Cisco Networking Service コンフィギュレーション ガイド

初版：2008 年 07 月 11 日

最終更新：2013 年 03 月 11 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system.All rights reserved.Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries.To view a list of Cisco trademarks, go to this URL:http://www.cisco.com/go/trademarks.Third-party trademarks mentioned are the property of their respective owners.The use of the word partner does not imply a partnership relationship between Cisco and any other company.(1110R)

© 2008-2013 Cisco Systems, Inc. All rights reserved.

目 次

最初にお読みください 1

Cisco Networking Service の設定 3

機能情報の確認 3

Cisco Networking Service の前提条件 4

Cisco Networking Service の制約事項 4

Cisco Networking Service について 5

Cisco Networking Service 5

Cisco Networking Service EXEC エージェント 5

Cisco Networking Service 結果メッセージ 6

Cisco Networking Service メッセージ フォーマット 6

Cisco Networking Service ID 9

Cisco Networking Service パスワード 10

Cisco Networking Service ゼロ タッチ 10

Cisco Networking Service の設定方法 11

Cisco Networking Service デバイスの配置 11

高度な Cisco Networking Service 機能の設定 14

Cisco Networking Service エージェントのトラブルシューティング 16

Cisco Networking Service の設定例 20

例：Cisco Networking Service デバイスの配置 20

例：Cisco Networking Service ゼロ タッチ ソリューションの使用 20

その他の参考資料 23

Cisco Networking Service の機能情報 24

CNS 設定エージェント 27

機能情報の確認 27

CNS 設定エージェントについて 27

Cisco Networking Service 設定エージェント 27

Cisco Networking Service の初期設定 28

Cisco Networking Service の差分設定 28
コンフィギュレーションの同期 28
CNS 設定エージェントの設定方法 29
Cisco Networking Service イベント エージェントおよび EXEC エージェントの設定 29
CNS 設定エージェントの設定例 31
例：Cisco Networking Service エージェントのイネーブル化および設定 31
例：Cisco Networking Service イメージのサーバからの取得 32
その他の参考資料 33
CNS 設定エージェントの機能情報 34
Cisco Networking Service 再試行/間隔指定の設定取得拡張 37
機能情報の確認 37
CNS 再試行/間隔指定の設定取得拡張について 38
Cisco Networking Service 再試行/間隔指定の設定取得拡張 38
CNS 再試行/間隔指定の設定取得拡張の設定方法 38
Cisco Networking Service 設定のサーバからの取得 38
CNS 再試行/間隔指定の設定取得拡張の設定例 39
例：Cisco Networking Service 設定のサーバからの取得 39
その他の参考資料 40
CNS 再試行/間隔指定の設定取得拡張の機能情報 41
Cisco Networking Service インタラクティブ CLI 43
機能情報の確認 43
CNS インタラクティブ CLI について 43
Cisco Networking Service インタラクティブ CLI 43
その他の参考資料 44
CNS インタラクティブ CLI の機能情報 44
コマンド スケジューラ（Kron） 47
機能情報の確認 47
コマンド スケジューラの制約事項 47
コマンド スケジューラ（Kron）について 48
コマンド スケジューラ 48
コマンド スケジューラ（Kron）の設定方法 48

コマンド スケジューラ ポリシー リストおよびオカレンスの設定 48
トラブルシューティングのヒント 52
コマンド スケジューラ（Kron）の設定例 52
例：コマンド スケジューラ ポリシー リストおよびオカレンス 52
その他の参考資料 53
コマンド スケジューラ（Kron）の機能情報 54
ネットワーク設定プロトコル 57
機能情報の確認 57
NETCONF の前提条件 58
NETCONF の概要 58
NETCONF 通知 58
NETCONF の設定方法 58
NETCONF ネットワーク マネージャ アプリケーションの設定 58
NETCONF ペイロードの配信 60
NETCONF 通知のフォーマット 61
NETCONF セッションのモニタリングおよびメンテナンス 65
NETCONF の設定例 66
例：NETCONF ネットワーク マネージャ アプリケーションの設定 66
例：NETCONF セッションのモニタリング 67
NETCONF に関する追加情報 69
NETCONF の機能情報 70
用語集 71
NETCONF over SSHv2 73
機能情報の確認 73
NETCONF over SSHv2 の前提条件 74
NETCONF over SSH の制約事項 74
NETCONF over SSHv2 について 74
NETCONF over SSHv2 74
NETCONF over SSHv2 の設定方法 76
ホスト名とドメイン名を使用した SSH バージョン 2 のイネーブル化 76
RSA キー ペアを使用した SSH バージョン 2 のイネーブル化 77
リモート デバイスとの暗号化セッションの開始 79
トラブルシューティングのヒント 80

次の作業 80
セキュア シェル接続のステータスの確認 80
NETCONF over SSHv2 のイネーブル化 81
NETCONF over SSHv2 の設定例 83
例：ホスト名およびドメイン名を使用した SSHv2 のイネーブル化 83
RSA キーを使用したセキュア シェル バージョン 2 のイネーブル化の例 83
リモート デバイスとの暗号化セッションの開始の例 83
NETCONF over SSHv2 の設定例 83
NETCONF over SSHv2 に関する追加情報 85
NETCONF over SSHv2 の機能情報 86
BEEP による設定への NETCONF アクセス 87
機能情報の確認 87
BEEP による設定への NETCONF アクセスの前提条件 88
BEEP による設定への NETCONF アクセスの制約事項 88
BEEP による設定への NETCONF アクセスについて 88
NETCONF over BEEP の概要 88
BEEP による設定への NETCONF アクセスの設定方法 90
SASL プロファイルの設定 90
NETCONF over BEEP のイネーブル化 91
BEEP による設定への NETCONF アクセスの設定例 95
例：NETCONF over BEEP のイネーブル化 95
BEEP による設定への NETCONF アクセスに関する追加情報 95
BEEP による設定への NETCONF アクセスの機能情報 96

第 1 章

最初にお読みください

Cisco IOS XE 16 に関する重要な情報

有効な 2 つのリリース、Cisco IOS XE Release 3.7.0E（Catalyst スイッチ用）および Cisco IOS XE Release 3.17S（アクセス/エッジルーティング用）が 1 つのバージョンとしての統合リリース Cisco IOS XE 16 に展開されて（マージされて）、スイッチング/ルーティング ポートフォリオに含まれる広範なアクセスおよびエッジ製品が盛り込まれた 1 つのリリースが実現されています。

（注） 技術構成ガイドにある機能情報の表には、その機能がいつ導入されたかが記載されています。その機能で他のプラットフォームがサポートされた時期については、記載されている場合もされていない場合もあります。特定の機能がご使用のプラットフォームでサポートされているかどうかを確認するには、製品のランディング ページに掲載されている技術構成ガイドを参照してください。ご使用の製品のランディング ページに技術構成ガイドが表示されていれば、その機能はプラットフォームでサポートされています。

第 2 章

Cisco Networking Service の設定

Cisco Networking Service（CNS）機能は、リモート イベント駆動型の Cisco IOS ネットワーキング デバイスの設定、および一部のコマンドライン インターフェイス（CLI）コマンドのリモート実行を可能にするサービスの集合です。

- 機能情報の確認，3 ページ
- Cisco Networking Service の前提条件，4 ページ
- Cisco Networking Service の制約事項，4 ページ
- Cisco Networking Service について，5 ページ
- Cisco Networking Service の設定方法，11 ページ
- Cisco Networking Service の設定例，20 ページ
- その他の参考資料，23 ページ
- Cisco Networking Service の機能情報，24 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

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。

Cisco Networking Service の前提条件

- Cisco Networking Service 設定エージェントおよび Cisco Networking Service イベント エージェントをサポートするよう、リモート デバイスが設定されていること。
- リモート デバイスの外部インターフェイスと互換性のあるトランスポート プロトコルが、そのリモート デバイスに設定されていること。次の表に、デバイス インターフェイスに応じて使用可能な、サポートされるトランスポート プロトコルを示します。
- Cisco Networking Service 設定エンジン プロビジョニング データベースに設定テンプレートが作成されていること（この作業は、上級ネットワーク設計者が行うのが最適です）。

表 *1*：デバイス インターフェイスおよび *Cisco Networking Service* サービスに必要なトランスポート プロトコル

デバイスインターフェイス	**SLARP** トランスポート プロトコル	**ATM InARP** トランスポート プロトコル	**PPP（IPCP）** トランスポート プロトコル
T1	○	○	○
ADSL	×	○	○
シリアル	○	×	○

Cisco Networking Service の制約事項

Cisco Networking Service 設定エンジン（CE）

- Cisco Networking Service 設定エンジンは、Cisco Intelligence Engine 2100（Cisco IE2100）シリーズである必要があり、ソフトウェア バージョン 1.3 を実行している必要があります。
- 設定エンジンは、設定を作成するための属性の情報データベースにアクセスできる必要があります。このデータベースは Cisco IE2100 自身にあってもかまいません。
- リモート デバイスを設置する前に、Cisco Networking Service 設定エンジンに設定テンプレートを準備しておく必要があります。
- Cisco Networking Service フロースルー プロビジョニングおよび Cisco Networking Service 設定エンジンのユーザは、ネットワーク トポロジの設計、設定テンプレートの設計、およびCisco Networking Service 設定エンジンの使用に精通している必要があります。

リモート デバイス

- リモート デバイスは、Cisco Networking Service 設定エージェントおよび Cisco Networking Service イベント エージェントをサポートする Cisco IOS イメージを実行する必要があります。
- ネットワークに接続できるように、リモートデバイスにポートを用意する必要があります。
- リモート デバイスは、Cisco Configuration Express を使用するように設定されている必要があります。

Cisco Networking Service について

Cisco Networking Service

Cisco Networking Service は、ユーザをネットワーキング サービスにリンクする基本テクノロジーで、大量のネットワーク デバイスを自動設定するためのインフラストラクチャを提供します。多くの IP ネットワークは複雑で多くのデバイスが存在し、現在のところは各デバイスを個別に設定する必要があります。標準設定が存在しない場合、または変更されている場合は、初期インストールとその後のアップグレードにかなりの時間がかかります。また、小規模化、標準化が進む顧客ネットワークの数の増加に、対応可能なネットワーク エンジニアの数の増加が追いついていません。現在、インターネット サービス プロバイダー（ISP）には、部分的な設定を送信して新しいサービスを導入するための手段が必要です。これらのすべての問題に対処するために、Cisco Networking Service は、中央のディレクトリ サービスと分散型エージェントを使用した、「プラグアンドプレイ」ネットワーク サービスを提供するように設計されています。Cisco Networking Service 機能には、Cisco Networking Service 設定エージェントとイベント エージェント、およびフロースループロビジョニング構造が含まれます。設定エージェントおよびイベントエージェントは、Cisco Networking Service 設定エンジンを使用してシスコ デバイスの初期設定、差分設定、および同期設定の更新を自動化するための方法を提供し、設定エンジンは、設定ロードのステータスをネットワーク モニタリングまたはワークフローアプリケーションが加入できるイベントとして報告します。Cisco Networking Service フロースループロビジョニングは、Cisco Networking Service 設定エージェントおよびイベント エージェントを使用して自動ワークフローを提供するため、現場に技術者がいる必要はありません。

Cisco Networking Service EXEC エージェント

CNS EXEC エージェントを使用すると、リモート アプリケーションは EXEC モード CLI コマンドを含むイベント メッセージを送信してシスコ デバイスで EXEC モード CLI コマンドを実行できます。すべての EXEC **show** コマンドがサポートされるわけではありません。

Cisco Networking Service 結果メッセージ

デバイスが部分設定を受信すると、設定の各行が受信された順に適用されます。設定のいずれかの行でシスコ パーサーのエラーがあった場合、その時点までの設定はすべてデバイスに適用されますが、エラー後の設定は適用されません。エラーが発生した場合、設定が正常に完了するまで **cnsconfigpartial** コマンドが再試行されます。プル モードでは、エラーの発生後コマンドは再試行されません。デフォルトでは、**no-persist** キーワードが設定されていなければ、NVRAM がアップデートされます。

部分設定が完了すると、Cisco Networking Service イベント バスにメッセージが発行されます。Cisco Networking Service イベント バスは、次のいずれかのステータス メッセージを表示します。

- cisco.mgmt.cns.config.complete：Cisco Networking Service 設定エージェントは正常に部分設定を適用しました。
- cisco.mgmt.cns.config.warning：Cisco Networking Service 設定エージェントは、部分設定を完全に適用しましたが、セマンティック エラーが発生する可能性があります。
- cisco.mgmt.cns.config.failure（CLI syntax）：Cisco Networking Service 設定エージェントは、コマンドライン インターフェイス（CLI）の構文エラーを発見したため、部分設定を適用できませんでした。
- cisco.mgmt.cns.config.failure（CLI semantic）：Cisco Networking Service 設定エージェントは、CLI セマンティック エラーを発見したため、部分設定を適用できませんでした。

CNS 拡張結果メッセージ機能により、上記の該当するメッセージに加えて、2 つめのメッセージがサブジェクト「cisco.cns.config.results」に送信されます。2 つめのメッセージには、送信された設定に関する全体的な情報と 1 行ごとの情報、および元のメッセージで要求されたアクションの結果が含まれます。要求されたアクションが設定の適用であった場合、結果メッセージ内の情報はセマンティクスに関するものになります。要求されたアクションが構文チェックだけであった場合、結果メッセージ内の情報は構文に関するものになります。

Cisco Networking Service メッセージ フォーマット

Service-Oriented Access Protocol（SOAP）メッセージ フォーマット

Service-Oriented Access Protocol（SOAP）プロトコルを使用すると、Cisco Networking Service メッセージのレイアウトを一貫性のある方法でフォーマットできます。SOAP は、非集中型の分散環境で構造化情報を交換するための軽量プロトコルです。Extensible Markup Language（XML）テクノロジーを使用して、さまざまな基本プロトコルで交換可能なメッセージ フォーマットを提供する、拡張性のあるメッセージング フレームワークを定義します。

SOAP メッセージ構造には、Cisco Networking Service 通知メッセージがユーザ クレデンシャルを認証できるセキュリティ ヘッダーがあります。

Cisco Networking Service メッセージは、要求、応答、および通知の 3 つのメッセージ タイプに分類されます。この 3 つのメッセージ タイプのフォーマットは次のように定義されます。

要求メッセージ

次に、シスコ デバイスへの Cisco Networking Service 要求メッセージのフォーマットを示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope">
  <SOAP:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="0">
      <wsse:usernameToken>
        <wsse:Username>john</wsse:Username>
        <wsse:Password>cisco</wsse:Password>
      </wsse:usernameToken>
    </wsse:Security>
    <cns:cnsHeader version="1.0" xmlns:cns="http://www.cisco.com/management/cns/envelope">
      <cns:Agent>CNS_CONFIG</cns:Agent>
      <cns:Request>
        <cns:correlationID>IDENTIFIER</cns:correlationID>
        <cns:ReplyTo>
          <cns:URL>http://10.1.36.9:80/cns/ResToServer</cns:URL>
        </cns:ReplyTo>
      </cns:Request>
      <cns:Time>2003-04-23T20:27:19.847Z</cns:Time>
    </cns:cnsHeader>
  </SOAP:Header>
  <SOAP:Body xmlns="http://www.cisco.com/management/cns/config">
    <config-event config-action="read" no-syntax-check="TRUE">
      <config-data>
        <config-id>AAA</config-id>
        <cli>access-list 1 permit any</cli>
      </config-data>
    </config-event>
  </SOAP:Body>
</SOAP:Envelope>
```

（注） ReplyTo フィールドは任意です。ReplyTo フィールドがない場合は、要求に対する応答は要求の発信元である宛先に送信されます。このメッセージの本体部分にはペイロードが含まれており、Agent フィールドに記述されている Cisco Networking Service エージェントによって処理されます。

応答メッセージ

次に、要求に対する応答としてのシスコ デバイスからの Cisco Networking Service 応答メッセージのフォーマットを示します。

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username infysj-7204-8 /wsse:Username
wsse:Password NTM3NTg2NzIzOTg2MTk2MjgzNQ==/wsse:Password
/wsse:UsernameToken /wsse:Security
CNS:cnsHeader Version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG /CNS:Agent
CNS:Response
CNS:correlationID IDENTIFIER /CNS:correlationID
/CNS:Response
CNS:Time 2005-06-23T16:27:36.185Z /CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config"
```

```
config-success config-id AAA /config-id /config-success
/SOAP:Body
/SOAP:Envelope
```

（注） CorrelationId の値は、対応する要求メッセージからエコーされます。

このメッセージの本体部分には、要求に対するシスコ デバイスからの応答が含まれます。要求が正常に処理された場合、本体部分には要求を処理したエージェントによって挿入された応答の値が含まれます。要求が正常に処理できなかった場合、本体部分にはエラー応答が含まれます。

