



ネットワークモデルの構成 : Cisco WAE CLI

ここでは、次の内容について説明します。

- [WAE CLI の概要 \(1 ページ\)](#)
- [エキスパートモードと WAE CLI の比較 \(12 ページ\)](#)
- [WAE CLI を使用したネットワークモデルの構成 \(14 ページ\)](#)

WAE CLI の概要

WAE は、WAE YANG ファイルで記述されたデータモデルを使用して自動的にレンダリングされるネットワーク CLI を提供します。CLI には、ネットワーク構成を操作するためのコマンドが含まれています。CLI は完全にデータモデル駆動型です。YANG モデルは構成要素の階層を定義します。CLI はこのツリーに従います。CLI には、管理対象デバイスのハードウェアおよびネットワーク接続を構成するためのさまざまなコマンドが用意されています。

CLI は 2 つのモードをサポートしています。WAE ノードの状態を監視するための動作モードと、ネットワークの状態を変更するための構成モードです。プロンプトは、CLI がどのモードにあるかを示します。**configure** コマンドを使用して動作モードから構成モードに移行すると、プロンプトが **user@wae#** から **user@wae(config)#** に変化します。プロンプトは、wae.conf ファイルの c-prompt1 および c-prompt2 設定を使用して構成できます。

次に例を示します。

```
admin@wae# configure
Entering configuration mode terminal
admin@wae(config)#
```

動作モード

動作モードは、CLI へのログインに成功した後の初期モードです。これは主に、システムステータスの表示、CLI 環境の制御、ネットワーク接続の監視とトラブルシューティング、および構成モードの開始に使用されます。

次のコマンドは、動作モードで使用できる基本コマンドです。追加のコマンドは、ロードされた YANG ファイルからレンダリングされます。

アクションを呼び出します。

<path> <parameters>

指定された *parameters* を使用して、*path* で見つかったアクションを呼び出します。このコマンドは YANG ファイルから自動生成されます。たとえば、YANG ファイルに次のアクション指定があるとします。

```
tailf:action shutdown {
  tailf:actionpoint actions;
  input {
    tailf:constant-leaf flags {
      type uint64 {
        range "1 .. max";
      }
      tailf:constant-value 42;
    }
    leaf timeout {
      type xs:duration;
      default PT60S;
    }
    leaf message {
      type string;
    }
    container options {
      leaf rebootAfterShutdown {
        type boolean;
        default false;
      }
      leaf forceFsckAfterReboot {
        type boolean;
        default false;
      }
      leaf powerOffAfterShutdown {
        type boolean;
        default true;
      }
    }
  }
}
```

このアクションは、次の方法で呼び出すことができます。

```
user@wae> shutdown timeout 10s message reboot options { \
forceFsckAfterReboot true }
```

組み込みの動作モードコマンド

コマンド	説明
commit (abort confirm)	<p>保留中のコミット確認を中止または確認します。commit confirm を実行せずに CLI セッションが終了した場合、保留中のコミット確認は中止されます。デフォルトは confirm です。例：</p> <pre>user@wae# commit abort</pre>

config (exclusive terminal) [no-confirm]	<p>構成モードを開始します。デフォルトは terminal です。</p> <ul style="list-style-type: none"> • terminal : 実行構成のプライベートコピーを編集します。ロックはかかりません。 • no-confirm : 確認ダイアログを無視して、構成モードを開始します。 <p>例 :</p> <pre>user@wae# config terminal Entering configuration mode terminal</pre>
file list <directory>	<p>ディレクトリ内のファイルを一覧表示します。例 :</p> <pre>user@wae# file list /config rollback10001 rollback10002 rollback10003 rollback10004 rollback10005</pre>
file show <file>	<p>ファイルの内容を表示します。例 :</p> <pre>user@wae# file show /etc/skel/.bash_profile # /etc/skel/.bash_profile # This file is sourced by bash for login shells. The following line # runs our .bashrc and is recommended by the bash info pages. [[-f ~/.bashrc]] && . ~/.bashrc</pre>
help <command>	<p>コマンドのヘルプテキストが表示されます。例 :</p> <pre>user@wae# help job Help for command: job Job operations</pre>
job stop <job id>	<p>特定のバックグラウンドジョブを停止します。デフォルトの CLI では、バックグラウンドジョブを作成するコマンドは monitor start だけです。例 :</p> <pre>user@wae# monitor start /var/log/messages [ok][...] admin@ncs# show jobs JOB COMMAND 3 monitor start /var/log/messages admin@ncs# job stop 3 admin@ncs# show jobs JOB COMMAND</pre>

