gNMI：管理インターフェイス

gNMI について

gNMI（gRPC ネットワーク管理インターフェイス、gRPC Network Management Interface）は gRPC 上で構成されます。gNMI を構成すると、ネットワークデバイスの構成および動作状態の編集、読み取りに役立ちます。また、ネットワークデバイスは、指定されたデータ収集システムへのテレメトリストリームを生成できます。

（注） gNMI は、gRPC 機能の子サービス機能としてのみ構成できます。

Cisco NX-OSは、次の表に示すすべての gNMI RPC をサポートします。

表 *1:* サポートされる *gNMI RPC*

gNMI RPC	サポート対象
機能	はい
get	はい
設定	はい
登録	はい

RPC を登録するために、 Cisco NX-OSは次のサブモードをサポートしています。

表 *2:* サブスクライブ オプション

タイプ	サブタイプ	サポート対象	説明
[1 回（Once）]		はい	スイッチは、指定されたすべてのパスに対して現在の値を 1 回だけ送信します。
ポーリング（Poll）		はい	スイッチは、ポーリング メッセージを受信するたびに、指定されたすべてのパスの現在の値を送信します。
ストリーム	サンプル	はい	ストリーム サンプル間隔ごとに 1 回、スイッチは指定されたすべてのパスの現在の値を送信します。サポートされるサンプル間隔の範囲は 1 ～ 604800 秒です。 デフォルトのサンプル間隔は 10 秒です。

タイプ	サブタイプ	サポート対象	説明
	[変更時（On_Change）]	はい	スイッチは初期状態として現在の値を送信しますが、値の更新は、指定されたパスで作成、変更、または削除などの変更が発生した場合にのみ行われます。

gNMIサブスクリプションは、一般にダイヤルインテレメトリと呼ばれます。gNMI では、ネットワークデバイスで要求を開始する外部の顧客を必要とします。

Cisco NX-OSは、ネットワーキングデバイスが外部受信者にテレメトリデータをプッシュする、別のテレメトリ機能をサポートしています。これはダイヤルアウトテレメトリと呼ばれます。詳細については、「テレメトリ」のセクションを参照してください。

Cisco NX-OS リリース 10.4(3)F 以降、OpenConfig パスの openconfig-system:/system/processes は、On-change gnmi サブスクリプションをサポートします。このパスは、gnmi-proto と gnmi-json の両方のエンコーディングをサポートします。サブスクリプションは、クライアントが同じパスのサブスクリプションを解除するまでアクティブなままです。

効率を高めるために、可能な限り厳密なサブスクリプションを作成することをお勧めします。たとえば、cpu-usage-user をモニターにする場合、/system/processes などの一般的なパスに On-target gnmi サブスクリプションを使用するのではなく、データを直接登録すれば、イベントデータを回避できます。

gNMI に関する注意事項と制限事項

gNMI に関するガイドラインと制限事項は次のとおりです。

- 次のような既存の CLI 構成を使用して OpenConfig ルーティングポリシーをサブスクライブしようとすると、OpenConfig モデルの現在の実装により、空の値が返されます。

  ```
  ip prefix-list bgp_v4_drop seq 5 deny 125.2.0.0/16 le 32
   ipv6 prefix-list bgp_v6_drop seq 5 deny cafe:125:2::/48 le 128
  ```

  以下のパスを使用します。

  ```
  openconfig-routing-policy:/routing-policy/defined-sets/prefix-sets/prefix-set[name=bgp_v4_drop]/config
  openconfig-routing-policy:/routing-policy/defined-sets/prefix-sets/prefix-
  set[name=bgp_v6_drop]/config
  ```

- origin、use_models、またはその両方の使用は、gNMI サブスクリプションではオプションです。

- Cisco NX-OS 9.3(x) リリースより前にサポートされるプラットフォームについては、そのリリース向けガイドの「プログラマビリティ機能のプラットフォームサポート」を参照してください。Cisco NX-OS リリース 9.3(x) 以降でサポートされているプラットフォームについては、『Nexus Switch Platform Support Matrix』を参照してください。

- この機能は、JSON および gNMI proto エンコーディングをサポートします。
- この機能は、protobuf any エンコーディングをサポートしていません。
- この機能はサブスクリプション要求のパスプレフィックスをサポートしていませんが、サブスクリプションには空のプレフィックスフィールドを含めることができます。

ワイルドカードパス

- パスではマルチレベルワイルドカード「...」は使用できません。
- パスの先頭にワイルドカード「*」を使用することはできません
- キー名でワイルドカード「*」を使用することはできません
- キーでのワイルドカードと値は互換性がありません

show grpc gnmi コマンドには、次のガイドラインと制限事項があります。

- このコマンドは、xml または json 出力フォーマットをサポートしていません。
- gRPC エージェントは、呼び出しが終了した後、最大 1 時間 gNMI 呼び出しを保持します。
- コールの合計数が 2000 を超えると、gRPC エージェントは、内部クリーンアップルーチンに基づいて終了したコールを消去します。

表 *3: gNMI* 要求のワイルドカード サポート

リクエストの種類	ワイルドカード サポート
gNMI GET	はい
gNMI 設定	いいえ
gNMI サブスクリプション（1 回）	はい
gNMI SUBSCRIBE、POLL	はい
gNMI SUBSCRIBE、STREAM、SAMPLE	はい
gNMI SUBSCRIBE、STREAM、TARGET_DEFINED	はい
gNMI SUBSCRIBE、STREAM、ON_CHANGE	いいえ

スケールの考慮事項

- 各 gNMI メッセージの最大サイズは 12 MB です。収集されたデータの量が最大値を超えると、収集されたデータはドロップされます。これは、gNMI ON_CHANGE モードにのみ適用されます。