通知メッセージ

次に、シスコ デバイスから送信される Cisco Networking Service 通知メッセージのフォーマットを示します。

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username dvlpr-7200-2 /wsse:Username
wsse:Password /wsse:Password
/wsse:UsernameToken
/wsse:Security
CNS:cnsHeader version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG_CHANGE/CNS:Agent
CNS:Notify /CNS:Notify
CNS:Time 2006-01-09T18:57:08.441Z/CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config-change"
configChanged version="1.1" sessionData="complete"
sequence lastReset="2005-12-11T20:18:39.673Z" 7 /sequence
changeInfo
user/user
async port con_0 /port /async
when
absoluteTime 2006-01-09T18:57:07.973Z /absoluteTime
/when
/changeInfo
changeData
changeItem
context /context
enteredCommand
cli access-list 2 permit any /cli
/enteredCommand
oldConfigState
cli access-list 1 permit any /cli
/oldConfigState
newConfigState
cli access-list 1 permit any /cli
cli access-list 2 permit any /cli
/newConfigState
/changeItem
/changeData
/configChanged
/SOAP:Body
/SOAP:Envelope
```

通知メッセージは、設定変更が行われたときに対応する要求メッセージなしでシスコ デバイスから送信されます。メッセージの本体には通知のペイロードが含まれ、エラー情報が含まれる場合もあります。シスコ デバイスに送信された要求メッセージが XML 解析に失敗し、CorrelationId フィールドを解析できない場合、エラー応答の代わりにエラー通知メッセージが送信されます。

エラー レポート

エラーは、応答メッセージまたは通知メッセージの本体の SOAP Fault エレメントで報告されます。次に、レポート エラーのフォーマットを示します。

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username dvlpr-7200-2 /wsse:Username
wsse:Password /wsse:Password
/wsse:UsernameToken
/wsse:Security
CNS:cnsHeader version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG /CNS:Agent
CNS:Response
CNS:correlationID SOAP_IDENTIFIER /CNS:correlationID
/CNS:Response
CNS:Time 2006-01-09T19:10:10.009Z /CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config"
SOAP:Detail
config-failure
config-id AAA /config-id
error-info
line-number 1 /line-number
error-message CNS_INVALID_CLI_CMD /error-message
/error-info
/config-failure
/SOAP:Detail
/SOAP:Fault
/SOAP:Body
/SOAP:Envelope
```

Cisco Networking Service ID

Cisco Networking Service ID は、特定の Cisco Networking Service エージェントだけで使用されるテキスト文字列です。Cisco Networking Service ID は、Cisco Networking Service エージェントが通信するサーバ アプリケーションに対して自身を識別するために使用されます。たとえば、Cisco Networking Service 設定エージェントには、ネットワーキング デバイスとコンフィギュレーション サーバとの間で通信する場合の設定 ID が含まれます。コンフィギュレーション サーバは、Cisco Networking Service 設定 ID をキーとして使用して、設定プルの発信元であるデバイス用の Cisco CLI 設定を含む属性を見つけます。

ネットワーク管理者は、ルーティング デバイスで定義されている Cisco Networking Service エージェント ID と、ルーティング デバイス用の設定に対応するディレクトリ属性に含まれる Cisco Networking Service エージェント ID が一致していることを確認する必要があります。ルーティング デバイスでは、Cisco Networking Service エージェント ID のデフォルト値は常にホスト名に設定されます。ホスト名が変更されると、Cisco Networking Service エージェント ID も変更されます。Cisco Networking Service エージェント ID が CLI を使用して設定されている場合、変更が行われるとメッセージが syslog に送信されるか、またはイベント メッセージが送信されます。

Cisco Networking Service エージェント ID はセキュリティ問題に対処しません。

Cisco Networking Service パスワード

Cisco Networking Service パスワードは、Cisco Networking Service デバイスの認証に使用されます。初めてデバイスを配置したとき、Cisco Networking Service パスワードを設定する必要があります。Cisco Networking Service パスワードは、設定エンジン（CE）に設定されているブートストラップパスワードと同じにする必要があります。デバイスおよびCEブートストラップの両方のパスワードにデフォルト設定を使用している場合、新しく配置されたデバイスは CE に接続できます。接続されると、CE は Cisco Networking Service パスワードを管理します。ネットワーク管理者は、Cisco Networking Service パスワードが変更されていないことを確認する必要があります。Cisco Networking Service パスワードが変更されると、CE への接続は失われます。

Cisco Networking Service ゼロ タッチ

Cisco Networking Service ゼロ タッチ機能は、デバイスが Cisco Networking Service 設定エンジンに接続し、全設定を自動的に取得するゼロ タッチ展開ソリューションを提供します。この機能は、サービスに加入しているサービスプロバイダーのエンドユーザすべてに共通する単一の汎用ブートストラップ設定ファイルによって可能になります。Cisco Networking Service フレームワークでは、顧客は、インターフェイスタイプ、回線タイプ、コントローラタイプ（該当する場合）などのデバイス固有またはネットワーク固有の情報を使用せずに、この汎用ブートストラップ設定を作成できます。

Cisco Networking Service 接続機能は、Cisco Networking Service 接続テンプレート セットを使用して設定されます。Cisco Networking Service 接続プロファイルは、Cisco Networking Service 設定エンジンに接続し、加入者宅内機器（CPE）デバイスに Cisco Networking Service 接続テンプレートを実装するために作成します。Cisco Networking Service 接続変数は、Cisco Networking Service 接続テンプレート設定内のプレースホルダーとして使用できます。アクティブ DLCI などのこの変数は、Cisco Networking Service 接続テンプレートがデバイスのパーサーに送信される前に、実際の値と置き換えられます。

ゼロ タッチ機能を使用するには、初期化されるデバイスに汎用ブートストラップ設定が必要です。この設定には、Cisco Networking Service 接続テンプレート、Cisco Networking Service 接続プロファイル、および **cnsconfiginitial** コマンドが含まれます。このコマンドは、Cisco Networking Service 接続機能を起動します。

Cisco Networking Service 接続機能は、デバイスのインターフェイス、回線、および使用可能なコントローラを介して複数の ping 繰り返しを実行します。繰り返しごとに、Cisco Networking Service 接続機能は Cisco Networking Service 設定エンジンに ping を試みます。ping が正常に実行されると、Cisco Networking Service 設定エンジンから関連する設定情報をダウンロードできます。Cisco Networking Service 設定エンジンに接続できない場合、Cisco Networking Service 接続機能は選択されたインターフェイスに適用された設定を削除し、Cisco Networking Service 接続プロセスが Cisco Networking Service 接続プロファイルで指定された次に使用可能なインターフェイスで再開されます。

Cisco Networking Service ゼロ タッチ機能には、次の利点があります。

- Cisco Networking Service コマンドの一貫性を確保できます。

- チャネル サービス ユニット（E1 または T1 コントローラ）を使用できます。

Cisco Networking Service の設定方法

Cisco Networking Service デバイスの配置

差分（部分）設定を使用すると、リモート デバイスを初期設定後、差分的に設定できます。この設定は、Cisco Networking Service 設定エンジンを介して手動で行う必要があります。レジストラを使用すると、設定テンプレートの変更、パラメータの編集、およびデバイスへの新規設定サブミットを、ソフトウェアやハードウェアを再起動せずに実行できます。

はじめる前に

Cisco Networking Service の初期設定を手動でインストールするには、次の作業を実行します。

リモート デバイスは、ブートストラップ設定が適用された状態で出荷されます。初回電源投入時に、デバイスは自動的に Cisco Networking Service 設定エンジンから全初期設定をプルします（手動実行するようにアレンジすることもできます）。初期設定後、同期を取るために定期的差分（部分）設定をアレンジすることもできます。

Cisco CNS 設定エンジンを使用して CNS 初期設定を自動的にインストールする方法の詳細については、『*Cisco CNS Configuration Engine Administrator's Guide*』（http://www.cisco.com/en/US/docs/net_mgmt/configuration_engine/1.3/administration/guide/ag13.html）を参照してください。

Cisco Networking Service の初期設定

リモートデバイスがネットワーク上で初期化されると、リモートデバイスの初期設定が自動的に行われます。任意で、この設定を手動で実行することもできます。

Cisco Networking Service は、一意の IP アドレスまたはホスト名をリモート デバイスに割り当てます。IP アドレスの解決後（Serial Line Address Resolution Protocol（SLARP）、ATM Inverse ARP（ATM InARP）、または PPP プロトコルを使用）、システムは任意でドメイン ネーム システム（DNS）リバース ルックアップを使用して、デバイスにホスト名を割り当て、Cisco Networking Service エージェントを起動し、Cisco Networking Service 設定エンジンから初期設定をダウンロードします。

差分設定

差分設定を設定するには、Cisco Networking Service が稼働しており、必要な Cisco Networking Service エージェントが設定されている必要があります。

手順の概要

1. **enable**
2. **configureterminal**
3. **cnstemplateconnect***name*
4. **cli***config-text*
5. ステップ 4 を繰り返して、必要な CLI コマンドをすべて追加します。
6. **exit**
7. **cnsconnect***name* [**retry-interval***interval-seconds*] [**retries***number-retries*] [**timeout***timeout-seconds*] [**sleep***sleep-seconds*]
8. 次のいずれかを実行します。
   - **discover** {**line***line-type* | **controller***controller-type* | **interface** [*interface-type*]}
   - **template***name*
9. **exit**
10. **cnsconfiginitial** {*host-name*|*ip-address*} [**encrypt**] [*port-number*] [**page***page*] [**syntax-check**] [**no-persist**] [**source***interfacename*] [**status***url*] [**event**] [**inventory**]
11. **exit**

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cnstemplateconnect***name* 例： `Device(config)# cns template connect template 1`	Cisco Networking Service テンプレート接続コンフィギュレーション モードを開始し、Cisco Networking Service 接続テンプレートの名前を定義します。
ステップ 4	**cli***config-text* 例： `Device(config-templ-conn)# cli encapsulation ppp`	インターフェイスを設定するコマンドを指定します。

	コマンドまたはアクション	目的
ステップ 5	ステップ 4 を繰り返して、必要な CLI コマンドをすべて追加します。 例： `Device(config-templ-conn)# cli ip directed-broadcast`	ステップ 4 を繰り返して、インターフェイスまたはモデム回線を設定するための他の CLI コマンドを追加します。
ステップ 6	**exit** 例： `Device(config-templ-conn)# exit`	Cisco Networking Service テンプレート接続コンフィギュレーション モードを終了し、Cisco Networking Service 接続テンプレートの設定を完了します。 （注） **exit** コマンドの入力は必須です。誤って **cli** コマンドを付けずにコマンドを実行することがないよう、この要件が規定されています。
ステップ 7	**cnsconnect***name* [**retry-interval***interval-seconds*] [**retries***number-retries*] [**timeout***timeout-seconds*] [**sleep***sleep-seconds*] 例： `Device(config)# cns connect profile-1 retry-interval 15 timeout 90`	Cisco Networking Service 接続コンフィギュレーション モードを開始し、Cisco Networking Service 設定エンジンに接続するための Cisco Networking Service 接続プロファイルのパラメータを定義します。

ステップ 8	次のいずれかを実行します。
	
	• **discover** {**line***line-type* | **controller***controller-type* | **interface** [*interface-type*]}
	
	• **template***name*
	
	例：
	
	`Device(config-cns-conn)# discover interface serial`
	
	例：
	
	`Device(config-cns-conn)# template template-1`
		（任意）汎用ブートストラップ設定を設定します。
		
		• **discover**：Cisco Networking Service 設定エンジンへの接続用に、Cisco Networking Service 接続プロファイル内のインターフェイス パラメータを定義します。
		
		または
		
		• **template**：デバイスの設定に適用する、Cisco Networking Service 接続プロファイル内の Cisco Networking Service 接続テンプレートのリストを指定します。

ステップ 9	**exit** 例： `Device(config-cns-conn)# exit`	Cisco Networking Service 接続コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 10	**cnsconfiginitial** {*host-name*|*ip-address*} [**encrypt**] [*port-number*] [**page***page*] [**syntax-check**] [**no-persist**] [**source***interfacename*] [**status***url*] [**event**] [**inventory**] 例： `Device(config)# cns config initial 10.1.1.1 no-persist`	Cisco Networking Service 設定エージェントを起動し、Cisco Networking Service 設定エンジンに接続し、初期設定を開始します。このコマンドを使用できるのは、初回システム起動の前に限られます。 （注） Secure Socket Layer（SSL）をサポートするイメージに限り、オプションの **encrypt** キーワードを使用できます。 注意 NVRAM に新規設定を書き込むときに **no-persist** キーワードを省略すると、元のブートストラップ設定は上書きされます。
ステップ 11	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

高度な Cisco Networking Service 機能の設定

より高度な Cisco Networking Service 機能を設定するには、次の作業を実行します。Cisco Networking Service エージェントが動作していると、その他の機能を設定できます。Cisco Networking Service インベントリ エージェントをイネーブル化（つまり、デバイスのラインカードとモジュールのインベントリを Cisco Networking Service 設定エンジンに送信）して、Cisco Networking Service インベントリ モードを開始できます。

その他の高度な機能により、ソフトウェア開発キット（SDK）を使用して Cisco Networking Service 通知の送信方法や MIB 情報へのアクセス方法を指定できます。非粒状（SNMP）カプセル化と粒状（XML）カプセル化の、2 つのカプセル化方式を使用できます。

手順の概要

1. **enable**
2. **configureterminal**
3. **cnsmib-accessencapsulation** {**snmp**|**xml**[**size***bytes*]}
4. **cnsnotificationsencapsulation** {**snmp**|**xml**}
5. **cnsinventory**
6. **transportevent**
7. **exit**

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cnsmib-accessencapsulation** {**snmp**|**xml**[**size***bytes*]} 例： `Device(config)# cns mib-access encapsulation snmp`	（任意）MIB 情報へのアクセスに使用するカプセル化のタイプを指定します。 • MIB 情報へのアクセスに非粒状カプセル化を使用するように指定するには、**snmp** キーワードを使用します。 • MIB 情報へのアクセスに粒状カプセル化を使用するように指定するには、**xml** キーワードを使用します。オプションの **size** キーワードは、応答イベントの最大サイズ（バイト）を指定します。デフォルトのバイト値は 3072 です。
ステップ 4	**cnsnotificationsencapsulation** {**snmp**|**xml**} 例： `Device(config)# cns notifications encapsulation xml`	（任意）Cisco Networking Service 通知の送信時に使用するカプセル化のタイプを指定します。 • Cisco Networking Service 通知の送信時に非粒状カプセル化を使用するように指定するには、**snmp** キーワードを使用します。 • Cisco Networking Service 通知の送信時に粒状カプセル化を使用するように指定するには、**xml** キーワードを使用します。
ステップ 5	**cnsinventory** 例： `Device(config)# cns inventory`	Cisco Networking Service インベントリ エージェントをイネーブルにし、Cisco Networking Service インベントリ モードを開始します。 • デバイスのラインカードおよびモジュールのインベントリが、Cisco Networking Service 設定エンジンに送信されます。

	コマンドまたはアクション	目的
ステップ 6	**transportevent** 例： `Device(cns-inv)# transport event`	インベントリ要求が各 Cisco Networking Service インベントリ エージェント メッセージで送信されるように指定します。
ステップ 7	**exit** 例： `Device(cns-inv)# exit`	Cisco Networking Service インベントリ モードを終了し、グローバル コンフィギュレーション モードに戻ります。 • このコマンドを繰り返して、特権 EXEC モードに戻ります。

Cisco Networking Service エージェントのトラブルシューティング

ここでは、Cisco Networking Service エージェントの問題をトラブルシューティングする方法について説明します。

Cisco Networking Service イメージ エージェント用に作成された **show** コマンドは、デバイスが正常にリロードされた後にゼロにリセットされる情報を表示します。イメージ配信プロセスの設定によっては、新しいイメージがすぐにリロードされない場合があります。すぐにリロードされない場合やリロードに失敗した場合は、Cisco Networking Service イメージ エージェントの **show** コマンドを使用して、イメージ エージェントが HTTP でイメージ配信サーバに接続されているかどうか、またはイメージ エージェントが Cisco Networking Service イベント バスを介してアプリケーションからイベントを受信しているかどうかを確認します。

手順の概要

1. **enable**
2. **showcnsimagestatus**
3. **clearcnsimagestatus**
4. **showcnsimageconnections**
5. **showcnsimageinventory**
6. **debugcnsimage**[**agent**| **all**| **connection**| **error**]
7. **showcnseventconnections**
8. **showcnseventsubject** [*name*]