<p>logout session <session ID></p>	<p>WAE から特定のユーザーセッションをログアウトします。ユーザーが構成排他ロックを保持している場合、ロックは解放されます。例：</p> <pre> user@wae# who Session User Context From Proto Date Mode 25 oper cli 192.0.2.254 ssh 12:10:40 operational *24 admin cli 192.0.2.254 ssh 12:05:50 operational user@wae# logout session 25 user@wae# who Session User Context From Proto Date Mode *24 admin cli 192.0.2.254 ssh 12:05:50 operational </pre>
<p>logout user <username></p>	<p>WAE から特定のユーザーをログアウトします。ユーザーが構成排他ロックを保持している場合、ロックは解放されます。例：</p> <pre> user@wae# who Session User Context From Proto Date Mode 25 oper cli 192.0.2.254 ssh 12:10:40 operational *24 admin cli 192.0.2.254 ssh 12:05:50 operational user@wae# logout user oper user@wae# who Session User Context From Proto Date Mode *24 admin cli 192.0.2.254 ssh 12:05:50 operational </pre>
<p>script reload</p>	<p>scripts/command ディレクトリにあるスクリプトをリロードします。新しいスクリプトが追加されます。スクリプトファイルが削除されている場合、対応する CLI コマンドは消去されます。</p>
<p>send (all <user>) <message></p>	<p>デバイスにログインしているすべてのユーザーの画面または特定の画面にメッセージを表示します。</p> <ul style="list-style-type: none"> • all : 現在ログインしているすべてのユーザーにメッセージを表示します。 • <user> : 特定のユーザーにメッセージを表示します。 <p>例：</p> <pre> user@wae# send oper "I will reboot system in 5 minutes." oper ユーザーの画面には、次のメッセージが表示されます。 oper@wae# Message from user@wae at 13:16:41... I will reboot system in 5 minutes. EOF </pre>

show cli	<p>CLI プロパティを表示します。例：</p> <pre> user@wae# show cli autowizard false complete-on-space true display-level 99999999 history 100 idle-timeout 1800 ignore-leading-space false output-file terminal paginate true prompt1 \h\M# prompt2 \h(\m)# screen-length 71 screen-width 80 service prompt config true show-defaults false terminal xterm-256color timestamp disable </pre>
show history [<i><limit></i>]	<p>CLI コマンドの履歴を表示します。デフォルトでは、最新の 100 個のコマンドが一覧表示されます。履歴リストのサイズは、history CLI 設定を使用して構成されます。履歴制限が指定されている場合、その制限までの最新のコマンドのみが表示されます。例：</p> <pre> user@wae# show history 06-19 14:34:02 -- ping router 06-20 14:42:35 -- show running-config 06-20 14:42:37 -- who 06-20 14:42:40 -- show history user@wae# show history 3 14:42:37 -- who 14:42:40 -- show history 14:42:46 -- show history 3 </pre>
show jobs	<p>現在バックグラウンドで実行中のジョブを表示します。例：</p> <pre> user@wae# show jobs JOB COMMAND 3 monitor start /var/log/messages </pre>
show log <i><file></i>	<p>ログファイルの内容を表示します。例：</p> <pre> user@wae# show log messages </pre>
show parser dump <i><command prefix></i>	<p>指定されたコマンドプレフィックスで始まるすべての可能なコマンドを表示します。</p>