  収集データのドロップを回避するには、小規模できめ細かいデータ収集セットを管理する、対象を絞ったサブスクリプションを作成してください。したがって、1 つの上位レベ

ルのパスに対するサブスクリプションの代わりに、パスの異なる下位レベルの部分に対する複数のサブスクリプションを作成します。

すべてのサブスクリプションで、最大 250K の集約 MO がサポートされます。より多くのMOに登録すると、収集データがドロップする可能性があります。

- すべてのサブスクリプションに対し、最大 250K の集約 MO がサポートされます。より多くのMOに登録すると、収集データのドロップに影響が出ます。

gNMI サブスクリプションに関するガイドラインと制限事項

- 変更時サブスクリプションは gnmi とテレメトリの両方で機能しますが、一度にアクティブにできるのはいずれか 1 つだけです。これらのエージェントのいずれかからサブスクリプションが作成された場合、既存のサブスクリプション情報は上書きされます。

  次に、openconfig-system:/system/processes の変更時 gnmiサブスクリプションの例を示します。

```
Request:
./gnmi-console_enhanced_plus --host 172.22.244.142 --port 50051 -u admin -p insieme
 --tls --cafile /tmp/grpc.pem --hostnameoverride ems.cisco.com  --operation=Subscribe
 --submode ON_CHANGE  --xpath "openconfig-system:system/processes/process[pid=1]"
-e JSON

Response:

////// initial snapshot

Received response 1 --------------------------
/system/
{
    "processes": {
        "process": [
            {
                "pid": "1",
                "state": {
                    "pid": "1",
                    "name": "init",
                    "start-time": "1706643350384761800",
                    "cpu-usage-user": "14",
                    "cpu-usage-system": "12",
                    "cpu-utilization": 0,
                    "memory-usage": "180224",
                    "memory-utilization": 0
                }
            }
        ]
    }
}

///// first event update

/system/
{
    "processes": {
        "process": [
            {
                "pid": "1",
                "state": {
                    "start-time": "1706643350384761800",
```

```
                    "cpu-usage-user": "14",
                    "cpu-usage-system": "12"
                }
            }
        ]
    }
}
```

gNMI の構成

gNMI 機能は、gRPC gNMI コマンド を使用して構成します。

gNMI オプションの構成

始める前に

gNMI は、gRPC 機能のみの子サービス機能です。

gRPC エージェントを有効にする方法については、gPRC エージェントのドキュメントを参照してください。

手順の概要

1. switch# **configure terminal**
2. （任意） **grpc gnmi max-cocrrent-call**
3. **grpc gnmi subscription target-defined min-interval**<*interval*>
4. **grpc gnmi subscription query-condition keep-data-timestamp**
5. （任意） **grpc gnmi keepalive-timeout** <*timeout*>

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	switch# **configure terminal**
	例：
	`switch# config terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	（任意） **grpc gnmi max-cocrrent-call** 例： `switch(config)# grpc gnmi max-concurrent-call 16`	このコマンドは、スイッチ上の gNMI サーバーに対する同時ダイヤルイン呼び出しの制限を設定します。デフォルトの制限は 8 です。値は、1 ～ 16 の制限範囲で設定できます。 構成する最大値は、各 VRF に対するものです。制限を 16 に設定し、gNMI が管理 VRF とデフォルト VRF

	コマンドまたはアクション	目的
		の両方に構成されている場合、各 VRF は 16 の同時 gNMI 呼び出しをサポートします。 このコマンドは、進行中または入力中の gNMI 呼び出しには影響しません。 gRPC は新しい呼び出しに制限を適用し、アクティブな呼び出しにはは影響を与えないため、アクティブな呼び出しは完了することができます。
ステップ 3	**grpc gnmi subscription target-defined min-interval**<*interval*> 例： `switch(config)# grpc gnmi subscription target-defined min-interval 60`	ターゲット定義のデフォルトのサンプル間隔を、30 秒から、必要な他の値に変更できます。
ステップ 4	**grpc gnmi subscription query-condition keep-data-timestamp** 例： `switch(config)# grpc gnmi subscription query-condition keep-data-timestamp`	このコマンドは、sample/once/poll サブスクリプションが、データが最後に更新されたときにデータベースからタイムスタンプを取得できるようにします。 （注） • このコマンドは PROTO でのみサポートされ、JSON コードではサポートされません。 • once、poll、sample をサポートしますが、on_change サブスクリプションはサポートしません。 • DME、YANG、および OpenConfig データソースにおいてサポートします。 • サポートされていないプロパティの場合、コマンドのデフォルトは、直前のデータベース変更時刻ではなく、収集時刻に戻ります。 • このコマンドは、タイムスタンプごとに冗長応答を生成します。時間どおりメッセージを収集できない場合、スイッチはメッセージの収集をドロップします。
ステップ 5	（任意） **grpc gnmi keepalive-timeout** <*timeout*> 例： `switch(config)# grpc gnmi keepalive-timeout 1200`	このコマンドを構成すると、非アクティブまたは無許可の接続を削除できます。gRPCエージェントは、クライアントに空の応答を定期的に送信します。この応答がクライアントに届かなかった場合、接続は停止します。

	コマンドまたはアクション	目的
		デフォルトの間隔値は 600 秒です。キープアライブインターバルを 600 ～ 86400 秒の間隔範囲で変更するように構成できます。

gNMI RPC

このセクションでは、それぞれの gNMI RPC について説明します。このセクションでは、参照クライアント *gnmi_cli* の使用方法を確認できます。

機能

RPC の機能は、gNMI サービスの機能のリストを返します。RPC 要求に対する応答メッセージには、gNMI サービスのバージョン、バージョンデータモデル、およびサーバでサポートされているデータエンコードが含まれます。