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 モードなど、高位の権限レベルをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**showcnsimagestatus** 例： `Device# show cns image status`	（任意）Cisco Networking Service イメージエージェントのステータスに関する情報を表示します。
ステップ 3	**clearcnsimagestatus** 例： `Device# clear cns image status`	（任意）Cisco Networking Service イメージエージェントのステータスの統計情報をクリアします。
ステップ 4	**showcnsimageconnections** 例： `Device# show cns image connections`	（任意）Cisco Networking Service イメージ管理サーバの HTTP または HTTPS 接続に関する情報を表示します。
ステップ 5	**showcnsimageinventory** 例： `Device# show cns image inventory`	（任意）Cisco Networking Service イメージ エージェントのインベントリ情報を表示します。 • このコマンドは、イメージエージェントのインベントリ要求メッセージに対する応答で送信される XML のダンプを表示します。XML 出力は、アプリケーションによって要求される情報を確認するために使用できます。
ステップ 6	**debugcnsimage**[**agent**| **all**| **connection**| **error**] 例： `Device# debug cns image all`	（任意）Cisco Networking Service イメージ エージェント サービスのデバッグ メッセージを表示します。
ステップ 7	**showcnseventconnections** 例： `Device# show cns event connections`	（任意）Cisco Networking Service イベント エージェントの接続のステータスを表示し（ゲートウェイに接続されているか、接続済み、またはアクティブなど）、このイベントエージェントによって使用されるゲートウェイとその IP アドレスとポート番号を表示します。

	コマンドまたはアクション	目的
ステップ 8	**showcnseventsubject** [*name*] 例： `Device# show cns event subject subject1`	（任意）アプリケーションによって加入される Cisco Networking Service イベント エージェントのサブジェクトのリストを表示します。

例

次に、**showcnsimagestatus** 特権 EXEC コマンドを使用して Cisco Networking Service イメージ エージェントのステータス情報を表示する例を示します。

```
Device# show cns image status
Last upgrade started at 11:45:02.000 UTC Mon May 6 2003
Last upgrade ended at 11:56:04.000 UTC Mon May 6 2003 status SUCCESS
Last successful upgrade ended at 11:56:04.000 UTC Mon May 6 2003
Last failed upgrade ended at 06:32:15.000 UTC Wed Apr 16 2003
Number of failed upgrades: 2
Number of successful upgrades: 6
 messages received: 12
 receive errors: 5
Transmit Status
  TX Attempts:4
    Successes:3         Failures 2
```

次に、**showcnsimageconnections** 特権 EXEC コマンドを使用して Cisco Networking Service イメージ管理 HTTP 接続のステータスに関する情報を表示する例を示します。

```
show cns image connections
CNS Image Agent:  HTTP connections
Connection attempts 1
never connected:0   Abrupt disconnect:0
Last successful connection at 11:45:02.000 UTC Mon May 6 2003
```

次に、**showcnsimageinventory** 特権 EXEC コマンドを使用して Cisco Networking Service イメージ エージェント インベントリに関する情報を表示する例を示します。

```
show cns image inventory
Inventory Report
imageInventoryReport deviceName imageID Router /imageID hostName Router /ho
IOS (tm) C2600 Software (C2600-I-M), Experimental Version 12.3(20030414:081500)]
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Mon 14-Apr-03 02:03 by engineer /versionString imageFile tftp://10.25.2.1.
```

次に、**debugcnsimage** 特権 EXEC コマンドを使用してすべての Cisco Networking Service イメージ エージェント サービスのデバッグ メッセージを表示する例を示します。この例の Cisco Networking Service イメージ エージェントは HTTP でイメージ サーバに接続しています。接続後、イメージ サーバはシスコ デバイスのインベントリを要求します。

```
Device# debug cns image all
All cns image debug flags are on
Device# cns image retrieve

May  7 06:11:42.175: CNS Image Agent: set EXEC lock
May  7 06:11:42.175: CNS Image Agent: received message from EXEC
May  7 06:11:42.175: CNS Image Agent: set session lock 1
```

```
May  7 06:11:42.175: CNS Image Agent: attempting to send to
destination(http://10.1.36.8:8080/imgsrv/xgate):
?xml version="1.0" encoding="UTF-8"? cnsMessageversion="1.0" senderCredentials userName
dvlpr-7200-6 /userName /senderCredentials
messageID dvlpr-7200-6_2 /messageID sessionControl imageSessionStart version="1.0"
initiatorInfotrigger EXEC/trigger initiatorCredentials userName dvlpr-7200-6/userName
/initiatorCredentials /initiatorInfo /imageSessionStart /sessionControl /cnsMessage
May  7 06:11:42.175: CNS Image Agent: clear EXEC lock
May  7 06:11:42.175: CNS Image Agent: HTTP message sent url:http://10.1.36.8:8080/imgsrv/xgate
May  7 06:11:42.191: CNS Image Agent: response data alloc 4096 bytes
May  7 06:11:42.191: CNS Image Agent: HTTP req data free
May  7 06:11:42.191: CNS Image Agent: response data freed
May  7 06:11:42.191: CNS Image Agent: receive message
?xml version="1.0" encoding="UTF-8"?
cnsMessage version="1.0"
senderCredentials
userName myImageServer.cisco.com/userName
passWord R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b/passWord
/senderCredentials
messageID dvlpr-c2600-2-476456/messageID
request
replyTo
serverReply http://10.1.36.8:8080/imgsrv/xgate /serverReply
/replyTo
imageInventory
inventoryItemList
all/
/inventoryItemList
/imageInventory
/request
/cnsMessage
```

次に、プライマリ ゲートウェイおよびバックアップ ゲートウェイの IP アドレスとポート番号の例を示します。

```
Device# show cns event connections
The currently configured primary event gateway:
 hostname is 10.1.1.1.
 port number is 11011.
Event-Id is Internal test1
Keepalive setting:
 none.
Connection status:
 Connection Established.
The currently configured backup event gateway:
 none.
The currently connected event gateway:
 hostname is 10.1.1.1.
 port number is 11011.
```

次に、アプリケーションによって加入される Cisco Networking Service イベント エージェントのサブジェクトのリストを表示する例を示します。

```
Device# show cns event subject
The list of subjects subscribed by applications.
 cisco.cns.mibaccess:request
 cisco.cns.config.load
 cisco.cns.config.reboot
 cisco.cns.exec.cmd
```

Cisco Networking Service の設定例

例：Cisco Networking Service デバイスの配置

次に、リモート デバイス上の初期設定例を示します。リモート デバイスのホスト名は一意の ID です。Cisco Networking Service 設定エンジンの IP アドレスは 172.28.129.22 です。

```
cns template connect template1
 cli ip address negotiated
 cli encapsulation ppp
 cli ip directed-broadcast
 cli no keepalive
 cli no shutdown
 exit
cns connect host1 retry-interval 30 retries 3
exit
 hostname RemoteRouter
 ip route 172.28.129.22 255.255.255.0 10.11.11.1
 cns id Ethernet 0 ipaddress
 cns config initial 10.1.1.1 no-persist
 exit
```

例：Cisco Networking Service ゼロ タッチ ソリューションの使用

シリアル インターフェイス上の PPP の設定

次に、シリアル インターフェイス上で PPP を設定するためのブートストラップ設定例を示します。

```
cns template connect ppp-serial
cli ip address negotiated
cli encapsulation ppp
cli ip directed-broadcast
cli no keepalive
exit
cns template connect ip-route
cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect serial-ppp ping-interval 1 retries 1
discover interface serial
template ppp-serial
template ip-route
exit
hostname 26ML
cns config initial 10.1.1.1 no-persist inventory
```

非同期インターフェイス上の PPP の設定

次に、非同期インターフェイスに PPP を設定するためのブートストラップ設定例を示します。

```
cns template connect async
cli modem InOut
 .
 .
 .
exit
```

```
cns template connect async-interface
cli encapsulation ppp
cli ip unnumbered FastEthernet0/0
cli dialer rotary-group 0
exit
cns template connect ip-route
cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect async
discover line Async
template async
discover interface
template async-interface
template ip-route
exit
hostname async-example
cns config initial 10.1.1.1 no-persist inventory
```

シリアル インターフェイス上の HDLC の設定

次に、シリアル インターフェイスにハイレベル データ リンク制御（HDLC）を設定するためのブートストラップ設定例を示します。

```
cns template connect hdlc-serial
cli ip address slarp retry 1
exit
cns template connect ip-route
cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect hdlc-serial ping-interval 1 retries 1
discover interface serial
template hdlc-serial
template ip-route
exit
hostname host1
cns config initial 10.1.1.1 no-persist inventory
```

集約デバイス インターフェイスの設定

次に、標準シリアルインターフェイスおよび、集約デバイス（DCE とも呼ばれる）のコントローラのシリアル インターフェイスを設定する例を示します。接続を確立するために、集約デバイスにはポイントツーポイント サブインターフェイスを設定する必要があります。

標準シリアル インターフェイス

```
interface Serial0/1
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
exit
interface Serial0/1.1 point-to-point
 10.0.0.0 255.255.255.0
 frame-relay interface-dlci 8
```

コントローラのシリアル インターフェイス

```
controller T1 0
 framing sf
 linecode ami
 channel-group 0 timeslots 1-24
exit
interface Serial0:0
 no ip address
```

```
 encapsulation frame-relay
 frame-relay intf-type dce
exit
interface Serial0:0.1 point-to-point
 ip address ip-address mask
 frame-relay interface-dlci dlci
```

IP over Frame Relay の設定

次に、CPE デバイスに IP over Frame Relay を設定するためのブートストラップ設定例を示します。

```
cns template connect setup-frame
 cli encapsulation frame-relay
 exit
cns template connect ip-over-frame
 cli frame-relay interface-dlci ${dlci}
 cli ip address dynamic
 exit
cns template connect ip-route
 cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
 exit
cns connect ip-over-frame
 discover interface Serial
 template setup-frame
 discover dlci
 template ip-over-frame
 template ip-route
exit
cns config initial 10.1.1.1
```

T1 を介した IP over Frame Relay の設定

次に、CPE デバイスに、T1 を介した IP over Frame Relay を設定するためのブートストラップ設定例を示します。

```
cns template connect setup-frame
 cli encapsulation frame-relay
 exit
cns template connect ip-over-frame
 cli frame-relay interface-dlci ${dlci}
 cli ip address dynamic
 exit
cns template connect ip-route
 cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
 exit
cns template connect t1-controller
 cli framing esf
 cli linecode b8zs
 cli channel-group 0 timeslots 1-24 speed 56
 exit
cns connect ip-over-frame-over-t1
 discover controller T1
 template t1-controller
 discover interface
 template setup-frame
 discover dlci
 template ip-over-frame
 template ip-route
exit
cns config initial 10.1.1.1
```

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
Cisco Networking Service コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例。	『Cisco IOS Cisco Networking Services Command Reference』
Cisco Networking Service 設定エンジン（CE）	『Cisco CNS Configuration Engine Administrator Guide, 1.3』

標準および RFC

標準/RFC	タイトル
この機能によりサポートされる新規または変更された標準/RFC はありません。またこの機能による既存の標準/RFC のサポートに変更はありません。	--

MIB

MIB	MIB のリンク
この機能によってサポートされる新しい MIB または変更された MIB はありません。またこの機能による既存 MIB のサポートに変更はありません。	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

Cisco Networking Service 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 *2*：*Cisco Networking Service* の機能情報

機能名	リリース	機能情報
Cisco Networking Service	Cisco IOS XE Release 2.1 12.2(25)S 12.2(33)SRA 12.2(33)SB 12.2(33)SXI	Cisco Networking Service 機能は、リモート イベント駆動型の Cisco IOS ネットワーキングデバイスの設定、および一部の CLI コマンドのリモート実行を可能にするサービスの集合です。 この機能により、次のコマンドが導入または変更されました。**clearcnsconfigstats**、**clearcnscounters**、**clearcnseventstats**、**cli(cns)**、**cnsconfigcancel**、**cnsconfiginitial**、**cnsconfignotify**、**cnsconfigpartial**、**cnsconfigretrieve**、**cnsconnect**、**cnsevent**、**cnsexec**、**cnsid**、**cnstemplateconnect**、**cnstrusted-server**、**debugcnsconfig**、**debugcnsexec**、**debugcnsxml-parser**、**loggingcns-events**、**showcnsconfigstats**、**showcnseventconnections**、**showcnseventstats**、**showcnseventsubject**

第 3 章

CNS 設定エージェント

- 機能情報の確認，27 ページ
- CNS 設定エージェントについて，27 ページ
- CNS 設定エージェントの設定方法，29 ページ
- CNS 設定エージェントの設定例，31 ページ
- その他の参考資料，33 ページ
- CNS 設定エージェントの機能情報，34 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

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。

CNS 設定エージェントについて

Cisco Networking Service 設定エージェント

Cisco Networking Service 設定エージェントは、シスコ デバイスの初期設定および以後の部分設定に含まれています。Cisco Networking Service 設定エージェントをアクティブにするには、**cnsconfig** CLI コマンドのいずれかを入力します。

Cisco Networking Service の初期設定

ルーティング デバイスは、初めて起動すると、標準 CLI コマンドである **cns config initial** コマンドを使用して TCP 接続を確立することによって Cisco Networking Service 設定エージェントのコンフィギュレーション サーバ コンポーネントに接続します。デバイスは要求を発行し、コンフィギュレーション サーバに一意の設定 ID を提供してデバイス自身を識別します。

Cisco Networking Service Web サーバはコンフィギュレーション ファイルの要求を受信すると、Java サーブレットを呼び出し、該当する埋め込みコードを実行します。この埋め込みコードの指示によって、Cisco Networking Service Web サーバはディレクトリ サーバおよびファイル システムにアクセスし、このデバイス（設定 ID）用のコンフィギュレーション リファレンスとテンプレートを読み取ります。設定エージェントは、テンプレート内に指定されているすべてのパラメータ値に、このデバイスの有効な値を代入して、コンフィギュレーション ファイルのインスタンスを作成します。コンフィギュレーションサーバは、設定ファイルをルーティングデバイスに転送するために Cisco Networking Service Web サーバに転送します。

Cisco Networking Service 設定エージェントは、Cisco Networking Service Web サーバから設定ファイルを受信して、XML 解析を実行し、構文をチェックし（任意）、設定ファイルをロードします。ルーティングデバイスは設定ロードのステータスを、ネットワーク モニタリングまたはワークフロー アプリケーションがサブスクライブできるイベントとして報告します。

Cisco Networking Service 設定エンジンを使用して Cisco Networking Service の初期設定を自動的にインストールする方法の詳細については、『*Cisco Networking Services Configuration Engine Administrator's Guide*』（http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cns/ce/rel13/ag13/index.htm）を参照してください。

Cisco Networking Service の差分設定

ネットワークが稼働すると、Cisco Networking Service 設定エージェントを使用して新しいサービスを追加できます。差分（部分）設定はルーティング デバイスに送信できます。実際の設定を、イベントペイロードとしてイベントゲートウェイを介して（プッシュ処理）、またはデバイスにプル オペレーションを開始させる信号イベントとして送信できます。

ルーティング デバイスは、設定を適用する前にその構文をチェックできます。構文が正しい場合は、ルーティングデバイスは差分設定を適用し、コンフィギュレーションサーバに成功を信号で伝えるイベントを発行します。ルーティング デバイスが差分設定を適用しなかった場合、エラーを示すイベントを発行します。

ルーティング デバイスが差分設定を適用した後、その設定を NVRAM に書き込むことも、書き込み指示の信号が来るまで待機することもできます。

コンフィギュレーションの同期

ルーティング デバイスは設定を受信しても、その設定の適用を書き込み信号イベントの受信時まで据え置くことができます。Cisco Networking Service 設定エージェント機能を使用すると、デバイスの設定を他の依存ネットワーク アクティビティと同期化できます。

CNS 設定エージェントの設定方法

Cisco Networking Service イベント エージェントおよび EXEC エージェントの設定

手順の概要

1. **enable**
2. **configureterminal**
3. **cnsconfigpartial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**source***interfacename*] [**inventory**]
4. **loggingcns-events** [*severity-level*]
5. **cns exec** [**encrypt**] [*port-number*] [**source** {*ip-address* | *interface-type-number*}]
6. **cnsevent**{*hostname*|*ip-address*} [**encrypt**] [*port-number*] [**backup**] [**failover-time***seconds*] [**keepalive***secondsretry-count*] [**source***ip-address* | *interface-name*][**clock-timeout***time*] [**reconnect-time***time*]
7. **exit**