<p>show running-config [<path filter> [sort-by <idx>]]</p>	<p>現在の設定を表示します。デフォルトでは、構成全体が表示されます。パスフィルタを指定することで、表示される内容を制限できます。パスフィルタでは、特定のインスタンスを指すパスを使用することも、またはインスタンスIDを省略した場合は、省略されたインスタンス以降の部分をフィルタとして扱うこともできます。</p> <p>sort-by 引数は、パスフィルタがセカンダリインデックスを持つリスト要素を指す場合に使用できます。セカンダリインデックスの名前は <i>idx</i> です。使用すると、テーブルはセカンダリインデックスで定義された順序で並べ替えられます。これにより、インスタンスを表示する順序を制御できます。</p> <p>たとえば、管理者ユーザーの aaa 設定を表示するには：</p> <pre>user@wae# show running-config aaa authentication users user admin aaa authentication users user admin uid 1000 gid 1000 password \$1\$JA.103Tx\$ZtlycpnMlg1bVMqM/zSZ7/ ssh_keydir /var/ncs/homes/admin/.ssh homedir /var/ncs/homes/admin !</pre> <p>グループ ID 1000 のすべてのユーザーを表示し、ユーザー ID を省略して代わりに gid 1000 を指定するには：</p> <pre>user@wae# show running-config aaa authentication users user * gid 1000 ...</pre>
<p>show <path> [sort-by <idx>]</p>	<p>パスがリスト要素につながり、データをテーブルとしてレンダリングできる（つまり、テーブルが画面に収まる）場合、構成をテーブルとして表示します。 tab パイプコマンドを使用して、リストのテーブルフォーマットを強制することもできます。</p> <p>sort-by 引数は、パスがセカンダリインデックスを持つリスト要素を指す場合に使用できます。セカンダリインデックスの名前は <i>idx</i> です。使用すると、テーブルはセカンダリインデックスで定義された順序で並べ替えられます。これにより、インスタンスを表示する順序を制御できます。例：</p> <pre>user@wae# show devices device-module NAME REVISION URI DEVICES ----- junos - http://xml.juniper.net/xnm/1.1/xnm [pe2] tailf-ned-cisco-ios - urn:ios [ce1 ce0] tailf-ned-cisco-ios-stats - urn:ios-stats [ce1 ce0] tailf-ned-cisco-ios-xr - http://tail-f.com/ned/cisco-ios-xr [p1 p0]</pre>
<p>source <file></p>	<p>ユーザーが入力したかのように、指定されたファイルからコマンドを実行します。ファイルからコマンドを実行する場合、autowizard は無効になります。</p>

timecmd <command>	<p>コマンドの実行時間を測定して表示します。timecmd は、CLIセッション設定で devtools が true に設定されている場合にのみ使用できることに注意してください。例：</p> <pre>user@wae# timecmd id user = admin(501), gid=20, groups=admin, gids=12,20,33,61,79,80,81,98,100 Command executed in 0.00 sec user@wae#</pre>
who	<p>現在、ログオンしているユーザーを表示します。現在のセッション (show status コマンドを実行しているセッション) には、アスタリスクが付いています。例：</p> <pre>user@wae# who Session User Context From Proto Date Mode 25 oper cli 192.0.2.254 ssh 12:10:40 operational *24 admin cli 192.0.2.254 ssh 12:05:50 operational admin@ncs#</pre>

構成モード

構成モードは、動作モードで **configure** コマンドを入力することで開始できます。ネットワーク構成に対するすべての変更は、アクティブな構成のコピーに対して行われます。これらの変更は、コミットまたはコミット確認コマンドが正常に入力されるまで有効になりません。