機能に関する注意事項と制限事項

次は機能に関するガイドラインと制限事項です。

クライアント出力の例

次に言及している例は、機能のクライアント出力を示しています。機能 *openconfig* が有効になっていることを確認します。

この例は、YANG モデル、gNMIバージョン、およびサポートされているエンコードのサポートを示しています。

```
$ ./gnmi_cli -a 172.1.1.1:50051 -ca_crt ./grpc.pem -insecure -capabilities
supported_models: <
  name: "Cisco-NX-OS-device"
  organization: "Cisco Systems, Inc."
  version: "2019-11-13"
>
supported_models: <
  name: "openconfig-acl"
  organization: "OpenConfig working group"
  version: "1.0.0"
>
supported_models: <
  name: "openconfig-bgp-policy"
  organization: "OpenConfig working group"
  version: "4.0.1"
>
.
.
.
supported_models: <
  name: "openconfig-spanning-tree"
  organization: "OpenConfig working group"
  version: "0.2.0"
>
supported_models: <
```

```
  name: "openconfig-system"
  organization: "OpenConfig working group"
  version: "0.3.0"
>
supported_models: <
  name: "openconfig-telemetry"
  organization: "OpenConfig working group"
  version: "0.5.1"
>
supported_models: <
name: "openconfig-vlan"
  organization: "OpenConfig working group"
  version: "3.0.2"
>
supported_models: <
  name: "DME"
  organization: "Cisco Systems, Inc."
>
supported_models: <
  name: "Cisco-NX-OS-Syslog-oper"
  organization: "Cisco Systems, Inc."
  version: "2019-08-15"
>
supported_encodings: JSON
supported_encodings: PROTO
gNMI_version: "0.5.0"
```

Get について

Get RPC を使用すれば、スイッチからデータツリーのスナップショットを取得できます。1 つの要求で複数のパスを要求できます。gNMI パスの規則に従って、単純な形式の XPATH を使用できます。「gNMI のスキーマパスエンコーディング規則」を参照してください。GET 操作の詳細については、『gRPC ネットワーク管理インターフェイス』の「状態情報のスナップショットの取得」セクションを参照してください。

Get に関する注意事項と制限事項

次に、Get のガイドラインと制限事項を示します。

- GetRequest エンコードは JSONフォーマットのみをサポートします。
- GetRequest.type の場合、DataType CONFIG と STATE のみが YANG 形式との直接の関係と式を持ちます。操作はサポートされていません。
- 1 つの要求に OpenConfig（OC）YANG パスとデバイス YANG パスの両方を含めることはできません。1 つの要求には 1 つのパスのみが含まれます。
- ルートパス（**すべて**のモデルからのすべてで「/」）に対する GetRequest は許可されていません。
- gNMI Get は、すべてのデフォルト値を返します。Report-all モード RFC 6243 [4]を参照してください。
- Get は Cisco-NX-OS-syslog-oper モデルをサポートしていません。

- openconfig-procmonデータを取得するには、クエリをパス /system/processes または /system に送信します。

（注） Cisco NX-OSリリース 10.3(x) より前では、 /system パスからデータを取得することはできません。

- 以下のオプションはサポートされていません。
  - パスのプレフィックス
  - パスのエイリアス
  - パス内のワイルドカード
- 1 つの GetReqeuestで 10 のパスを要求することができます。
- GetResponse のサイズ返り値が 12 MB を超える場合、システムはエラーステータス grpc::RESOURCE_EXHAUSTED を返します。
- gRPC の最大受信バッファサイズは 8 MB です。
- より多くの構成があるスイッチに対して Get 操作を実行すると、gRPC プロセスは使用可能なすべてのメモリを消費します。メモリが使い果たされると、次のsyslogが生成されます。

```
MTX-API: The memory usage is
        reaching the max memory resource limit (3072) MB
```

メモリが連続して使い果たされると、次の syslog が生成されます。

```
The process has become unstable and
        the feature should be restarted.
```

Cisco では、gNMI トランザクションを正常に機能させるため、gRPC 機能を再起動することをお勧めします。

- Get の合計同時セッションの最大数は、構成されている最大同時呼び出し数の 75% です。たとえば、MTX 同時呼び出しが 16 に構成されている場合、Get の合計同時セッションの最大数は 12 になります。
- Get と Set の同時セッションの合計数は、構成されている gNMI の同時最大数から 1 を引いたものです。

  たとえば、gNMI 同時呼び出しが 16 に構成されている場合、Get と Set の合計同時セッションの最大数は 15 になります。

Set について

Set RPC を使用して、スイッチの構成を変更できます。スイッチの削除、交換、更新が行えます。1 つの Set 要求に含まれるこれらすべての操作は、1 つのトランザクションと見なされて、すべて正常に完了するか、またはスイッチが元の状態にとどまるかのいずれかになります。

Set 操作は、Set Request 構成で指定された順序で実行されます。パスが複数回要求されている場合、パスが互いを上書きする場合でも、変更は実行されます。データの最終状態は、トランザクションの最終操作によって実現されます。フィールドの削除、置換、更新など、構成データパスであり、Set 要求で指定されているパスは編集できます。

Set 操作の詳細については、gNMI の仕様の「状態の変更」のセクションを参照してください。

Set に関する注意事項と制限事項

次に、Set のガイドラインと制限事項を示します。

- SetRequest エンコードは JSONフォーマットのみをサポートします。
- 1 つの要求に OpenConfig（OC）YANG パスとデバイス YANG パスの両方を含めることはできません。1 つの要求には 1 つのパスのみが含まれます。
- 以下のオプションはサポートされていません。
  - パスのプレフィックス
  - パスのエイリアス
  - パス内のワイルドカード
- 1 つの SetReqeuest で 20 のパスを要求できます。
- gRPC の最大受信バッファサイズは 8 MB です。
- Get と Set の合計同時セッションの最大数は、構成されている最大 gNMI 同時呼び出し数です。たとえば、gNMI 同時呼び出しが 16 に構成されている場合、Get と Set の合計同時セッションの最大数は 15 になります。
- スイッチに対して Set::Delete RPC 操作を実行した場合、構成が大規模だと、下に示すような MTX 警告メッセージが生成されます。

```
Configuration size for this
       namespace exceeds operational limit. Feature may become unstable and require

        restart.
```