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cnsconfigpartial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**source***interfacename*] [**inventory**] 例： `Device(config)# cns config partial 172.28.129.22 80`	（任意）Cisco Networking Service 設定エージェントを起動します。これにより、シスコ クライアントに Cisco Networking Service 設定サービスが提供され、差分（部分）設定が開始されます。 • コンフィギュレーション サーバのポート番号を指定するには、オプションの *port-number* 引数を使用します。デフォルトは 80 です。 • Cisco Networking Service 設定エージェントの通信の送信元として IP アドレスを使用するように指定するには、オプションの **source** キーワードと *ip-address* 引数を使用します。

	コマンドまたはアクション	目的
		• HTTP 要求の一部として Cisco Networking Service 設定エンジンにデバイスのラインカードとモジュールのインベントリを送信するには、オプションの **inventory** キーワードを使用します。 （注） SSL をサポートするイメージに限り、オプションの **encrypt** キーワードを使用できます。
ステップ 4	**loggingcns-events** [*severity-level*] 例： `Device(config)# logging cns-events 2`	（任意）XML フォーマットのシステム イベント メッセージ ロギングを Cisco Networking Service イベント バスを介して送信できます。 • メッセージをログに記録する重大度の番号または名前を指定するには、オプションの *severity-level* 引数を使用します。デフォルトはレベル 7（デバッグ）です。
ステップ 5	**cns exec** [**encrypt**] [*port-number*] [**source** {*ip-address* | *interface-type-number*}] 例： `Device(config)# cns exec source 172.17.2.2`	（任意）Cisco Networking Service EXEC エージェントをイネーブル化して設定します。これにより、シスコ クライアントに Cisco Networking Service EXEC サービスが提供されます。 • EXEC サーバのポート番号を指定するには、オプションの *port-number* 引数を使用します。デフォルトは 80 です。 • Cisco Networking Service EXEC エージェントの通信の送信元として IP アドレスを使用するように指定するには、オプションの **source** キーワードと *ip-address*/*interface-type number* 引数を使用します。 （注） SSL をサポートするイメージに限り、オプションの **encrypt** キーワードを使用できます。
ステップ 6	**cnsevent**{*hostname*|*ip-address*} [**encrypt**] [*port-number*] [**backup**] [**failover-time***seconds*] [**keepalive***secondsretry-count*] [**source***ip-address* | *interface-name*][**clock-timeout***time*] [**reconnect-time***time*] 例： `Device(config)# cns event 172.28.129.22 source 172.22.2.1`	（任意）Cisco Networking Service イベント ゲートウェイを設定します。これにより、シスコ クライアントに Cisco Networking Service イベント サービスが提供されます。 • SSL をサポートするイメージに限り、オプションの **encrypt** キーワードを使用できます。 • イベント サーバのポート番号を指定するには、オプションの *port-number* 引数を使用します。デフォルトは、11011（暗号化なし）および 11012（暗号化あり）です。 • このゲートウェイがバックアップ ゲートウェイであることを示すには、オプションの **backup** キーワードを使用します。バックアップ ゲートウェイを設定する前に、プライマリ ゲートウェイが設定されていることを確認します。 • バックアップ ゲートウェイへのルートが確立された後、プライマリ ゲートウェイのルートを待機する時間間隔（秒）を指定するには、オプションの **failover-time** キーワードと *seconds* 引数を使用します。

	コマンドまたはアクション	目的
		• キープアライブ タイムアウト（秒）および再試行回数を指定するには、オプションの **keepalive** キーワードと *seconds* および *retry-count* 引数を使用します。 • Cisco Networking Service イベント エージェントの通信の送信元として IP アドレスを使用するように指定するには、オプションの **source** キーワードと *ip-address*/*interface-name* 引数を使用します。 • 正確なクロックを必要とする転送（SSL など）にクロックが設定されるまで Cisco Networking Service イベント エージェントが待機する最大時間（分）を指定するには、オプションの **clock-timeout** キーワードを使用します。 • 最大再試行タイムアウトの設定可能な上限を指定するには、オプションの **reconnect-time** キーワードを使用します。 （注） **cnsevent** コマンドを入力するまで、Cisco Networking Service イベント バスへの転送接続が確立されないため、その他の Cisco Networking Service エージェントは稼働しません。
ステップ 7	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

トラブルシューティングのヒント

- Cisco Networking Service イベント エージェントが Cisco Networking Service イベント ゲートウェイに接続されていることを確認するには、**showcnseventconnections** コマンドを使用します。
- イメージ エージェントのサブジェクト名が登録されていることを確認するには、**showcnseventsubject** コマンドを使用します。Cisco Networking Service イメージ エージェントのサブジェクト名は cisco.mgmt.cns.image で始まります。

CNS 設定エージェントの設定例

例：Cisco Networking Service エージェントのイネーブル化および設定

次に、**cnsconfigpartial** コマンドで設定エージェントをイネーブルにすることから始まり、さまざまな Cisco Networking Service エージェントをイネーブル化および設定して、リモート デバイス上

で差分（部分）設定を行う例を示します。Cisco Networking Service 設定エンジンの IP アドレスは 172.28.129.22、ポート番号は 80 です。Cisco Networking Service EXEC エージェントを IP アドレス 172.28.129.23 で、Cisco Networking Service イベント エージェントを IP アドレス 172.28.129.24 でイネーブルにします。Cisco Networking Service イベント エージェントをイネーブルにするまで、他の Cisco Networking Service エージェントは動作しません。

```
cns config partial 172.28.129.22 80
cns exec 172.28.129.23 source 172.22.2.2
cns event 172.28.129.24 source 172.22.2.1
exit
```

次に、CLI を使用して Cisco Networking Service イメージ エージェント パラメータを設定する例を示します。GigabitEthernet インターフェイス 0/1/1 の IP アドレスを使用するようにイメージ ID を指定し、Cisco Networking Service イメージ エージェント サービスのパスワードを設定し、Cisco Networking Service イメージ アップグレード再試行間隔を 4 分間に設定し、イメージ管理サーバおよびステータス サーバを設定します。

```
cns id GigabitEthernet0/1/1 ipaddress image
cns image retry 240
cns image password abctext
cns image server https://10.21.2.3/cns/imgsvr status https://10.21.2.3/cns/status/
```

次の例では、Cisco Networking Service イベント バスを使用するように Cisco Networking Service イメージ エージェントを設定します。ネットワーキング デバイスのハードウェア シリアル番号としてイメージ ID を指定し、複数のパラメータを指定して Cisco Networking Service イベント エージェントをイネーブル化します。さらにキーワードまたはオプションを指定せずに Cisco Networking Service イメージ エージェントをイネーブル化します。Cisco Networking Service イメージ エージェントは、Cisco Networking Service イベント バス上でイベントを待ち受けます。

```
cns id hardware-serial image
cns event 10.21.9.7 11011 keepalive 240 120 failover-time 5
cns image
cns image password abctext
```

例：Cisco Networking Service イメージのサーバからの取得

次に、**cnsimageretrieve** コマンドを使用して、Cisco Networking Service イメージ エージェントがファイル サーバをポーリングする例を示します。Cisco Networking Service イメージ エージェントがすでにイネーブルになっているとすると、ここで指定されたファイルサーバとステータスサーバのパスによって既存のイメージエージェントサーバおよびステータス設定が上書きされます。新しいファイルサーバがポーリングされ、新しいイメージがある場合はネットワーキングデバイスにダウンロードされます。

```
cns image retrieve server https://10.19.2.3/cns/ status https://10.19.2.3/cnsstatus/
```

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
Cisco Networking Service コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例。	『Cisco IOS Cisco Networking Services Command Reference』
Cisco Networking Service 設定エンジン（CE）	『Cisco CNS Configuration Engine Administrator Guide, 1.3』

標準および RFC

標準/RFC	タイトル
この機能によりサポートされる新規または変更された標準/RFC はありません。またこの機能による既存の標準/RFC のサポートに変更はありません。	--

MIB

MIB	MIB のリンク
この機能によってサポートされる新しい MIB または変更された MIB はありません。またこの機能による既存 MIB のサポートに変更はありません。	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

CNS 設定エージェントの機能情報

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 *3* : ***CNS*** 設定エージェントの機能情報

機能名	リリース	機能情報
CNS 設定エージェント	Cisco IOS XE Release 2.1 12.0(18)ST 12.0(22)S 12.2(2)T 12.2(8)T 12.2(33)SRA 12.2(33)SB 12.2(33)SXI	Cisco Networking Service 設定エージェント機能は、次を提供してルーティング デバイスをサポートします。 • 初期設定 • 差分（部分）設定 • 同期設定更新 この機能により、次のコマンドが導入または変更されました。**cns config cancel**、**cns config initial**、**cns config partial**、**cns config retrieve**、**cns password**、**debug cns config**、**debug cns xml-parser**、**show cns config outstanding**、**show cns config stats**、**show cns config status**

第 4 章

Cisco Networking Service 再試行/間隔指定の設定取得拡張

- 機能情報の確認，37 ページ
- CNS 再試行/間隔指定の設定取得拡張について，38 ページ
- CNS 再試行/間隔指定の設定取得拡張の設定方法，38 ページ
- CNS 再試行/間隔指定の設定取得拡張の設定例，39 ページ
- その他の参考資料，40 ページ
- CNS 再試行/間隔指定の設定取得拡張の機能情報，41 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

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。

CNS 再試行/間隔指定の設定取得拡張について

Cisco Networking Service 再試行/間隔指定の設定取得拡張

Cisco Networking Service 再試行/間隔指定の設定取得拡張機能は、**cnsconfigretrieve** コマンドに新しい機能を追加します。これにより、トラステッド サーバから設定の取得を試行する再試行間隔および試行まで待機する時間（秒）を指定できるようになります。

CNS 再試行/間隔指定の設定取得拡張の設定方法

Cisco Networking Service 設定のサーバからの取得

コンフィギュレーション サーバにデバイスの設定を要求するには、次の作業を実行します。**cnstrusted-server** コマンドを使用して、使用可能な（トラステッド）コンフィギュレーション サーバを指定します。

はじめる前に

この作業では、トラステッド サーバが指定済みであることを前提としています。

手順の概要

1. **enable**
2. **configureterminal**
3. **cnsconfigretrieve** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page***page*] [**overwrite-startup**] [**retry***retries***interval***seconds*] [**syntax-check**] [**no-persist**] [**source***interface name*] [**status***url*] [**event**] [**inventory**]

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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。

	コマンドまたはアクション	目的
ステップ 3	**cnsconfigretrieve** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page***page*] [**overwrite-startup**] [**retry***retries***interval***seconds*] [**syntax-check**] [**no-persist**] [**source***interface name*] [**status***url*] [**event**] [**inventory**] 例： `Device(config)# cns config retrieve server1 retry 5 interval 45`	デバイスが Web サーバから設定データを取得できるようにします。 • **retry** キーワードは 1 ～ 100 の範囲の数値で、1 ～ 3600 秒の範囲の **interval** の入力を要求します。 （注） トラブルシューティングのヒント 取得プロセスを停止する場合は、Ctrl+Shift+6 キーを入力します。

CNS 再試行/間隔指定の設定取得拡張の設定例

例：Cisco Networking Service 設定のサーバからの取得

Cisco Networking Service トラステッド サーバからの設定データの取得

次に、10.1.1.1 のトラステッド サーバに設定を要求する例を示します。

```
cns trusted-server all 10.1.1.1
exit
cns config retrieve 10.1.1.1
```

次に、**cns config retrieve** コマンドを使用して、10.1.1.1 にあるトラステッド サーバに設定を要求し、Cisco Networking Service 設定取得間隔を設定する例を示します。

```
cns trusted-server all 10.1.1.1
exit
cns config retrieve 10.1.1.1 retry 50 interval 1500
CNS Config Retrieve Attempt 1 out of 50 is in progress
Next cns config retrieve retry is in 1499 seconds (Ctrl-Shft-6 to abort this command).
..
00:26:40: %CNS-3-TRANSPORT: CNS_HTTP_CONNECTION_FAILED:10.1.1.1 -Process= "CNS config retv",
 ipl= 0, pid= 43
```

00:26:40: %CNS-3-TRANSPORT: CNS_HTTP_CONNECTION_FAILED -Process= "CNS config retv", ipl= 0, pid= 43.....

```
cns config retrieve 10.1.1.1
```

取得したデータの実行コンフィギュレーション ファイルへの適用

次に、サーバから取得した設定データをチェックし、実行コンフィギュレーション ファイルにだけ適用する例を示します。Cisco Networking Service 設定エージェントは、設定データの取得に成功するか、または 5 回失敗するまで、30 秒間隔で設定データを取得しようとします。

```
cns config retrieve 10.1.1.1 syntax-check no-persist retry 5 interval 30
```

取得データによるスタートアップ コンフィギュレーション ファイルの上書き

次に、スタートアップコンフィギュレーションファイルをサーバから取得した設定データで上書きする例を示します。この設定データは実行コンフィギュレーションには適用されません。

```
cns config retrieve 10.1.1.1 syntax-check no-persist retry 5 interval 30
cns config retrieve 10.1.1.1 overwrite-startup
```

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
Cisco Networking Service コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例。	『Cisco IOS Cisco Networking Services Command Reference』
Cisco Networking Service 設定エンジン（CE）	『Cisco CNS Configuration Engine Administrator Guide, 1.3』

標準および RFC

標準/RFC	タイトル
この機能によりサポートされる新規または変更された標準/RFC はありません。またこの機能による既存の標準/RFC のサポートに変更はありません。	--

MIB

MIB	MIB のリンク
この機能によってサポートされる新しい MIB または変更された MIB はありません。またこの機能による既存 MIB のサポートに変更はありません。	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

CNS 再試行/間隔指定の設定取得拡張の機能情報

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 *4*：*Cisco Networking Service* 再試行/間隔指定の設定取得拡張の機能情報

機能名	リリース	機能情報
Cisco Networking Service 再試行/間隔指定の設定取得拡張	Cisco IOS XE Release 2.1 12.4(15)T 12.2(33)SRC 12.2(33)SB 12.2(50)SY	Cisco Networking Service 再試行/間隔指定の設定取得拡張機能は、**cnsconfigretrieve** コマンドに 2 つのオプションを追加して、トラステッド サーバの設定の取得を試行するまでに待機する時間（秒）を指定できるようにします。設定エージェントが到達不能サーバに対して試行し続けることがないように、再試行回数は100回に制限されています。**cnsconfigretrieve** コマンドを中断するには、キーボードの **Ctrl+Shift+6** の組み合わせを使用します。 この機能により、**cnsconfigretrieve** コマンドが変更されました。

第 5 章

Cisco Networking Service インタラクティブ CLI

- 機能情報の確認，43 ページ
- CNS インタラクティブ CLI について，43 ページ
- その他の参考資料，44 ページ
- CNS インタラクティブ CLI の機能情報，44 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

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。

CNS インタラクティブ CLI について

Cisco Networking Service インタラクティブ CLI

Cisco Networking Service インタラクティブ CLI 機能は、ユーザ入力のプロンプトを生成するコマンドなど、インタラクティブ コマンドをデバイスに送信できる XML インターフェイスを提供します。この機能の利点は、インタラクティブ コマンドが完全に処理される前にコマンドを中断できることです。たとえば、大量の出力を生成するコマンドの場合、XML インターフェイスをカス

タマイズして、出力サイズや出力の累積時間を制限できます。プログラム可能なインターフェイスを使用して（コマンドを手動で中断する場合と同様に）正常終了前にコマンドを中断する機能は、その機能を使用する可能性のある診断アプリケーションの効率を大幅に向上させます。この新しい XML インターフェイスでは、単一のセッションで複数のコマンドを処理することも可能です。各コマンドの応答は 1 つにまとめられ、単一の応答イベントで送信されます。

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
Cisco Networking Service コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例	『Cisco IOS Cisco Networking Services Command Reference』
Cisco Networking Service 設定エンジン（CE）	『Cisco CNS Configuration Engine Administrator Guide, 1.3』

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

CNS インタラクティブ CLI 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 5：*Cisco Networking Service* インタラクティブ *CLI* の機能情報

機能名	リリース	機能情報
Cisco Networking Service インタラクティブ CLI	Cisco IOS XE Release 2.1 12.0(28)S 12.2(18)SXE 12.2(18)SXF2 12.2(33)SRC 12.2(33)SXI	Cisco Networking Service インタラクティブ CLI 機能では、ユーザ入力のプロンプトを生成するコマンドなど、インタラクティブ コマンドをデバイスに送信できる XML インターフェイスが導入されます。

第 6 章

コマンド スケジューラ（Kron）

- 機能情報の確認，47 ページ
- コマンド スケジューラの制約事項，47 ページ
- コマンド スケジューラ（Kron）について，48 ページ
- コマンド スケジューラ（Kron）の設定方法，48 ページ
- コマンド スケジューラ（Kron）の設定例，52 ページ
- その他の参考資料，53 ページ
- コマンド スケジューラ（Kron）の機能情報，54 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

