TLDP 変更時の通知	Cisco IOS XE Amsterdam 17.2.1	<p>TLDP 変更時の通知機能は、TLDPセッションが起動または停止したとき、および TLDP が設定またはディセーブルになったときにユーザに通知します。</p> <p>この機能は、次のプラットフォームに実装されていました。</p> <ul style="list-style-type: none">• Cisco 4000 シリーズ サービス統合型ルータ• Cisco Catalyst 9200 シリーズ スイッチ• Cisco Catalyst 9300 シリーズ スイッチ• Cisco Catalyst 9400 シリーズ スイッチ• Cisco Catalyst 9500 シリーズ スイッチ

機能名	リリース	機能情報
GRPC ダイアルアウト用のTLS	Cisco IOS XE Amsterdam 17.1.1	

機能名	リリース	機能情報
		<p>トランスポート層セキュリティは、gRPCダイヤルアウトでサポートされます。この機能は、次のプラットフォームでサポートされます。</p> <ul style="list-style-type: none"> • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-40 シリーズ ワイヤレス コントローラ • Cisco Catalyst 9800-80 シリーズ ワイヤレス コントローラ • Cisco cBR-8 コンバージド

機能名	リリース	機能情報
		ブロードバンドルータ <ul style="list-style-type: none">• Cisco Cloud Services Router 1000V シリーズ• Cisco Network Convergence System 520 シリーズ• Cisco Network Convergence System 4200 シリーズ

機能名	リリース	機能情報
gRPC サブスクリプションの FQDN サポート	Cisco IOS XE Bengaluru 17.6.1	

機能名	リリース	機能情報
		<p>gRPCサブスクリプション機能のFQDNサポートの導入により、IPアドレスに加え、FQDNもgRPCサブスクリプションに使用できます。</p> <ul style="list-style-type: none"> • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイパフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-40 シリーズ ワイヤレス コントローラ • Cisco Catalyst 9800-80 シリーズ ワイヤレス コントローラ • Cisco cBR-8 コンバージド

機能名	リリース	機能情報
		<p>ブロードバンドルータ</p> <ul style="list-style-type: none"> • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ
リーフレベルのフィルタリング	Cisco IOS XE Cupertino 17.7.1	<p>テレメトリ機能におけるリーフレベルのフィルタリングにより、最適化されたコードパスで収集ポイントレベル以下のフィルタリングが可能になります。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイパフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ

機能名	リリース	機能情報
gRPC テレメトリの相互認証	Cisco IOS XE Cupertino 17.9.1	<p>相互認証にクライアントID証明書を指定できるように、トラストポイントのペアを含む新しい gRPC TLS プロファイルがテレメトリ構成に追加されました。この新しいプロファイルは、受信者プロファイルを設定するときに、サーバー CA 証明書を含まないトラストポイントの代わりに使用できます。サーバー CA 証明書を含まないトラストポイントは、gRPC TLS プロファイルの一部として構成されました。</p> <p>この機能は、次のプラットフォームでサポートされます。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9800-CL シリーズワイヤレスコントローラ • Cisco Catalyst 9800-40 シリーズワイヤレスコントローラ • Cisco Catalyst 9800-80 シリーズワイヤレスコントローラ

機能名	リリース	機能情報
Pubd の再起動可能性	Cisco IOS XE Cupertino 17.9.1	

機能名	リリース	機能情報
		<p>このリリース以降、pubd プロセスが再起動可能になりました。</p> <p>この機能は、次のプラットフォームでサポートされます。</p> <ul style="list-style-type: none"> • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-CL シリーズ ワイヤレス コントローラ • Cisco Catalyst 9800-40 シリーズ ワイヤレス コントローラ

機能名	リリース	機能情報
		<ul style="list-style-type: none">• Cisco Catalyst 9800-80 シリーズワイヤレスコントローラ• Cisco cBR-8 コンバージドブロードバンドルータ• Cisco Cloud Services Router 1000V シリーズ• Cisco Network Convergence System 520 シリーズ• Cisco Network Convergence System 4200 シリーズ

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。