Set の拡張機能

Set RPC では、次の拡張機能がサポートされています。

拡張の Confirm-Commit

この拡張機能により、新しい設定によってデバイスへの接続が失われた場合に、適用された設定を自動的にロールバックできます。すべての protobuf メッセージの定義を含む拡張の詳細については、次を参照してください：https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-commit-confirmed.md

gnmi confirm-commit 機能は、既存の netconf confirm-commit の簡素化されたサブセットです。gnmi confirm-commit トランザクションの場合、拡張は単一の **Set** 操作に適用されます。この **Set** 操作は、指定された時間内に確認される必要があります。そうしないと、設定リクエストの操作が以前の状態にロールバックされます。**Set** 操作を確認するのではなく、明示的にキャンセルすることもできます。

この拡張機能の一般的な使用例は簡単です。gnmi クライアントは通常のように **Set** 操作を使用して構成変更を適用しますが、**Commit** 拡張メッセージも含まれています。**Commit** メッセージには、文字列識別子と、ロールバック期間を設定する **CommitRequest** メッセージが含まれます。このロールバック期間が期限切れになる前に、指定された id 文字列で識別されるこの **CommitRequest** メッセージを確認する必要があります。**CommitRequest** は、データを含めず、拡張機能で **CommitConfirm** メッセージを伝送する別の **Set** 要求を送信することによって確認されます。構成の変更によってネットワーク接続が中断された場合、この確認メッセージはデバイスで受信されず、構成は以前の構成にロールバックされます。クライアントが希望する場合は、構成をすぐにキャンセルしてロールバックするために **CommitCancel** メッセージを送信することもできます。ロールバック期間タイマーを延長するメッセージもあります。CommitConfirm メッセージが正常に送信されて **CommitRequest Set** 操作が確認されると、適用された設定が保持されます。メッセージの詳細については、上記のリファレンスを参照してください。

Confirm-Commit のガイドラインと制限事項

gnmiconfirmed-commit 機能を使用して設定をプッシュする場合、ユーザーがいくつかの点に留意する必要があります。

- 構成変更の意図と効果を理解する必要がある。たとえば、ユーザーはインターフェイスをシャットダウンしたり、grpc 自体を無効にしたりすることができます。そのような場合、CommitConfirm メッセージを受信できないため、このような場合、confirm-commit 拡張は意味を持ちません。
- 場合によっては、構成の変更により一時的にネットワーク接続が中断され、回復に時間がかかることがあります。この一時的な中断を予測し、確認が行われるときに構成が安定した状態に達するように、適切なタイムアウト期間を選択する必要があります。構成変更に不適切なタイムアウトを選択すると、不要な構成ロールバックが発生する可能性があります。このタイムアウトの選択は、ユーザーの責任です。
- 管理者とオペレータは、構成のロールバックの可能性を認識している必要があります。これは、ログ収集やその他のデバッグ/モニタリングツールに干渉する可能性があるためです。

- CommitRequest メッセージを使用して Set 操作が適用され、「id」が確立されると、他の Set 操作は受け入れられません。アクティブなセット（「id」と一致）は、確認、キャンセル、またはタイムアウトを許可する必要があります。
- CommitConfirm、CommitCancel、または CommitSetRollbackDuration メッセージを伝送する Set オペレーションは、更新/置換/削除オペレーションのデータを伝送することはできません。操作対象のパスを含むセット（トランザクションの初期セット）とともに伝送できるのは、CommitRequestメッセージだけです。
- 1 つの CommitConfirm トランザクションはいつでもアクティブになる可能性があります。
- タイムアウト値の最小単位は、Duration.seconds で指定された秒レベルのみです。指定された期間のナノ秒以外の値がある場合、その期間は拒否されます。

コマンドの表示

次のコマンドは、gnmi の確認済みコミットトランザクションに関する情報を表示するように拡張されています。以下の例は、Set 要求の後で、Lo1 インターフェイスの説明を変更し、600 秒のタイムアウトで commit-id を「commitId-123」に設定した後です。

- `show grpc gnmi transactions`

  このコマンドは、コミット「id」、タイムアウト期間、メッセージタイプ（CommitRequest、CommitConfirm、CommitCancel など）などの confirm-commit 拡張の gnmi メッセージレベルの詳細を表示します。

```
RPC           DataType   Session         Time In              Duration(ms) Status
------------ ---------- --------------- -------------------- ------------ ------
Set           -          3               06/16 20:02:51       5            0
Commit Type: CommitConfirm   Persist: commitId-123         Timeout :600


Set           -          2               06/16 20:01:41       75           0
subtype: dtx:  st: path:
Replace  -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo1]/descr
Commit Type: CommitRequest   Persist: commitId-123         Timeout :600


Set           -          1               06/16 20:00:55       13           0
```

- `show grpc internal mtx commit history`

  このコマンドは、セッション ID やタイムスタンプなどの内部状態情報を表示します。

```
MTX Commit History
```

```
Session    Op      Persist          PersistId        Timeout      Ret    Timestamp

3          COMMIT                   commitId-123                  1      2025-06-16
  20:02:51

2          COMMIT  commitId-123                      600          1      2025-06-16
  20:01:41
```

- `show grpc internal mtx db`

  このコマンドは、DB レベルの内部状態情報を表示します。

```
MTX DME DB state variables
Running Config locked : false
Session that locked Running : 18446744073709551615
Session doing confirmed commit : 18446744073709551615
Is a confirmed commit active : false
Thread Local Data instances : 1
Confirmed Commit start time : 2025-06-16 20:01:41
Confirmed Commit timeout : 600
Confirmed Commit time remaining : 0
Persistent Session id : 3
Commit Candidate to Running :
    Transaction Token : commitId-123
    Commit Type : COMMIT_CONFIRM_END
```

登録について

登録 RPC を使用すれば、特定のパスのテレメトリストリームを登録できます。登録されたパスに変更があると、スイッチは設定またはパスの状態の変更に関する通知をにプッシュします。