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。

コマンド スケジューラの制約事項

コマンド スケジューラのポリシー リスト内に指定する EXEC CLI は、プロンプトを生成するものや、キーストロークで強制終了できるものであってはいけません。コマンド スケジューラは完全に自動化された機能として設計されており、手動による介入はできません。

コマンド スケジューラ（Kron）について

コマンド スケジューラ

システム起動用コマンド スケジューラ（KRON）ポリシー機能は、システム起動時にコマンド スケジューラをサポートできるようにします。

コマンド スケジューラを使用すると、省略していない EXEC モードの CLI コマンドを、特定の間隔で、特定の日時に、またはシステム起動時に、1 回実行するようにスケジュールできます。当初 Cisco Networking Service コマンドで動作するよう設計されたコマンド スケジューラは、より広範なアプリケーションになりました。Cisco Networking Service イメージ エージェント機能を使用すると、ファイアウォール外のリモート デバイスやネットワーク アドレス変換（NAT）アドレスを使用するリモート デバイスは、コマンド スケジューラを使用して周期的に CLI を起動してデバイスで稼働するイメージを更新できます。

コマンド スケジューラには 2 つの基本的なプロセスがあります。ポリシー リストは、同時刻または同間隔で実行される、完全修飾された EXEC CLI コマンドを含む行で構成されます。次に、1 つまたは複数のポリシー リストが一定間隔後、特定の日時、またはシステム起動時に実行されるようスケジューリングします。スケジュールした各オカレンスは、一度だけまたは繰り返し実行するように設定できます。

コマンド スケジューラ（Kron）の設定方法

コマンド スケジューラ ポリシー リストおよびオカレンスの設定

コマンド スケジューラのオカレンスは、スケジュール イベントとして定義されます。ポリシー リストは、一定間隔後、特定の日時、またはシステム起動時に実行されるように設定します。ポリシー リストは、1 回、ワンタイム イベントとして、または繰り返しイベントとして実行できます。

コマンド スケジューラ オカレンスは、関連付けられたポリシー リストが設定される前にスケジューリングできますが、ポリシー リストが実行されるようスケジューリングする前にポリシー リストを設定するように勧める警告が表示されます。

はじめる前に

EXEC Cisco Networking Service コマンドのコマンド スケジューラ ポリシー リストをセットアップし、コマンド スケジューラ オカレンスを設定して、Cisco Networking Service コマンドを実行するまでの時間または間隔を指定するには、次の作業を実行します。

コマンド スケジューラ ポリシー リスト

ポリシー リストは、1 行以上の完全修飾 EXEC CLI コマンドで構成されます。ポリシー リスト内のすべてのコマンドは、コマンド スケジューラが **kronoccurrence** コマンドを使用してポリシー リストを実行するときに実行されます。異なる時刻に実行される CLI コマンドには別のポリシー

リストを使用します。編集機能はありません。ポリシー リストは設定した順序で実行されます。エントリを削除するには、**cli** コマンドの **no** 形式の後に適切な EXEC コマンドを使用します。既存のポリシーリスト名を使用すると、新しいエントリはそのポリシーリストの最後に追加されます。ポリシー リスト内のエントリを表示するには、**showrunning-config** コマンドを使用します。ポリシーリストを1回だけ実行するようにスケジューリングした場合、実行後は**showrunning-config** コマンドを使用してもポリシー リストは表示されません。

ポリシーリストは、ポリシーリストがスケジューリングされた後に設定できますが、各ポリシーリストは、実行するようスケジューリングされる前に設定する必要があります。

コマンド スケジューラ オカレンス

クロック時間は、コマンド スケジューラ オカレンスが実行されるようスケジューリングする前に、ルーティング デバイスに設定する必要があります。クロック時間が設定されていない場合、**kronoccurrence** コマンドを入力すると、警告メッセージがコンソール画面に表示されます。クロック時間を設定するには、**clock** コマンドまたはネットワーク タイム プロトコル（NTP）を使用します。

コマンド スケジューラによって実行される EXEC CLI は、ルーティング デバイス上でテストして、プロンプトを生成したり、キーストロークで実行が中断したりすることなく実行されるかどうかを確認する必要があります。CLI 構文エラーがある場合、コマンド スケジューラはそのポリシーリスト全体を削除してしまうため、初めにテストしておくことが重要です。ポリシーリストを削除する場合は、CLI の依存関係によってエラーが発生しないようにします。

conditional キーワードを **kronpolicy-list** コマンドに指定すると、エラーが発生した場合にコマンドの実行は停止されます。

（注）
- 同時に実行するようスケジューリングできるポリシー リストは 31 個以下です。
- ワンタイム オカレンスをスケジューリングした場合は、オカレンスの実行後に **showrunning-config** コマンドを使用しても、そのオカレンスは表示されません。

>

手順の概要

1. **enable**
2. **configureterminal**
3. **kronpolicy-list***list-name*[**conditional**]
4. **cli***command*
5. **exit**
6. **kronoccurrence***occurrence-name*[**user***username*] {**in**[[*numdays:*]*numhours:*]*nummin*| **at***hours:min*[[*month*] *day-of-month*] [*day-of-week*]} {**oneshot**| **recurring**| **system-startup**}
7. **policy-list***list-name*
8. **exit**
9. **showkronschedule**

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**kronpolicy-list***list-name*[**conditional**] 例： `Device(config)# kron policy-list cns-weekly`	新規または既存のコマンド スケジューラ ポリシー リストの名前を指定し、kron-policy コンフィギュレーション モードを開始します。 • *list-name* が新規の場合は、新しいポリシー リスト構造が作成されます。 • *list-name* が既存のものである場合は、その既存のポリシー リスト構造にアクセスします。ポリシー リストは設定した順に実行され、編集機能はありません。 • オプションの **conditional** キーワードを使用すると、エラーが発生した場合にコマンドの実行は停止されます。
ステップ 4	**cli***command* 例： `Device(config-kron-policy)# cli cns image retrieve server https://10.19.2.3/cnsweek/ status https://10.19.2.3/cnsstatus/week/`	指定されたコマンド スケジューラ ポリシー リストのエントリとして追加される完全修飾 EXEC コマンドおよび関連する構文を指定します。 • 各エントリは、設定した順にポリシー リストに追加されます。 • この手順を繰り返して、同時刻または同間隔で実行する他の EXEC CLI コマンドをポリシー リストに追加します。 （注） プロンプトを生成したり、キーストロークで実行が中断されたりする EXEC コマンドは、エラーとなります。
ステップ 5	**exit** 例： `Device(config-kron-policy)# exit`	kron-policy コンフィギュレーション モードを終了し、デバイスをグローバル コンフィギュレーション モードに戻します。

	コマンドまたはアクション	目的
ステップ 6	**kronoccurrence***occurrence-name*[**user***username*] {**in**[[*numdays:*]*numhours:*]*nummin*| **at***hours:min*[[*month*] *day-of-month*] [*day-of-week*]} {**oneshot**| **recurring**| **system-startup**} 例： `Device(config)# kron occurrence may user sales at 6:30 may 20 oneshot`	新規または既存のコマンドスケジューラ オカレンスの名前とスケジュールを指定し、kron-occurrence コンフィギュレーション モードを開始します。 • このコマンドの設定時に開始するタイマーが設定されたデルタ時間間隔を指定するには、**in** キーワードを使用します。 • 日時を指定するには、**at** キーワードを使用します。 • コマンドスケジューラ オカレンスを 1 回または繰り返しスケジューリングするには、**oneshot** キーワードまたは **recurring** キーワードのいずれかを選択します。システム起動時にオカレンスを実行する場合は、オプションの **system-startup** キーワードを追加します。
ステップ 7	**policy-list***list-name* 例： `Device(config-kron-occurrence)# policy-list sales-may`	コマンド スケジューラ ポリシー リストを指定します。 • 各エントリは、設定された順にオカレンス リストに追加されます。 （注） ポリシー リスト内の CLI コマンドが、プロンプトを生成したりキーストロークによって中断されたりすると、エラーが生成され、そのポリシー リストは削除されます。
ステップ 8	**exit** 例： `Device(config-kron-occurrence)# exit`	kron-occurrence コンフィギュレーション モードを終了し、デバイスをグローバル コンフィギュレーション モードに戻します。 • この手順を繰り返して、グローバル コンフィギュレーション モードを終了します。
ステップ 9	**showkronschedule** 例： `Device# show kron schedule`	（任意）コマンドスケジューラ オカレンスのステータスおよびスケジュール情報を表示します。

例

次の例では、設定されている全コマンドスケジューラ オカレンスのステータスおよびスケジュール情報が表示されます。

```
Device# show kron schedule
```

```
Kron Occurrence Schedule
cns-weekly inactive, will run again in 7 days 01:02:33
may inactive, will run once in 32 days 20:43:31 at 6:30 on May 20
```

トラブルシューティングのヒント

コマンド スケジューラのコマンド操作をトラブルシューティングするには、特権 EXEC モードで **debugkron** コマンドを使用します。デバッグ コマンドは注意して使用してください。生成される出力量によってデバイスの動作が遅くなったり、停止したりすることがあります。

コマンド スケジューラ（Kron）の設定例

例：コマンド スケジューラ ポリシー リストおよびオカレンス

次に、Cisco Networking Service コマンドを含む 2 つの EXEC CLI セットを実行するように、cns-weekly という名前のコマンドスケジューラ ポリシーを設定する例を示します。そして、そのポリシーを他の 2 つのポリシーと一緒に、7 日と 1 時間 30 分ごとに実行するようにスケジュールします。

```
kron policy-list cns-weekly
cli cns image retrieve server http://10.19.2.3/week/ status http://10.19.2.5/status/week/
cli cns config retrieve page /testconfig/config.asp no-persist
exit
kron occurrence week in 7:1:30 recurring
policy-list cns-weekly
policy-list itd-weekly
policy-list mkt-weekly
```

次に、Cisco Networking Service コマンドを実行してリモート サーバから特定のイメージを取得するように、sales-may という名前のコマンド スケジューラ ポリシーを設定する例を示します。そして、そのポリシーを 5 月 20 日の午前 6:30 に一度だけ実行するようにスケジュールします。

```
kron policy-list sales-may
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence may at 6:30 May 20 oneshot
policy-list sales-may
```

次に、Cisco Networking Service コマンドを実行してリモート サーバから特定のイメージを取得するように、image-sunday という名前のコマンド スケジューラ ポリシーを設定する例を示します。そして、そのポリシーを毎週日曜日の午前 7:30 に実行するようにスケジュールします。

```
kron policy-list image-sunday
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence sunday user sales at 7:30 sunday recurring
policy-list image-sunday
```

次に、Cisco Networking Service コマンドを実行してリモート サーバから特定のファイルを取得するように、file-retrieval という名前のコマンド スケジューラ ポリシーを設定する例を示します。そして、そのポリシーをシステム起動時に実行するようにスケジュールします。

```
kron policy-list file-retrieval
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
```

```
kron occurrence system-startup
policy-list file-retrieval
```

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
Cisco Networking Service コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例。	『Cisco IOS Cisco Networking Services Command Reference』
Cisco Networking Service 設定エンジン（CE）	『Cisco CNS Configuration Engine Administrator Guide, 1.3』

標準および RFC

標準/RFC	タイトル
この機能によりサポートされる新規または変更された標準/RFC はありません。またこの機能による既存の標準/RFC のサポートに変更はありません。	--

MIB

MIB	MIB のリンク
この機能によってサポートされる新しい MIB または変更された MIB はありません。またこの機能による既存 MIB のサポートに変更はありません。	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

コマンド スケジューラ（Kron）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 *6*：コマンド スケジューラ（*Kron*）の機能情報

機能名	リリース	機能情報
コマンドスケジューラ（Kron）	Cisco IOS XE Release 2.1 12.3(1) 12.2(33)SRA 12.2(33)SRC 12.2(33)SB 12.2(33)SXI 12.2(50)SY	コマンドスケジューラ機能は、一部の EXEC CLI コマンドの実行を特定の時刻または特定の間隔でスケジュールする機能を提供します。 この機能により、次のコマンドが導入または変更されました。**cli**、**debugkron**、**kronoccurrence**、**kronpolicy-list**、**policy-list**、**showkronschedule**

機能名	リリース	機能情報
システム起動用コマンド スケジューラ（Kron）ポリシー	12.2(33)SRC 12.2(50)SY 12.2(33)SB 12.4(15)T	システム起動用コマンド スケジューラ（Kron）ポリシー機能は、システム起動時にコマンド スケジューラ機能をサポートできるようにします。

第 7 章

ネットワーク設定プロトコル

ネットワーク設定プロトコル（NETCONF）は、ネットワークデバイスの管理、設定データの取得、および新しい設定データのアップロードと操作を行うための簡単なメカニズムを定義するものです。NETCONF では、設定データおよびプロトコルメッセージとして拡張可能マークアップ言語（XML）ベースのデータ符号化を使用します。

- 機能情報の確認，57 ページ
- NETCONF の前提条件，58 ページ
- NETCONF の概要，58 ページ
- NETCONF の設定方法，58 ページ
- NETCONF の設定例，66 ページ
- NETCONF に関する追加情報，69 ページ
- NETCONF の機能情報，70 ページ
- 用語集，71 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

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。

NETCONF の前提条件

netconfmax-session コマンドで指定した NETCONF セッションごとに vty 回線を用意する必要があります。

NETCONF の概要

NETCONF 通知

NETCONF は、NETCONF 上で設定変更の通知を送信します。通知は、設定変更が行われたことを示すイベントです。変更には、設定の追加、削除、または修正があります。通知は、適切に行われた設定作業の最後に、設定内で変更された設定の各行について個別のメッセージではなく、一連の変更を示す 1 つのメッセージとして送信されます。

NETCONF の設定方法

NETCONF ネットワーク マネージャ アプリケーションの設定

手順の概要

1. NETCONF を SSH サブシステムとして呼び出すように、NETCONF ネットワーク マネージャ アプリケーションを設定するには、次の CLI 文字列を使用します。
2. NETCONF セッションの確立後すぐに、<hello> を含む次のような XML 文書を送信することによって、サーバの機能を示します。
3. 次の XML 文字列を使用して、NETCONF ネットワーク マネージャ アプリケーションが NETCONF 通知を送受信できるようにします。
4. NETCONF ネットワーク マネージャ アプリケーションの NETCONF 通知の送信または受信を停止するには、次の XML 文字列を使用します。

手順の詳細

ステップ 1 NETCONF を SSH サブシステムとして呼び出すように、NETCONF ネットワーク マネージャ アプリケーションを設定するには、次の CLI 文字列を使用します。

例：

```
Unix Side: ssh-2 -s companyname@10.1.1.1 netconf
```

ステップ 2 NETCONF セッションの確立後すぐに、<hello> を含む次のような XML 文書を送信することによって、サーバの機能を示します。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
    <hello>
      <capabilities>
        <capability>
            urn:ietf:params:xml:ns:netconf:base:1.0
          </capability>
          <capability>
            urn:ietf:params:ns:netconf:capability:startup:1.0
          </capability>
       </capabilities>
    <session-id>4<session-id>
</hello>]]>]]>
```

クライアントは、<hello> を含む XML 文書を送信して応答します。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
 <hello>
   <capabilities>
       <capability>
           urn:ietf:params:xml:ns:netconf:base:1.0
     </capability>
    </capabilities>
</hello>]]>]]>
```

（注） この例では、サーバが <hello> メッセージを送信した後にクライアントからのメッセージが続いていますが、NETCONF サブシステムの初期化後すぐに、両サイドからほぼ同時にメッセージが送信されます。

ヒント すべての NETCONF 要求は、要求の終わりを示す]]>]]> で終わる必要があります。]]>]]> のシーケンスが送信されるまで、デバイスは要求を処理しません。

特定の例については、「例：NETCONF over SSHv2 の設定」を参照してください。

ステップ 3 次の XML 文字列を使用して、NETCONF ネットワーク マネージャ アプリケーションが NETCONF 通知を送受信できるようにします。

例：

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.0"><notification-on/>
</rpc>]]>]]>
```

ステップ 4 NETCONF ネットワーク マネージャ アプリケーションの NETCONF 通知の送信または受信を停止するには、次の XML 文字列を使用します。

例：

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.13"><notification-off/>
</rpc>]]>]]>
```