登録 RPCで使用可能な、オプションのフラグを使用できます。Cisco NX-OS リリース 9.3(1) 以降では、UPDATES_ONLY オプション フラグがサポートされています。これは、ON_CHANGE サブスクリプションにのみ適用されます。このオプションのフラグが設定されている場合、スイッチは現在の状態を抑制し、最初の応答とともに通知が送信されます。

これらのフラグを使用して、次の表に示すオプションへの応答を変更できます。

表 *4: SUBSCRIBE* フラグのサポートマトリクス

サブスクリプションタイプ	**heartbeat_interval**	**[冗長抑制（suppress_redundant）]**
ON_CHANGE	発信元：デバイス YANG、OpenConfig YANG、DME	該当なし
SAMPLE	発信元：デバイス YANG、OpenConfig YANG、DME	発信元：デバイス YANG、OpenConfig YANG

Cisco NX-OS リリース 10.2(3)F 以降、次のオプションフラグがサポートされています。

- heartbeat_interval
- [冗長抑制（suppress_redundant）]

サンプルサブスクリプションでの suppress_redundant の動作を変更するために、heartbeat_interval を指定できます。この場合、ターゲットは、suppress_redundant フラグがtrueに設定されているかどうかに関係なく、hearbeat_interval ごとに1つのテレメトリ更新を生成する必要があります。この値は、ナノ秒単位の符号なし64ビット整数として指定します。

sample サブスクリプションの場合は、suppress_redundant フラグを使用できます。この場合、設定ステータスはtrueです。ターゲットは、レポートされたパスの値が、最後の更新が生成されてから変更されていた場合、テレメトリ更新メッセージを生成します。更新は、サブスクリプションが変更された個々のリーフノードに対して生成されます。

たとえば、サブスクリプション名が「A」と「B」で、「B」ノードからブランチ「C」と「D」が出ていたとします。「D」の値ではなく、「C」の値が変更されたとします。更新は「C」に対してのみ生成され、「D」に対しては生成されません。

注意事項と制約事項

次に、Set サブスクリプションのガイドラインと制限事項を示します。

- 次のフラグはサポートされていません。
  - エイリアス
  - allow_aggregation
  - 内線番号
  - prefix
  - QoS（Dest. QoS）
- 同じサブスクリプション要求内のすべてのパスのサンプル間隔が同じになっていることを確認します。同じパスで異なるサンプル間隔が必要な場合は、複数のサブスクリプションを作成します。
- メモリが不足している状態では、OC サブスクリプション要求はメモリが使用可能になるのを待たずに、すぐに失敗する場合があります。
- システムによって過剰なイベントが生成された場合、OC イベント通知がドロップされる場合があります。
- サブスクリプションスナップショット中はOCイベント通知が無効になり、スナップショットが完了するとフラッシュされます。
- OC スナップショットの完了時間は30分に制限されています。その時間内に完了しない場合、サブスクリプションはキャンセルされます。

gNMI サブスクライブオプション

次の表に、サブスクライブオプションのモードを示します。

表 *5:*

Type	サブスクリプション タイプ	説明
ONCE		データを受信するように登録し、セッションを閉じます。
投票		登録すると、アクティブセッションを保持します。 ポーリング要求を発生させる場合は、要求ごとにデータを入力してください。
STREAM	SAMPLE	特定の頻度でデータを登録し、受信します。 ペイロード値の単位はナノ秒です。1 秒 = 1000000000 になります。
	ON_CHANGE	スナップショットを受信し、ツリーに変更がある場合にのみデータを受信するように登録します。
TARGET_DEFINED		作成できる最適なサブスクリプションを見つけるように登録します。

モードの設定

それぞれのモードは、サブスクリプション内部とサブスクリプション外部の2つの設定を必要とします。次に、サブスクリプションの組み合わせについて説明します。

- ONCE：SAMPLE、ONCE
- POLL：SAMPLE、POLL
- STREAM：SAMPLE、STREAM
- ON_CHANGE：ON_CHANGE、STREAM
- TARGET_DEFINED：TARGET_DEFINED、STREAM

Origin

- DME：DME モデルへの登録
- DEVICE：YANG モデルへの登録
- OPENCONFIG：OpenConfig モデルへの登録

名前（Name）

- DME：DME モデルへの登録
- Cisco-NX-OS-device：YANG モデルへの登録

エンコーディング（Encoding）

- JSON：ストリームは JSON 形式で送信される。
- PROTO：ストリームは元のプロトコルフォーマットで送信される。

ペイロードの構成例

次のセクションでは、ペイロードのクライアントの例を示します。

DME ストリーム/サンプルへの登録

```
{
    "SubscribeRequest":
    [
        {
            "subscribe":
            {
                "subscription":
                [
                    {
                        "path":
                        {
                            "origin": "DME",
                            "elem":
                            [
                                {
                                    "name": "sys"
                                },
                                {
                                    "name": "bgp"
                                 }
                             ]
                        },
                        "mode": "SAMPLE"
                    }
                ],
                "mode": "ONCE",
                "allow_aggregation" : false,
                "use_models":
                [
                    {
                        "name": "DME",
                        "organization": "Cisco Systems, Inc.",
                        "version": "1.0.0"
                    }
                ],
                "encoding": "JSON"
            }
        }
    ]
}
```

OpenConfig YANG/サンプルへの登録

```
{
    "SubscribeRequest":
    [
        {
            "subscribe":
            {
                "subscription":
                [
                    {
                        "path":
                        {
                            "origin": "openconfig",
                            "elem":
                            [
                                {
                                    "name": "interfaces"
                                }
                            ]
                        },
                        "mode": "SAMPLE",
                        "Sample_interval": 10000000000
                    }
                ],
                "mode": "ONCE",
                "allow_aggregation" : false,
                "use_models":
                [
                    {
                        "name": "openconfig-interfaces",
                        "organization": "OpenConfig working group",
                        "version": "0.8.1"
                    }
                ],
                "encoding": "JSON"
            }
        }
    ]
}
```

カスタム Syslog ストリームの登録について

Cisco NX-OSは、オリジナルのモデルと OpenConfig モデルをサポートします。gNMI の場合、または ON_CHANGE を登録する場合、カスタム YANG モデルは syslog イベントをストリーミングするように設計されています。この機能は、8 GB 以上のメモリを搭載した Cisco Nexus 9000 シリーズスイッチで構成できます。

注意事項と制約事項

Syslog ストリームに関するガイドラインと制限事項は次のとおりです。

- 無効な syslog はサポートされません。たとえば、フィルタまたはクエリ条件を含む syslog はサポートされません。
- Syslog ストリームは以下のパスのみをサポートします。

- Cisco-NX-OS-Syslog-oper: syslog
- Cisco-NX-OS-Syslog-oper: syslog messages

- ストリームサンプルモードと POLL モードのみがサポートされます。
- サポートされるエンコーディングフォーマットは JSON と PROTO です。

オリジナルの YANG Syslog モデル

次に、YANG Syslog モデルの構成例を示します。

（注） デフォルトでは、タイムゾーンフィールドは空です。タイムゾーンフィールドは、`clock format show-timezone syslog` を構成するときに設定されます。

```
PYANG Tree for Syslog Native Yang Model:
>>> pyang -f tree Cisco-NX-OS-infra-syslog-oper.yang module: Cisco-NX-OS-syslog-oper
+--ro syslog
+--ro messages
+--ro message* [message-id]
+--ro message-id int32
+--ro node-name? string
+--ro time-stamp? uint64
+--ro time-of-day? string
+--ro time-zone? string
+--ro category? string
+--ro group? string
+--ro message-name? string
+--ro severity? System-message-severity
+--ro text? string
```

サブスクライブ要求

次に、サブスクライブ要求の例を示します。

```
{
    "SubscribeRequest":
    [
        {
            "subscribe":
            {
                "subscription":
                [
                    {
                        "path":
                        {
                            "origin": "syslog-oper",
                            "elem":
                              [
                                {
                                    "name": "syslog"
                                },
                                {
                                    "name":"messages"
                                }
                              ]
                        },
                        "mode": "ON_CHANGE"
```

```
                    }
                ],
                "mode": "ON_CHANGE",
                "allow_aggregation" : false,
                "use_models":
                [
                    {
                        "name": "Cisco-NX-OS-Syslog-oper",
                        "organization": "Cisco Systems, Inc.",
                        "version": "0.0.0"
                    }
                ],
                "encoding":"JSON"
            }
        }
    ]
}
```

応答の例

次に、PROTO エンコードでの出力応答の例を示します。

```
[Subscribe]--------------------------------
Sat Aug 24 14:38:06 2019
### Generating request : 1 -----------
### Comment : STREAM request
### Delay : 2 sec(s) ...
### Delay : 2 sec(s) DONE

subscribe {
subscription {
path {
origin: "syslog-oper"
elem {
name: "syslog"
}
elem {
name: "messages"
}
}
mode: ON_CHANGE
}

use_models {
name: "Cisco-NX-OS-Syslog-oper"
organization: "Cisco Systems, Inc."
version: "0.0.0"
}
encoding: PROTO
}

Thu Nov 21 14:26:41 2019
Received response 3 --------------------------
update {
timestamp: 1574375201665688000
prefix {
origin: "Syslog-oper"
elem {
name: "syslog"
}
elem {
name: "messages"
}
}
…
```

```
update {
path {
elem {
name: "time-stamp"
}
}
val {
uint_val: 1574375200000
}
}
update {
path {
elem {
name: "severity"
}
}
val {
uint_val: 5
}
}
}

/Received -------------------------------------
```

次に、JSON エンコードでの出力応答の例を示します。

```
[Subscribe]--------------------------------
### Reading from file ' testing_bl/stream_on_change/OC_SYSLOG.json '

Tue Nov 26 11:47:00 2019
### Generating request : 1 -----------
### Comment : STREAM request
### Delay : 2 sec(s) ...
### Delay : 2 sec(s) DONE
subscribe {
subscription {
path {
origin: "syslog-oper"
elem {
name: "syslog"
}
elem {
name: "messages"
}
}
mode: ON_CHANGE
}
use_models {
name: "Cisco-NX-OS-Syslog-oper"
organization: "Cisco Systems, Inc."
version: "0.0.0"
}
}

Tue Nov 26 11:47:15 2019
Received response 5 --------------------------
update {
timestamp: 1574797636002053000
prefix {
}
update {
path {
origin: "Syslog-oper"
elem {
name: "syslog"
```

```
}
}
val {
json_val: "[ { \"messages\" : [[
{\"message-id\":657},{\"node-name\":\"task-n9k-1\",\"time-stamp\":\"1574797635000\",\"time-of-day\":\"Nov
 26 2019
11:47:15\",\"severity\":3,\"message-name\":\"HDR_L2LEN_ERR\",\"category\":\"ARP\",\"group\":\"ARP\",\"text\":\"arp
 [30318] Received packet with incorrect layer 2 address length (8 bytes), Normal pkt
with S/D MAC: 003a.7d21.d55e ffff.ffff.ffff eff_ifc mgmt0(9), log_ifc mgmt0(9), phy_ifc
 mgmt0(9)\",\"time-zone\":\"\"} ]] } ]"
}
}
}
```

gNMI のトラブルシューティング

show コマンドを実行して、gNMI のステータスを表示できます。特定の gNMI 構成を確認するには、次の表に記載されているコマンドを入力します。

表 *6 : show* コマンド

コマンド	説明
`show grpc gnmi service statistics`	それぞれの管理 VRF、またはデフォルト構成 VRF のエージェント実行ステータスのサマリーを表示します。また、次の項目も表示されます。 • 基本の全般的なカウンタ • 証明書の有効期限日時
`show grpc gnmi rpc summary`	次のステータスが表示されます。 • 受信した機能 RPC の数。 • RPC の機能のエラー。 • 受信した Get RPC の数。 • Get RPC エラー。 • 受信した Set RPC の数。 • Set RPC エラー。