NETCONF ペイロードの配信

NETCONF ペイロードをネットワーク マネージャ アプリケーションに配信するには、次の XML 文字列を使用します。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.cisco.com/cpi_10/schema" elementFormDefault="qualified"
 attributeFormDefault="unqualified" xmlns="http://www.cisco.com/cpi_10/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <!--The following elements define the cisco extensions for the content of the filter
element in a <get-config> request. They allow the client to specify the format of the
response and to select subsets of the entire configuration to be included.-->
   <xs:element name="config-format-text-block">
      <xs:annotation>
         <xs:documentation>If this element appears in the filter, then the client is
requesting that the response data be sent in config command block format.</xs:documentation>

      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="text-filter-spec" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="config-format-text-cmd">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="text-filter-spec"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="config-format-xml">
      <xs:annotation>
         <xs:documentation>When this element appears in the filter of a get-config request,
 the results are to be returned in E-DI XML format. The content of this element is treated
 as a filter.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:complexContent>
            <xs:extension base="xs:anyType"/>
         </xs:complexContent>
      </xs:complexType>
   </xs:element>
   <!--These elements are used in the filter of a <get> to specify operational data to
return.-->
   <xs:element name="oper-data-format-text-block">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="show" type="xs:string" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="oper-data-format-xml">
      <xs:complexType>
         <xs:sequence>
```

```
            <xs:any/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <!--When confing-format-text format is specified, the following describes the content
of the data element in the response-->
   <xs:element name="cli-config-data">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="cmd" type="xs:string" maxOccurs="unbounded">
               <xs:annotation>
                  <xs:documentation>Content is a command. May be multiple
lines.</xs:documentation>
               </xs:annotation>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="cli-config-data-block" type="xs:string">
      <xs:annotation>
         <xs:documentation>The content of this element is the device configuration as it
would be sent to a terminal session. It contains embedded newline characters that must be
preserved as they represent the boundaries between the individual command
lines</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="text-filter-spec">
      <xs:annotation>
         <xs:documentation>If this element is included in the config-format-text element,
then the content is treated as if the string was appended to the "show running-config"
command line.</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="cli-oper-data-block">
      <xs:complexType>
         <xs:annotation>
            <xs:documentation> This element is included in the response to get operation.
Content of this element is the operational data in text format.</xs:documentation>
         </xs:annotation>
         <xs:sequence>
            <xs:element name="item" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="exec"/>
                     <xs:element name="show"/>
                     <xs:element name="response"/>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:schema>
```

NETCONF 通知のフォーマット

NETCONF ネットワーク マネージャ アプリケーションは、.xsd スキーマ ファイルを使用して、NETCONF ネットワーク マネージャ アプリケーションと NETCONF over SSHv2 または NETCONF over BEEP が稼働するデバイスとの間で送信される XML NETCONF 通知メッセージのフォーマットを記述します。それらのファイルはブラウザまたはスキーマ読み取りツールで表示できます。これらのスキーマを使用して XML の妥当性を検証できます。これらのスキーマで記述するのは、交換されるデータのフォーマットであって内容ではありません。

NETCONF は <edit-config> 機能を使用して、特定の設定すべてを特定のターゲット設定にロードします。この新しい設定を入力した場合、ターゲット設定は置き換えられません。ターゲット設定は、要求の送信元のデータおよび要求された動作に応じて変更されます。

次に、CLI、CLI ブロック、および XML の各フォーマットの NETCONF <edit-config> 機能のスキーマを示します。

NETCONF <edit-config> 要求：CLI フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cli-config-data>
<cmd>hostname test</cmd>
            <cmd>interface fastEthernet0/1</cmd>
            <cmd>ip address 192.168.1.1 255.255.255.0</cmd>
</cli-config-data>
      </config>
   </edit-config>
</rpc>]]>]]>
```

NETCONF <edit-config> 応答：CLI フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:netconf:base:1.0">
   <ok/>
</rpc-reply>]]>]]>
```

NETCONF <edit-config> 要求：CLI ブロック フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="netconf.mini.edit.3">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cli-config-data-block>
            hostname bob
            interface fastEthernet0/1
            ip address 192.168.1.1 255.255.255.0
         </cli-config-data-block>
      </config>
   </edit-config>
</rpc>]]>]]>
```

NETCONF <edit-config> 応答：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="netconf.mini.edit.3" xmlns="urn:ietf:params:netconf:base:1.0">
   <ok/>
</rpc-reply>]]>]]>
```

次に、CLI および CLI ブロックの各フォーマットの NETCONF <get-config> 機能のスキーマを示します。

NETCONF <get-config> 要求：CLI フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get-config>
      <source>
         <running/>
      </source>
      <filter>
         <config-format-text-cmd>
            <text-filter-spec> | inc interface </text-filter-spec>
         </config-format-text-cmd>
</filter>
   </get-config>
</rpc>]]>]]>
```

NETCONF <get-config> 応答：CLI フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <data>
      <cli-config-data>
         <cmd>interface FastEthernet0/1</cmd>
         <cmd>interface FastEthernet0/2</cmd>
      </cli-config-data>
   </data>
</rpc-reply>]]>]]>
```

NETCONF <get-config> 要求：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get-config>
      <source>
         <running/>
      </source>
      <filter>
         <config-format-text-block>
            <text-filter-spec> | inc interface </text-filter-spec>
         </config-format-text-block>
      </filter>
   </get-config>
</rpc>]]>]]>
```

NETCONF <get-config> 応答：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <data>
      <cli-config-data-block>
         interface FastEthernet0/1
         interface FastEthernet0/2
      </cli-config-data-block>
   </data>
</rpc-reply>]]>]]>
```

NETCONF は <get> 機能を使用して、設定およびデバイスの状態情報を取得します。NETCONF <get> フォーマットは、Cisco IOS **show** コマンドに相当します。<filter> パラメータは、システム設定およびデバイス状態データの取得部分を指定します。<filter>パラメータが空の場合は、何も返されません。

次に、CLI および CLI ブロックの各フォーマットの <get> 機能のスキーマを示します。

NETCONF <get> 要求：CLI フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get>
       <filter>
          <config-format-text-cmd>
             <text-filter-spec> | include interface </text-filter-spec>
          </config-format-text-cmd>
          <oper-data-format-text-block>
             <exec>show interfaces</exec>
             <exec>show arp</exec>
          </oper-data-format-text-block>
       </filter>
    </get>
 </rpc>]]>]]>
```

NETCONF <get> 応答：CLI フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
       <cli-config-data>
<cmd>interface Loopback0</cmd>
<cmd>interface GigabitEthernet0/1</cmd>
<cmd>interface GigabitEthernet0/2</cmd>
</cli-config-data>
<cli-oper-data-block>
          <item>
             <exec>show interfaces</exec>
             <response>
                <!-- output of "show interfaces" ------>
             </response>
          </item>
          <item>
             <exec>show arp</exec>
             <response>
                <!-- output of "show arp" ------>
             </response>
          </item>
       </cli-oper-data-block>
    </data>
</rpc-reply>]]>]]>
```

NETCONF <get> 要求：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get>
       <filter>
          <config-format-text-block>
             <text-filter-spec> | include interface </text-filter-spec>
          </config-format-text-block>
          <oper-data-format-text-block>
             <exec>show interfaces</exec>
             <exec>show arp</exec>
          </oper-data-format-text-block>
       </filter>
    </get>
 </rpc>]]>]]>
```

NETCONF <get> 応答：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
```

```
        <cli-config-data-block>
interface Loopback0
interface GigabitEthernet0/1
interface GigabitEthernet0/2
        </cli-config-data-block>
        <cli-oper-data-block>
           <item>
              <exec>show interfaces</exec>
              <response>
                 <!-- output of "show interfaces" ------>
              </response>
           </item>
           <item>
              <exec>show arp</exec>
              <response>
                 <!-- output of "show arp" ------>
              </response>
           </item>
        </cli-oper-data-block>
    </data>
</rpc-reply>]]>]]>
```

NETCONF セッションのモニタリングおよびメンテナンス

（注）
- 4 個以上の同時 NETCONF セッションを設定する必要があります。
- 最大 16 個の同時 NETCONF セッションを設定できます。
- NETCONF では SSHv1 はサポートされません。

手順の概要

1. **enable**
2. **shownetconf**{**counters** | **session**|**schema**}
3. **debugnetconf**{**all** | **error**}
4. **clearnetconf**{**counters** | **sessions**}

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**shownetconf**{**counters** | **session**|**schema**} 例： `Device# show netconf counters`	NETCONF 情報を表示します。

	コマンドまたはアクション	目的
ステップ 3	**debugnetconf**{**all** | **error**} 例： `Device# debug netconf error`	NETCONF セッションのデバッグをイネーブルにします。
ステップ 4	**clearnetconf**{**counters** | **sessions**} 例： `Device# clear netconf sessions`	NETCONF 統計カウンタおよび NETCONF セッションをクリアし、関連するリソースを解放し、ロックを解除します。

NETCONF の設定例

例：NETCONF ネットワーク マネージャ アプリケーションの設定

次に、NETCONF を SSH サブシステムとして呼び出すように、NETCONF ネットワーク マネージャ アプリケーションを設定する例を示します。

```
Unix Side: ssh-2 -s companyname@10.1.1.1 netconf
```

NETCONF セッションの確立後すぐに、<hello> を含む次のような XML 文書を送信することによって、サーバの機能を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
    <hello>
      <capabilities>
        <capability>
            urn:ietf:params:xml:ns:netconf:base:1.0
          </capability>
          <capability>
            urn:ietf:params:ns:netconf:capability:startup:1.0
          </capability>
       </capabilities>
    <session-id>4<session-id>
</hello>]]>]]>
```

クライアントも <hello> を含む XML 文書を送信して応答します。

```
<?xml version="1.0" encoding="UTF-8"?>
 <hello>
   <capabilities>
       <capability>
           urn:ietf:params:xml:ns:netconf:base:1.0
     </capability>
    </capabilities>
</hello>]]>]]>
```

次の XML 文字列を使用して、NETCONF ネットワーク マネージャ アプリケーションが NETCONF 通知を送受信できるようにします。

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.0"><notification-on/>
</rpc>]]>]]>
```

次の XML 文字列を使用して、NETCONF ネットワーク マネージャ アプリケーションによる NETCONF 通知の送受信を停止します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.13"><notification-off/>
</rpc>]]>]]>
```

例：NETCONF セッションのモニタリング

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

```
Device# show netconf counters
NETCONF Counters
Connection Attempts:0: rejected:0 no-hello:0 success:0
Transactions
        total:0, success:0, errors:0
detailed errors:
        in-use 0        invalid-value 0         too-big 0
        missing-attribute 0     bad-attribute 0         unknown-attribute 0
        missing-element 0       bad-element 0   unknown-element 0
        unknown-namespace 0     access-denied 0         lock-denied 0
        resource-denied 0       rollback-failed 0       data-exists 0
        data-missing 0  operation-not-supported 0       operation-failed 0
        partial-operation 0
```

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

```
Device# show netconf session
(Current | max) sessions:   3 | 4
Operations received: 100                Operation errors: 99
Connection Requests: 5                  Authentication errors: 2   Connection Failures: 0
ACL dropped : 30
Notifications  Sent: 20
```

shownetconfschema コマンドの出力は、NETCONF 要求およびその要求に対する応答のエレメント構造を表示します。このスキーマは、適切な NETCONF 要求の作成およびその要求に対する応答の解析に使用できます。スキーマのノードについては RFC 4741 で規定されています。次に、**shownetconfschema** コマンドの出力例を示します。

```
Device# show netconf schema
New Name Space 'urn:ietf:params:xml:ns:netconf:base:1.0'
<VirtualRootTag> [0, 1] required
  <rpc-reply> [0, 1] required
    <ok> [0, 1] required
    <data> [0, 1] required
    <rpc-error> [0, 1] required
      <error-type> [0, 1] required
      <error-tag> [0, 1] required
      <error-severity> [0, 1] required
      <error-app-tag> [0, 1] required
      <error-path> [0, 1] required
      <error-message> [0, 1] required
      <error-info> [0, 1] required
        <bad-attribute> [0, 1] required
        <bad-element> [0, 1] required
```

```
      <ok-element> [0, 1] required
      <err-element> [0, 1] required
      <noop-element> [0, 1] required
      <bad-namespace> [0, 1] required
      <session-id> [0, 1] required
<hello> [0, 1] required
  <capabilities> 1 required
    <capability> 1+ required
<rpc> [0, 1] required
  <close-session> [0, 1] required
  <commit> [0, 1] required
    <confirmed> [0, 1] required
    <confirm-timeout> [0, 1] required
  <copy-config> [0, 1] required
    <source> 1 required
      <config> [0, 1] required
        <cli-config-data> [0, 1] required
          <cmd> 1+ required
        <cli-config-data-block> [0, 1] required
        <xml-config-data> [0, 1] required
          <Device-Configuration> [0, 1] required
            <> any subtree is allowed
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
    <target> 1 required
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
  <delete-config> [0, 1] required
    <target> 1 required
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
  <discard-changes> [0, 1] required
  <edit-config> [0, 1] required
    <target> 1 required
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
    <default-operation> [0, 1] required
    <test-option> [0, 1] required
    <error-option> [0, 1] required
    <config> 1 required
      <cli-config-data> [0, 1] required
        <cmd> 1+ required
      <cli-config-data-block> [0, 1] required
      <xml-config-data> [0, 1] required
        <Device-Configuration> [0, 1] required
          <> any subtree is allowed
  <get> [0, 1] required
    <filter> [0, 1] required
      <config-format-text-cmd> [0, 1] required
        <text-filter-spec> [0, 1] required
      <config-format-text-block> [0, 1] required
        <text-filter-spec> [0, 1] required
      <config-format-xml> [0, 1] required
      <oper-data-format-text-block> [0, 1] required
        <exec> [0, 1] required
        <show> [0, 1] required
      <oper-data-format-xml> [0, 1] required
        <exec> [0, 1] required
        <show> [0, 1] required
  <get-config> [0, 1] required
    <source> 1 required
      <config> [0, 1] required
        <cli-config-data> [0, 1] required
          <cmd> 1+ required
        <cli-config-data-block> [0, 1] required
```

```
        <xml-config-data> [0, 1] required
          <Device-Configuration> [0, 1] required
            <> any subtree is allowed
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
    <filter> [0, 1] required
      <config-format-text-cmd> [0, 1] required
        <text-filter-spec> [0, 1] required
      <config-format-text-block> [0, 1] required
        <text-filter-spec> [0, 1] required
      <config-format-xml> [0, 1] required
  <kill-session> [0, 1] required
    <session-id> [0, 1] required
  <lock> [0, 1] required
    <target> 1 required
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
  <unlock> [0, 1] required
    <target> 1 required
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
  <validate> [0, 1] required
    <source> 1 required
      <config> [0, 1] required
        <cli-config-data> [0, 1] required
          <cmd> 1+ required
        <cli-config-data-block> [0, 1] required
        <xml-config-data> [0, 1] required
          <Device-Configuration> [0, 1] required
            <> any subtree is allowed
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
  <notification-on> [0, 1] required
  <notification-off> [0, 1] required
```

NETCONF に関する追加情報

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
NETCONF コマンド：コマンド構文、コマンドモード、コマンド履歴、デフォルト、使用に関する注意事項、および例	『*Cisco IOS Cisco Networking Services Command Reference*』
セキュリティおよび IP アクセス リスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例	『*Cisco IOS Security Command Reference*』

標準および RFC

標準/RFC	タイトル
RFC 4251	『*The Secure Shell (SSH) Protocol Architecture*』
RFC 4252	『*The Secure Shell (SSH) Authentication Protocol*』
RFC 4741	『*NETCONF Configuration Protocol*』
RFC 4744	『*Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)*』

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

NETCONF 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 *7*：***NETCONF*** の機能情報

機能名	リリース	機能情報
NETCONF		NETCONF プロトコルは、ネットワーク デバイスの管理、設定データの取得、および新しい設定データのアップロードと操作の簡単なメカニズムを定義します。NETCONF では、設定データおよびプロトコル メッセージとして拡張可能マークアップ言語（XML）ベースのデータ符号化を使用します。 この機能により、次のコマンドが導入または変更されました。**clearnetconf**、**debugnetconf**、**shownetconf**
NETCONF XML PI		NETCONF プロトコルが拡張され、すべての Cisco IOS EXEC コマンドのフォーマット属性のサポートが追加されました。 次のコマンドが変更されました。**clearnetconf**、**debugnetconf**、**shownetconf**

用語集

BEEP：ブロック拡張可能交換プロトコル。コネクション型非同期相互作用のための汎用アプリケーション プロトコル フレームワーク。

NETCONF：ネットワーク設定プロトコル。ネットワーク デバイスの管理、設定データの取得、および新しい設定データのアップロードと操作の簡単なメカニズムを定義するプロトコル。