コマンド	説明
show grpc gnmi transactions	

コマンド	説明
	show grpc gnmi transaction コマンドは最も密度が高く、かなりの情報が含まれています。これは、スイッチが受信した最新 50 の gNMI トランザクションの履歴バッファです。新しい RPC が着信すると、末尾から最も古い履歴エントリが削除されます。次に、表示内容について説明します。 このコマンドは、詳細情報を表示します。これは、スイッチが受信した最新 50 の gNMI トランザクションの履歴を表示します。新しい RPC が入力されると、最も古い入力が履歴から順に削除されます。次は表示される情報を示しています。 • RPC：受信した RPC のタイプ（Get、Set、機能）を示します。 • データタイプ（DataType）：Get の場合のみです。値は ALL、CONFIG、および STATE です。 • セッション（Session）：このトランザクションに割り当てられている一意のセッション ID を示します。他のログ ファイルで見つかったデータを関連付けるために使用できます。 • 入力時間（Time In）：gNMI ハンドラが RPC を受信したときのタイムスタンプを示します。 • 期間（Duration）：要求を受信してから応答を返すまでの時間差です（ミリ秒単位）。 • ステータス（Status）：クライアントに返された操作のステータス コードを示します（0 は成功、0 以外はエラー）。 次は、単一の gNMI トランザクション内のパスごとのデータを示します。たとえば、単一の Get または Set です。 • **サブタイプ（subtype）**：Set RPC の場合に、パスごとに要求される特定の操作（削除、更新、置換）を示します。Get の場合、サブタイプはありません。

コマンド	説明

	• **Dtx**：このパスが DTX ファストパスで処理されるかどうかを示します。ダッシュ（-）はいいえを示し、アスタリスク（*）ははいを示します。
	• **st**：このパスのステータスを表示します。表示されるステータスには次のものがあります。
	  • **OK**：パスは有効で、インフラによって正常に処理されました。
	  • **ERR**：パスが無効であるか、インフラによってエラーが生成されました
	  • **--**：パスはまだ処理されていません。または無効なため、インフラに送信されていません。

gRPC コマンドの構成例

gRPC gNMI サービス統計

次に、`show grpc gnmi service statistics` コマンドの出力例を示します。

```
==============
gRPC Endpoint
==============

Vrf : management
Server address : [::]:50051

Cert notBefore : Mar 13 19:05:24 2020 GMT
Cert notAfter : Nov 20 19:05:24 2033 GMT

Max concurrent calls : 8
Listen calls : 1
Active calls : 0

Number of created calls : 1
Number of bad calls : 0

Subscription stream/once/poll : 0/0/0

Max gNMI::Get concurrent : 5
Max grpc message size : 8388608
gNMI Synchronous calls : 74
gNMI Synchronous errors : 0
gNMI Adapter errors : 0
gNMI Dtx errors : 0
```

gRPC gNMI rPC のサマリー

次に、show grpc gnmi service statistics コマンドの出力例を示します。

```
=============
gRPC Endpoint
=============

Vrf            : management
Server address : [::]:50051

Cert notBefore : Mar 31 20:55:02 2020 GMT
Cert notAfter  : Apr  1 20:55:02 2020 GMT

Capability rpcs    : 1
Capability errors  : 0
Get rpcs           : 53
Get errors         : 19
Set rpcs           : 23
Set errors         : 8
Resource Exhausted : 0
Option Unsupported : 6
Invalid Argument   : 18
Operation Aborted  : 1
Internal Error     : 2
Unknown Error      : 0

RPC Type        State      Last Activity  Cnt Req    Cnt Resp   Client
--------------- ---------- -------------- ---------- ---------- -----------------
-----------------------
Subscribe       Listen     04/01 07:39:21          0          0
```

gRPC gNMI トランザクション

次に、show grpc gnmi transaction コマンドの出力例を示します。

```
=============
gRPC Endpoint
=============

Vrf            : management
Server address : [::]:50051

Cert notBefore : Mar 31 20:55:02 2020 GMT
Cert notAfter  : Apr  1 20:55:02 2020 GMT

RPC          DataType   Session         Time In              Duration(ms) Status
------------ ---------- --------------- -------------------- ------------ ------
Set          -          2361443608      04/01 07:43:49       173          0
subtype: dtx:  st: path:
Delete   -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo789]

Set          -          3445444384      04/01 07:43:33       3259         0
subtype: dtx:  st: path:
Delete   -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo789]
Delete   -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo790]
…
Delete   -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo807]
Delete   -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo808]

Set          -          2297474560      04/01 07:43:26       186          0
subtype: dtx:  st: path:
Update   -     OK  /System/ipv4-items/inst-items/dom-items/Dom-list[name=foo]/rt-
items/Route-list[prefix=0.0.0.0/0]/nh-items/Nexthop-list[nhAddr=192.168.1.1/32][n
hVrf=foo][nhIf=unspecified]/tag
```

```
Set          -          0               04/01 07:43:11       0            3
subtype: dtx:  st: path:
Update   -     --  /System/intf-items/lb-items/LbRtdIf-list[id=lo4]/descr
Update   -     ERR /system/processes

Set          -          2464255200      04/01 07:43:05       708          0
subtype: dtx:  st: path:
Delete   -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo2]
Replace  -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo3]/descr
Replace  -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo4]/descr
Replace  -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo5]/descr
Update   -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo3]/descr
Update   -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo5]/descr


Set          -          3491213208      04/01 07:42:58       14           0
subtype: dtx:  st: path:
Replace  -     OK  /System/intf-items/lb-items/LbRtdIf-list[id=lo3]/descr

...

Get          ALL        2293232352      04/01 07:42:35       258          0
subtype: dtx:  st: path:
-        -     OK  /system

Get          ALL        0               04/01 07:42:33       0            12
subtype: dtx:  st: path:
-        -     --  /intf-items
```