SASL：Simple Authentication and Security Layer。接続ベースのプロトコルに認証サポートを追加するためのインターネット標準方式。SASL を、セキュリティ アプライアンスと Lightweight Directory Access Protocol（LDAP）サーバとの間で使用して、ユーザ認証を強化できます。

SSHv2：セキュア シェル バージョン 2。SSH は、信頼性の高いトランスポート層の上部で実行され、強力な認証機能と暗号化機能を提供します。SSHv2 を使用すると、別のコンピュータにネットワークを介して安全にアクセスして安全にコマンドを実行できるようになります。

TLS：Transport Layer Security。相互認証、完全性のためのハッシュの使用、プライバシー保護のための暗号化を可能にすることで、クライアントとサーバとの間にセキュアな通信を実現するア

プリケーション レベルのプロトコルです。TLS では、証明書、公開キー、および秘密キーを使用します。

XML：Extensible Markup Language。World Wide Web Consortium（W3C）によって管理されている、情報構造を指定するマークアップ言語を作成するための構文を定義する標準。情報構造は、情報の外観（太字、イタリック体など）ではなく、情報のタイプ（加入者名やアドレスなど）を定義します。外部のプロセスでこれらの情報構造を操作し、さまざまなフォーマットで公開することができます。XML では、独自にカスタマイズしたマークアップ言語を定義できます。

第 8 章

NETCONF over SSHv2

ネットワーク設定プロトコル（NETCONF）over セキュア シェル バージョン 2（SSHv2）機能を使用して、暗号化転送により Cisco コマンドラインインターフェイス（CLI）を介してネットワーク設定を実行できます。NETCONF クライアントである NETCONF ネットワーク マネージャは、NETCONF サーバへのネットワーク転送としてセキュア シェル バージョン 2（SSHv2）を使用する必要があります。NETCONF サーバには複数の NETCONF クライアントが接続できます。

- 機能情報の確認，73 ページ
- NETCONF over SSHv2 の前提条件，74 ページ
- NETCONF over SSH の制約事項，74 ページ
- NETCONF over SSHv2 について，74 ページ
- NETCONF over SSHv2 の設定方法，76 ページ
- NETCONF over SSHv2 の設定例，83 ページ
- NETCONF over SSHv2 に関する追加情報，85 ページ
- NETCONF over SSHv2 の機能情報，86 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

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。

NETCONF over SSHv2 の前提条件

- NETCONF over SSHv2 では、**netconfmax-session** コマンドで指定した NETCONF セッションごとに vty 回線を用意する必要があります。

NETCONF over SSH の制約事項

- ネットワーク設定プロトコル（NETCONF）セキュア シェル バージョン 2（SSHv2）は、最大 16 の同時セッションをサポートします。
- SSH バージョン 2 のみサポートされます。

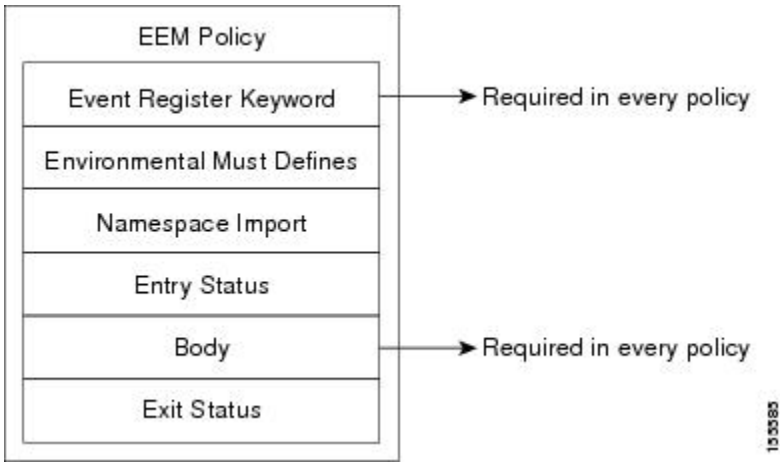
NETCONF over SSHv2 について

NETCONF over SSHv2

NETCONF over SSHv2 機能を実行するために、クライアント（シスコ ソフトウェアが稼働しているシスコ デバイス）はサーバ（NETCONF ネットワーク マネージャ）との SSH 転送接続を確立します。次の図に、基本的な NETCONF over SSHv2 ネットワークの構成を示します。クライアントとサーバは、セキュリティおよびパスワード暗号化に使用するキーを交換します。NETCONF を実行する SSHv2 セッションのユーザ ID およびパスワードは、認可および認証を行うために使用されます。そのユーザの権限レベルが適用されるため、十分に高い権限レベルでなければ、クライアントセッションから NETCONF 動作にフルアクセスできません。認証、許可、アカウンティング（AAA）が設定されている場合は、デバイスに対してユーザが直接SSHセッションを確立したかのように AAA サービスが使用されます。既存のセキュリティ設定を使用すると、NETCONF への移行がほぼシームレスに行われます。クライアントは認証に成功すると SSH 接続プロトコル

を呼び出し、SSH セッションを確立します。SSH セッションが確立されると、ユーザまたはアプリケーションは、「netconf」という SSH サブシステムとして NETCONF を呼び出します。

図 1：***NETCONF over SSHv2***



SSH バージョン 2

SSHv2 は、信頼性の高いトランスポート層の上部で実行され、強力な認証機能と暗号化機能を提供します。SSHv2 を使用すると、別のコンピュータにネットワークを介して安全にアクセスして安全にコマンドを実行できるようになります。

NETCONF は SSHv1 をサポートしていません。SSH バージョン 2 サーバの設定は、SSH バージョン 1 の設定と同様です。設定する SSH のバージョンを指定するには、**ipsshversion** コマンドを使用します。このコマンドを設定しない場合、デフォルトで SSH は互換モードで実行されます。バージョン 1 とバージョン 2 両方の接続が利用できます。

（注） SSH バージョン 1 は、標準で定義されていないプロトコルです。未定義のプロトコル（バージョン 1）にデバイスがフォールバックしないようにするには、**ipsshversion** コマンドを使用してバージョン 2 を指定する必要があります。

設定済みの Rivest, Shamir, and Adelman（RSA）キーを使用する SSH 接続をイネーブルにするには、**ipsshrsakeypair-name** コマンドを使用します。**ipsshrsakeypair-name** コマンドをキー ペアの名前を指定して設定すると、そのキー ペアが存在する場合は SSH がイネーブルになります。または、後でキー ペアが生成されると SSH がイネーブルになります。このコマンドを使用して SSH をイネーブルにする場合、ホスト名およびドメイン名を設定する必要はありません。

NETCONF over SSHv2 の設定方法

ホスト名とドメイン名を使用した SSH バージョン 2 のイネーブル化

このタスクを実行して、SSH バージョン 2 のデバイスを、ホスト名とドメイン名を使用して設定します。RSA キーペア設定を使用して、SSH バージョン 2 を設定することもできます（RSA キーペアを使用した SSH バージョン 2 のイネーブル化, （77 ページ）を参照）。

手順の概要

1. **enable**
2. **configureterminal**
3. **hostname***hostname*
4. **ipdomain-name***name*
5. **cryptokeygeneratersa**
6. **ipssh** [**timeout***seconds* | **authentication-retries***integer*]
7. **ipsshversion2**

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**hostname***hostname* 例： `Device(config)# hostname host1`	デバイスのホスト名を設定します。
ステップ 4	**ipdomain-name***name* 例： `Device(config)# ip domain-name domain1.com`	デバイスのドメイン名を設定します。

	コマンドまたはアクション	目的
ステップ 5	**cryptokeygeneratersa** 例： `Device(config)# crypto key generate rsa`	ローカルおよびリモート認証用に SSH サーバをイネーブルにします。
ステップ 6	**ipssh** [**timeout***seconds* | **authentication-retries***integer*] 例： `Device(config)# ip ssh timeout 120`	（任意）デバイス上で SSH 制御変数を設定します。
ステップ 7	**ipsshversion2** 例： `Device(config)# ip ssh version 2`	デバイスで実行する SSH のバージョンを指定します。

RSA キー ペアを使用した SSH バージョン 2 のイネーブル化

このタスクを実行して、ホスト名やドメイン名を設定せずに SSH バージョン 2 をイネーブルにします。設定したキー ペアがすでに存在している場合、または後で生成される場合、SSH バージョン 2 がイネーブルになります。ホスト名およびドメイン名の設定を使用して SSH バージョン 2 を設定することもできます（ホスト名とドメイン名を使用した SSH バージョン 2 のイネーブル化, （76 ページ）を参照）。

手順の概要

1. **enable**
2. **configureterminal**
3. **ipsshrsakeypair-name***keypair-name*
4. **cryptokeygeneratersausage-keyslabel***key-label***modulus***modulus-size*
5. **ipssh** [**timeout***seconds* | **authentication-retries***integer*]
6. **ipsshversion2**

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**ipsshrsakeypair-name***keypair-name* 例： `Device(config)# ip ssh rsa keypair-name sshkeys`	SSH を使用する際に使用する RSA キーペアを指定します。 （注） シスコ デバイスには複数の RSA キー ペアを設定できます。
ステップ 4	**cryptokeygeneratersausage-keyslabel***key-label***modulus***modulus-size* 例： `Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768`	デバイスでローカルおよびリモート認証を行う SSH サーバをイネーブルにします。 SSH バージョン 2 では、絶対サイズは 768 ビット以上である必要があります。 （注） RSA キー ペアを削除するには、**cryptokeyzeroizersa** コマンドを使用します。RSA コマンドを削除すると、SSH サーバが自動的にディセーブルになります。
ステップ 5	**ipssh** [**timeout***seconds* | **authentication-retries***integer*] 例： `Device(config)# ip ssh timeout 120`	デバイス上で SSH 制御変数を設定します。
ステップ 6	**ipsshversion2** 例： `Device(config)# ip ssh version 2`	デバイスで実行する SSH のバージョンを指定します。

リモート デバイスとの暗号化セッションの開始

リモートネットワーキングデバイスとの暗号化セッションを開始するには、次の作業を実行します（デバイスをイネーブルにする必要はありません。SSH はディセーブル モードで実行できます）。

UNIX または UNIX ライクなデバイスからは、通常、次のコマンドを使用して、SSH セッションを確立します。

```
ssh -2 -s user@router.example.com netconf
```

手順の概要

1. 次のいずれかを実行します。
   - **ssh** [**-v** {**1** | **2**}] [**-c** {**3des**| **aes128-cbc** | **aes192-cbc**| **aes256-cbc**}] [**-m**{**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*]

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	次のいずれかを実行します。
	• **ssh** [**-v** {**1** | **2**}] [**-c** {**3des**| **aes128-cbc** | **aes192-cbc**| **aes256-cbc**}] [**-m**{**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*]
	
	例：
	`Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24`
	
	例：
	`Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 user2@10.76.82.24`
		リモート ネットワーク デバイスとの暗号化されたセッションを開始します。
		1 つめの例は、SSH バージョン 2 の規定に準拠しています。より自然で一般的なセッション開始方法は、ユーザ名をホスト名に結合することです。たとえば、2 つめの設定例でも、1 つめの例と同じ結果が得られます。

トラブルシューティングのヒント

ip ssh version コマンドは、SSH の設定のトラブルシューティングに使用できます。バージョンを変更することによって、問題がある SSH バージョンを特定できます。

次の作業

ssh コマンドの詳細については、『Cisco IOS Security Command Reference』を参照してください。

セキュア シェル接続のステータスの確認

デバイス上の SSH 接続のステータスを表示するには、次の作業を実行します。

（注） 次の **show** コマンドは、ユーザ EXEC モードか特権 EXEC モードで使用できます。

手順の概要

1. **enable**
2. **showssh**
3. **showipssh**

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`	（任意）特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**showssh** 例： `Device# show ssh`	SSH サーバ接続のステータスを表示します。
ステップ 3	**showipssh** 例： `Device# show ip ssh`	SSH のバージョンおよび設定データを表示します。

例

次の **showssh** コマンドの出力には、SSH バージョン 2 の接続に関するステータスが表示されています。

```
Device# show ssh
Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
%No SSHv1 server connections running.
```

次の **showipssh** コマンドの出力には、イネーブルになっている SSH のバージョン、認証タイムアウト値、および認証の再試行回数が表示されています。

```
Device# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

NETCONF over SSHv2 のイネーブル化

NETCONF over SSHv2 をイネーブルにするには、次の作業を実行します。

はじめる前に

SSHv2 を有効にする必要があります。

（注） 同時 NETCONF セッションと同じ数以上の vty 行が設定されている必要があります。

（注）
- 4 個以上の同時 NETCONF セッションを設定する必要があります。
- 最大 16 個の同時 NETCONF セッションを設定できます。
- NETCONF では SSHv1 はサポートされません。

手順の概要

1. **enable**
2. **configureterminal**
3. **netconfssh**[**acl***access-list-number*]
4. **netconflock-time***seconds*
5. **netconfmax-sessions***session*
6. **netconfmax-message** size

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**netconfssh**[**acl***access-list-number*] 例： `Device(config)# netconf ssh acl 1`	NETCONF over SSHv2 をイネーブルにします。 • 任意で、この NETCONF セッションのアクセス コントロール リストを設定できます。
ステップ 4	**netconflock-time***seconds* 例： `Device(config)# netconf lock-time 60`	（任意）NETCONF 設定を中間操作が行われないようにロックする最長時間を秒単位で指定します。 • 有効な範囲は、1 ～ 300 秒です。デフォルト値は 10 秒です。
ステップ 5	**netconfmax-sessions***session* 例： `Device(config)# netconf max-sessions 5`	（任意）許容される同時 NETCONF セッションの最大数を指定します。 • 有効な範囲は、4 ～ 16 です。デフォルト値は 4 です。
ステップ 6	**netconfmax-message** size 例： `Device(config)# netconf max-message 37283`	（任意）NETCONF セッションで受信するメッセージの最大サイズをキロバイト（KB）で指定します。 • 有効な範囲は、1 ～ 2147483 KB です。デフォルト値は無限です。 • 最大サイズを無限に設定するには、**nonetconfmax-message** コマンドを使用します。

NETCONF over SSHv2 の設定例

例：ホスト名およびドメイン名を使用した SSHv2 のイネーブル化

```
configure terminal
hostname host1
ip domain-name example.com
crypto key generate rsa
ip ssh timeout 120
ip ssh version 2
```

RSA キーを使用したセキュア シェル バージョン 2 のイネーブル化の例

次に、RSA キーを使用してセキュア シェル バージョン 2 をイネーブルにする例を示します。

```
Device# configure terminal

Device(config)# ip ssh rsa keypair-name sshkeys

Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768
Device(config)# ip ssh timeout 120
Device(config)# ip ssh version 2
```

リモート デバイスとの暗号化セッションの開始の例

次に、UNIX または UNIX 系のデバイスから、リモート ネットワーキング デバイスとの暗号化 SSH セッションを開始する例を示します。

```
Device(config)# ssh -2 -s user@router.example.com netconf
```

NETCONF over SSHv2 の設定例

次に、NETCONF over SSHv2 を設定する例を示します。

```
Device# configure terminal
Device(config)# netconf ssh acl 1
Device(config)# netconf lock-time 60
Device(config)# netconf max-sessions 5
Device(config)# netconf max-message 2345
Device# ssh-2 -s username@10.1.1.1 netconf
```

次に、ループバック インターフェイス 113 の設定を取得する例を示します。

手順の概要

1. 最初に、「hello」を送信します。
2. 次に、get-config 要求を送信します。

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	最初に、「hello」を送信します。
	
	例：
	
	`<?xml version="1.0" encoding=\"UTF-8\"?>`
	`<hello><capabilities>`
	`<capability>u?rn:ietf:params:netconf:base:1.0</capability>`
	`<capability>urn:ietf:params:netconf:capability:writeable-running:1.0</capability>`
	`<capability>urn:ietf:params:netconf:ca?pability:roll?back-on-error:1.0</capability>`
	`<capability>urn:ietf:params:netconf:capability:startup:1.0</capability>`
	`<capability>urn:ietf:params:netconf:ca?pability:url:?1.0</capability>`
	`<capability>urn:cisco:params:netconf:capability:pi-data-model:1.0</capability>`
	`<capability>urn:cisco:params:netconf:capabili?ty:notificati?on:1.0</capability>`
	`</capabilities>`
	`</hello>]]>]]>`	

ステップ 2	次に、get-config 要求を送信します。
	
	例：
	
	`<?xml version="1.0"?>`
	`<rpc`
	`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"xmlns:cpi="http://www.cisco.com/cpi_10/schema"`
	
	`message-id="101">`
	`<get-config>`
	`<source>`
	`<running/>`
	`</source>`
	`<filter>`
	`<config-format-text-cmd>`
	`<text-filter-spec>`
	`interface Loopback113`
	`</text-filter-spec>`
	`</config-format-text-cmd>`
	`</filter>`
	`</get-config>`
	`</rpc>]]>]]>`	

デバイスに次の出力が表示されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101"xmlns=\"urn:ietf:params:netconf:base:1.0\">
   <data>
      <cli-config-data>
interface Loopback113
description test456
no ip address
load-interval 30
end
       </cli-config-data>
    </data>
</rpc-rep?ly>]]>]]>
```