デバッグ ログの収集

gNOIは、gRPCエージェントの子サービスです。詳細については、「診断と有用性」を参照してください。

gNMIのアカウンティング ログ

gNMI では、`Set RPC` はスイッチの構成を変更します。`SET` で構成の更新（`UPDATE`）、置き換え（`REPLACE`）、または削除（`DELETE`）をリクエストした場合、gNMIは対応するアカウンティングログを生成します。これらのログには、元のリクエストとスイッチで行われた変更の両方が含まれます。

アカウンティングログは、`show accouting log` コマンドを使用して表示できます。

次は、次の gNMI パスを使用する、`SetRequest`, encoding = JSON to localhost の例を示しています。

```
---
<<<<<<< set_delete >>>>>>>
[]
<<<<<<< set_replace >>>>>>>
[] []
<<<<<<< set_update >>>>>>>
[elem { name: "System"
} elem {
name: "tm-items"
} elem {
```

```
name: "certificate-items"
}
] [json_val: "{\"hostname\": \"test\", \"trustpoint\": \"foo\"}"
]
The SetRequest response is below ------------------------response { path {
elem {
name: "System"
} elem {
name: "tm-items"
} elem {
name: "certificate-items"
}
} op: UPDATE
} timestamp: 1656512303065384369
---
```

次の表に、スイッチで構成できるオプションを示します。アカウンティングログには次の項目が含まれます。

項目	説明
コンテキスト	セッション ID とユーザー
オペレーション	コミットまたは中止
データベース	実行または候補
ConfigMO	MO ツリーのテキスト表現。最大 3000 文字。
ステータス	成功または失敗

次にアカウンティングログのサンプルを示します。

```
Wed Jun 29 14:18:23
2022:type=update:id=1430425712:user=admin:cmd=(COMMIT),database=[candidate],
configMo=[<topSystem childAction="" dn="sys" status="created,modified"><telemetryEntity

childAction="" rn="tm" status="created,modified"><telemetryCertificate childAction=""
hostname="test" rn="certificate" status="created,modified"
trustpoint="foo"/></telemetryEntity></topSystem>] (SUCCESS)
Wed Jun 29 14:18:23
2022:type=update:id=1430425712:user=admin:cmd=(COMMIT:CANDIDATE-TO-RUNNING),
database=[running] (SUCCESS)
```

次の表に、スイッチで受信した元のリクエストを示します。

項目	説明
コンテキスト	セッション ID とユーザー
オペレーション	gNMI:SET:UPDATE, gNMI:SET:REPLACE, gNMI:SET:DELETE, COMMIT:CANDIDATE-TO-RUNNING
送信元 IP（Source IP）	gNMI クライアント IP

項目	説明
パス	テキスト フォーマットの gNMI パス
ペイロード	受け取った JSON 要求。最大 3000 文字。
ステータス	成功または失敗

```
Wed Jun 29 14:18:23
2022:type=update:id=1430425712:user=admin:cmd=(GNMI:SET:UPDATE),sourceIp=[192.168.1.2],

path=[/System/tm-items/certificate-items],payload=[{"hostname":"test","trustpoint":"foo"}]

(SUCCESS)
```

要求が失敗した場合、および失敗したシナリオに基づく場合、両方のログを表示することはできません。

無効な要求

無効な要求を送ると、この要求は構成を変更することなく拒否され、元の要求のみがログに記録されます。次に、無効な要求のログの例を示します。

```
Wed Jun 29 14:18:23
2022:type=update:id=1430425712:user=admin:cmd=(GNMI:SET:UPDATE),
sourceIp=[192.168.1.2],path=[/System/tm-items/certificateitems],
payload=[{"hostname":"test","trustpoint":"foo"}] (FAILED)
```

失敗した要求

要求を送り、それが何らかの構成上の制限で失敗した場合、元の要求と失敗した構成要求の両方がログに記録されます。次にログの例を示します。

```
Wed Jun 29 20:52:15
2022:type=update:id=1429663200:user=admin:cmd=(COMMIT),database=[candidate],
configMo=[<topSystem childAction="" dn="sys"
status="created,modified"><telemetryEntity childAction="" rn="tm"
status="created,modified"><telemetryCertificate childAction="" filename="foo"
hostname="test" rn="certificate" status="created,modified,replaced"
trustpoint="foo"/></telemetryEntity></topSystem>] (FAILED)

Wed Jun 29 20:52:15
2022:type=update:id=1429663200:user=admin:cmd=(GNMI:SET:REPLACE),
sourceIp=[192.168.1.2],path=[/System/tm-items/certificate-items],
payload=[{"hostname":"test","trustpoint":"foo","filename":"foo"}] (FAILED)
```

要求を送り、コミットに失敗した場合、元の要求と失敗した要求がログに記録されます。次にログの例を示します。

```
Wed Jun 29 14:18:23
2022:type=update:id=1430425712:user=admin:cmd
(COMMIT),database=[candidate],configMo=[<topSystem childAction="" dn="sys"
status="created,modified"><telemetryEntity childAction="" rn="tm"
status="created,modified"><telemetryCertificate childAction="" hostname="test"
rn="certificate" status="created,modified"
trustpoint="foo"/></telemetryEntity></topSystem>] (SUCCESS)

Wed Jun 29 14:18:23
2022:type=update:id=1430425712:user=admin:cmd=(GNMI:SET:UPDATE),
sourceIp=[192.168.1.2],path=[/System/tm-items/certificate-items],
payload=[{"hostname":"test","trustpoint":"foo"}] (SUCCESS)
```

```
Wed Jun 29 20:34:06
2022:type=update:id=1429665744:user=admin:cmd=(COMMIT:CANDIDATE-TO-RUNNING),
database=[running] (FAILED)
```

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。