NETCONF over SSHv2 に関する追加情報

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
NETCONF コマンド：コマンド構文、コマンドモード、コマンド履歴、デフォルト、使用に関する注意事項、および例	『*Cisco IOS Cisco Networking Services Command Reference*』
IP アクセス リスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例 セキュリティ コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例	『*Cisco IOS Security Command Reference*』
IP アクセス リスト	『Cisco IOS Security Configuration Guide: Securing the Data Plane』の「IP Access List Overview」および「Creating an IP Access List and Applying It to an Interface」の章
セキュア シェルおよびセキュア シェル バージョン 2	『Cisco IOS Security Configuration Guide: Securing User Services』の「Configuring Secure Shell」モジュール

標準および RFC

RFC	タイトル
RFC 2246	『*The TLS Protocol Version 1.0*』
RFC 4251	『*The Secure Shell (SSH) Protocol Architecture*』
RFC 4252	『*The Secure Shell (SSH) Authentication Protocol*』
RFC 4741	『NETCONF Configuration Protocol』
RFC 4742	『Using the NETCONF Configuration Protocol over Secure SHell (SSH)』

シスコのテクニカル サポート

説明	リンク
右の□□□にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。 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。	http://www.cisco.com/cisco/web/support/index.html

NETCONF over SSHv2 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 *8*：***NETCONF over SSHv2*** の機能情報

機能名	リリース	機能情報
NETCONF over SSHv2	Cisco IOS XE Release 2.1 12.2(33)SB 12.2(33)SRA 12.2(33)SXI 12.4(9)T	NETCONF over SSHv2 機能を使用すると、暗号化されたトランスポート上で Cisco コマンドライン インターフェイス（CLI）によるネットワーク設定を実行できます。 この機能により、次のコマンドが導入または変更されました。**netconflock-time**、**netconfmax-message**、**netconfmax-sessions**、**netconfssh**

第 9 章

BEEP による設定への NETCONF アクセス

ネットワーク設定プロトコル（NETCONF）overブロック拡張可能交換プロトコル（BEEP）機能を使用して、NETCONF 上で設定変更の通知を送信できます。通知は、設定変更が行われたことを示すイベントです。変更には、設定の追加、削除、または修正があります。通知は、設定操作の正常終了後に、一連の変更を示す1つのメッセージとして送信されます。変更された設定ごとに個別にメッセージを送信するわけではありません。

BEEP は、Simple Authentication and Security Layer（SASL）プロファイルを使用して既存のセキュリティ モデルに単純な直接マッピングを提供します。また、NETCONF over BEEP は、トランスポート層セキュリティ（TLS）を使用して、サーバ認証、またはサーバ側とクライアント側での認証のうち、いずれかの認証を行う強力な暗号化メカニズムを提供することもできます。

- 機能情報の確認，87 ページ
- BEEP による設定への NETCONF アクセスの前提条件，88 ページ
- BEEP による設定への NETCONF アクセスの制約事項，88 ページ
- BEEP による設定への NETCONF アクセスについて，88 ページ
- BEEP による設定への NETCONF アクセスの設定方法，90 ページ
- BEEP による設定への NETCONF アクセスの設定例，95 ページ
- BEEP による設定への NETCONF アクセスに関する追加情報，95 ページ
- BEEP による設定への NETCONF アクセスの機能情報，96 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

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。

BEEP による設定への NETCONF アクセスの前提条件

NETCONF over BEEP リスナーには、Simple Authentication and Security layer（SASL）を設定する必要があります。

BEEP による設定への NETCONF アクセスの制約事項

Transport Layer Security（TLS）を使用する BEEP を設定するには、暗号イメージを実行する必要があります。

BEEP による設定への NETCONF アクセスについて

NETCONF over BEEP の概要

BEEP による設定への NETCONF アクセス機能は、BEEP を転送プロトコルとしてイネーブルにして、NETCONF セッションで使用できるようにします。NETCONF over BEEP を使用すると、NETCONF サーバまたは NETCONF クライアントのいずれかが接続を開始するように設定できます。これによって、デバイスが断続的に接続された大規模ネットワークや、ファイアウォールおよびネットワーク アドレス変換（NAT）があるために管理接続を反転する必要のあるデバイスをサポートできます。

BEEP は、コネクション型非同期相互作用のための汎用アプリケーションプロトコルフレームワークです。これは、従来さまざまなプロトコルの実装で何度も利用されてきた機能を提供することを目的としています。BEEP は一般的に Transmission Control Protocol（TCP）上で動作し、メッセージの交換が可能です。HTTP および同様のプロトコルとは異なり、接続の両端でいつでもメッセージを送信できます。BEEP には暗号化と認証のファシリティも含まれており、高い拡張性があります。

BEEP プロトコルには、ピア同士が同時に独立してメッセージを交換できるフレーミング メカニズムが含まれています。通常これらのメッセージは XML を使用して構成されます。すべての交換は、転送セキュリティ、ユーザ認証、またはデータ交換などの明確に定義されたアプリケーション特性にバインドされたコンテキストで実行されます。このバインディングによってチャネルが形成されます。各チャネルには交換されるメッセージの構文およびセマンティクスを定義する関連付けられたプロファイルがあります。

BEEP セッションは NETCONF サービスにマップされます。セッションが確立されると、各 BEEP ピアは自身がサポートするプロファイルをアドバタイズします。チャネルの作成中に、クライアント（BEEP イニシエータ）はそのチャネルの 1 つまたは複数のプロファイルを提示します。サーバ（BEEP リスナー）がチャネルを作成する場合、サーバはいずれかのプロファイルを選択し、そ

のプロファイルを応答で送信します。サーバは、どのプロファイルも受け入れできないことを示し、チャネルの作成を断る場合もあります。

BEEP では、同時に複数のデータ交換チャネルを使用できます。

BEEP はピアツーピア プロトコルですが、特定のタイミングで実行している役割に応じて、各ピアにラベルが付けられます。BEEPセッションの確立時に、新規接続を待ち受けるピアがBEEPリスナーです。リスナーへの接続を確立するもう一方のピアが BEEP イニシエータになります。交換を開始する BEEP ピアがクライアントで、もう一方の BEEP ピアがサーバです。通常、サーバの役割を実行するBEEPピアは、リッスンする役割も実行します。ただし、BEEPはピアツーピアプロトコルであるからといって、サーバの役割を実行する BEEP ピアが、リッスンする役割も実行する必要はありません。

NETCONF over BEEP と SASL

SASL は、接続ベースのプロトコルに認証サポートを追加するためのインターネット標準方式です。SASL をセキュリティ アプライアンスと Lightweight Directory Access Protocol（LDAP）サーバとの間で使用してユーザ認証を保護できます。

BEEP リスナーには、SASL を設定する必要があります。

NETCONF over BEEP と TLS

TLS は、相互認証、完全性のためのハッシュの使用、プライバシー保護のための暗号化を可能にすることで、クライアントとサーバとの間にセキュアな通信を提供するアプリケーション レベルのプロトコルです。TLS では、証明書、公開キー、および秘密キーを使用します。

証明書はデジタル ID カードに似ています。この証明書は、クライアントに対してサーバの ID を証明します。各証明書には、発行した機関の名前、証明書の発行先エンティティの名前、エンティティの公開キー、および証明書の有効期限を示すタイム スタンプが含まれます。

公開キーおよび秘密キーは、情報の暗号化および復号化に使用される暗号キーです。公開キーは共有されますが、秘密キーは公開されることはありません。公開キーと秘密キーの各ペアは一緒に動作します。公開キーを使用して暗号化されたデータは、その秘密キーでのみ復号化できます。

NETCONF over BEEP とアクセス リスト

オプションで、NETCONF over SSHv2 セッション用のアクセス リストを設定できます。アクセス リストは、IP アドレスに対する許可および拒否の条件を順番に並べたものです。シスコ ソフトウェアは、アクセス リストの条件に対して、アドレスを 1 つずつテストします。最初の一致によって、ソフトウェアがアドレスを受け入れるか、拒否するかが決まります。最初に一致が見つかった後は条件のテストが終了するため、条件の順序は重要です。条件が一致しなければ、アドレスは拒否されます。

アクセス リストの使用に関連する 2 つの主要な作業は次のとおりです。

1. アクセス リストの番号または名前とアクセス条件を指定して、アクセス リストを作成する。
2. アクセス リストをインターフェイスまたは端末回線に適用する。

アクセス リストの設定の詳細については、『*Security Configuration Guide: Securing the Data Plane*』の「IP Access List Overview」および「Creating an IP Access List and Applying It to an Interface」モジュールを参照してください。

BEEP による設定への NETCONF アクセスの設定方法

SASL プロファイルの設定

SASL を使用して NETCONF over BEEP をイネーブルにするには、まず SASL プロファイルを設定する必要があります。SASL プロファイルは、デバイスへのアクセスが許可されるユーザを指定します。

手順の概要

1. **enable**
2. **configureterminal**
3. **saslprofile***profile-name*
4. **mechanismdigest-md5**
5. **server***user-name***password***password*
6. **exit**

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**saslprofile***profile-name* 例： `Device(config)# sasl profile beep`	SASL プロファイルを設定し、SASL プロファイル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**mechanismdigest-md5** 例： `Device(config-SASL-profile)# mechanism digest-md5`	SASL プロファイル メカニズムを設定します。
ステップ 5	**server***user-name***password***password* 例： `Device(config-SASL-profile)# server user1 password password1`	SASL サーバを設定します。
ステップ 6	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

NETCONF over BEEP のイネーブル化

はじめる前に

- 同時 NETCONF セッションと同じ数以上の vty 行が設定されている必要があります。
- SASL を使用する NETCONF over BEEP を設定するには、まず SASL プロファイルを設定する必要があります。

（注）
- 4 個以上の同時 NETCONF セッションを設定する必要があります。
- 最大 16 個の同時 NETCONF セッションを設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa general-keys**
4. **crypto pki trustpoint***name*
5. **enrollment url***url*
6. **subject-name***name*
7. **revocation-check***method1* [*method2* [*method3*]]
8. **exit**
9. **crypto pki authenticate***name*
10. **cryptopkienroll***name*
11. **netconf lock-time***seconds*
12. **linevty** line-number [*ending-line-number*]
13. **netconf max-sessions***session*
14. **netconfbeepinitiator**{*hostname* | *ip-address*} *port-number***user***sasl-user***password***sasl-password*[**encrypt***trustpoint*] [**reconnect-time***seconds*]
15. **netconfbeeplistener**[*port-number*] [**acl***access-list-number*] [**sasl***sasl-profile*] [**encrypt***trustpoint*]
16. **exit**

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 モードをイネーブルにします。 • 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	**crypto key generate rsa general-keys** 例： `Device(config)# crypto key generate rsa general-keys`	Rivest, Shamir, and Adelman（RSA）キーペアを生成し、汎用のキーペアを生成するように指定します。 この手順は一度だけ実行してください。

	コマンドまたはアクション	目的
ステップ 4	**crypto pki trustpoint***name* 例： `Device(config)# crypto pki trustpoint my_trustpoint`	ルータで使用するトラストポイントを宣言し、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 5	**enrollment url***url* 例： `Device(ca-trustpoint)# enrollment url http://10.2.3.3:80`	認証局（CA）の登録パラメータを指定します。
ステップ 6	**subject-name***name* 例： `Device(ca-trustpoint)# subject-name CN=dns_name_of_host.com`	証明書要求の所有者名を指定します。 （注） サブジェクト名は、デバイスのドメイン ネーム システム（DNS）名である必要があります。
ステップ 7	**revocation-check***method1* [*method2* [*method3*]] 例： `Device(ca-trustpoint)# revocation-check none`	証明書の失効ステータスをチェックします。
ステップ 8	**exit** 例： `Device(ca-trustpoint)# exit`	CA トラストポイント コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 9	**crypto pki authenticate***name* 例： `Device(config)# crypto pki authenticate my_trustpoint`	CA の証明書を取得して、認証局を認証します。
ステップ 10	**cryptopkienroll***name* 例： `Device(config)# crypto pki enroll my_trustpoint`	ルータの証明書を CA から取得します。
ステップ 11	**netconf lock-time***seconds* 例： `Device(config)# netconf lock-time 60`	（任意）NETCONF 設定を中間操作が行われないようにロックする最長時間を指定します。 seconds 引数の有効な値の範囲は 1 ～ 300 秒です。デフォルト値は 10 秒です。

	コマンドまたはアクション	目的
ステップ 12	**linevty** line-number [*ending-line-number*] 例： `Device(config)# line vty 0 15`	リモート コンソール アクセスの仮想端末回線を識別します。 NETCONF セッションの最大数と同じ数の vty 回線を設定する必要があります。
ステップ 13	**netconf max-sessions***session* 例： `Device(config)# netconf max-sessions 16`	（任意）許容される同時 NETCONF セッションの最大数を指定します。
ステップ 14	**netconfbeepinitiator**{*hostname* | *ip-address*} *port-number***user***sasl-user***password***sasl-password*[**encrypt***trustpoint*] [**reconnect-time***seconds*] 例： `Device(config)# netconf beep initiator host1 23 user user1 password password1 encrypt 23 reconnect-time 60`	（任意）BEEP を NETCONF セッションの転送プロトコルとして指定し、ピアを BEEP イニシエータとして設定します。 （注） この手順は、NETCONF BEEP イニシエータ セッションを設定する場合に実行します。任意で、BEEP リスナー セッションを設定することもできます。
ステップ 15	**netconfbeeplistener**[*port-number*] [**acl***access-list-number*] [**sasl***sasl-profile*] [**encrypt***trustpoint*] 例： `Device(config)# netconf beep listener 26 acl 101 sasl profile1 encrypt 25`	（任意）BEEP を NETCONF の転送プロトコルとして指定し、ピアを BEEP リスナーとして設定します。 （注） この手順は、NETCONF BEEP のリスナー セッションを設定する場合に実行します。任意で、BEEP イニシエータ セッションを設定することもできます。
ステップ 16	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

BEEP による設定への NETCONF アクセスの設定例

例：NETCONF over BEEP のイネーブル化

```
Device# configure terminal
Device(config)# crypto key generate rsa general-keys

Device(ca-trustpoint)# crypto pki trustpoint my_trustpoint

Device(ca-trustpoint)# enrollment url http://10.2.3.3:80
Device(ca-trustpoint)# subject-name CN=dns_name_of_host.com
Device(ca-trustpoint)# revocation-check none
Device(ca-trustpoint)# crypto pki authenticate my_trustpoint

Device(ca-trustpoint)# crypto pki enroll my_trustpoint

Device(ca-trustpoint)# line vty 0 15

Device(ca-trustpoint)# exit
Device(config)# netconf lock-time 60

Device(config)# netconf max-sessions 16

Device(config)# netconf beep initiator host1 23 user my_user password my_password encrypt
my_trustpoint reconnect-time 60

Device(config)# netconf beep listener 23 sasl user1 encrypt my_trustpoint
```

BEEP による設定への NETCONF アクセスに関する追加情報

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
NETCONF コマンド：コマンド構文、コマンドモード、コマンド履歴、デフォルト、使用に関する注意事項、および例	『*Cisco IOS Cisco Networking Services Command Reference*』

標準および RFC

標準/RFC	タイトル
RFC 2222	『*Simple Authentication and Security Layer (SASL)*』
RFC 3080	『*The Blocks Extensible Exchange Protocol Core*』

標準/RFC	タイトル
RFC 4741	『*NETCONF Configuration Protocol*』
RFC 4744	『*Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)*』

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。 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。	http://www.cisco.com/cisco/web/support/index.html

BEEP による設定への NETCONF アクセスの機能情報

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 *9*：*BEEP* による設定への *NETCONF* アクセスの機能情報

機能名	リリース	機能情報
BEEP による設定への NETCONF アクセス	Cisco IOS XE Release 2.1 12.2(33)SB 12.2(33)SRB 12.2(33)SXI 12.4(9)T	NETCONF over BEEP 機能を使用すると、NETCONF サーバまたはNETCONF クライアントのどちらかが接続を開始するように設定できます。これによって、デバイスが断続的に接続された大規模ネットワークや、ファイアウォールおよび Network Address Translators（NAT）があるために管理接続を反転する必要のあるデバイスをサポートできます。 この機能により、**netconfbeepinitiator** および **netconfbeeplistener** コマンドが導入または変更されました。