次のコマンドは、構成モードで使用できる基本コマンドです。追加のコマンドは、ロードされた YANG ファイルからレンダリングされます。

値を構成します。

<path> [**<value>**]

パラメータを設定します。新しい識別子が作成され、**autowizard** が **enabled** の場合、CLI は、その識別子のすべての必須サブ要素についてユーザーにプロンプトを表示します。このコマンドは YANG ファイルから自動生成されます。

<value> が提供されていない場合、CLI はその値についてユーザーにプロンプトを表示します。<path> が **tailf-common.yang** データモデルで記述されている **MD5DigestString**、**DESDigestString**、**DES3CBCEncryptedString**、または **AESCFB128EncryptedString** 型の暗号化された値である場合、入力された値のエコーは発生しません。

組み込みの構成モードコマンド

コマンド	説明
annotate <statement> <text>	<p>注釈を特定の構成に関連付けます。注釈を削除するには、テキストを空のままにします。このコマンドは、システムが属性を有効にして構成されている場合にのみ使用できます。</p>

commit (check and-quit confirmed to-startup) [comment <text>] [label <text>]	<p>現在の構成をコミットして running にします。</p> <ul style="list-style-type: none"> • check : 現在の構成を検証します。 • and-quit : コミットして running にし、構成モードを終了します。 • comment <text> : コメントをコミットに関連付けます。コメントは、ロールバックファイルを調べるときに表示されます。 • label <text> : ラベルをコミットに関連付けます。ラベルは、ロールバックファイルを調べるときに表示されます。 <p>(注) 便利なコマンドは、commit dry-run です。このコマンドは、構成の変更を検証して表示しますが、実際のコミットは実行しません。使用可能な commit コマンドの詳細については、コミットフラグを参照してください。</p>
copy <instance path> <new id>	<p>インスタンスのコピーを作成します。</p>
copy cfg [merge overwrite] <src path> to <dest path>	<p>ある構成ツリーから別の構成ツリーにデータをコピーします。接続先で意味のあるデータのみがコピーされます。コピーできないデータについてはエラーメッセージは生成されず、操作はエラーメッセージが生成されずに完全に失敗する可能性があります。たとえば、デバイス構成の一部からテンプレートを作成するには、最初にデバイスを構成してから、構成をテンプレート構成ツリーにコピーします。例：</p> <pre> user@wae(config)# devices template host_temp user@wae(config-template-host_temp)# exit user@wae(config)# copy cfg merge devices device ce0 config \ ios:ethernet to devices template host_temp config ios:ethernet user@wae(config)# show configuration diff +devices template host_temp + config + ios:ethernet cfm global + ! +!</pre>
copy compare <src path> to <dest path>	<p>2つの任意の構成ツリーを比較します。送信元ツリーにのみ表示される項目は無視されます。</p>
delete <path>	<p>データ要素を削除します。</p>
do <command>	<p>コマンドを動作モードで実行します。</p>
edit <path>	<p>サブ要素を編集します。パスに欠落している要素が作成されます。</p>
exit (level configuration-mode)	<ul style="list-style-type: none"> • level : このレベルを終了します。トップレベルで実行すると、構成モードを終了します。これは、オプションが指定されていない場合のデフォルトです。 • configuration-mode : 編集レベルに関係なく構成モードを終了します。
help <command>	<p>コマンドのヘルプテキストを表示します。</p>

hide <hide-group>	非表示グループに属する要素とアクションを再度非表示にします。非表示にするのにパスワードは必要ありません。このコマンドは非表示であり、コマンドの完了時に表示されません。
insert <path>	新しい要素を挿入します。要素がすでに存在し、データモデルに <code>indexedView</code> オプションが設定されている場合、古い要素の名前が <code>element+1</code> に変更され、新しい要素がその場所に挿入されます。
insert <path>[first last before <key> after <key>]	順序付きリストに新しい要素を挿入します。要素は、先頭、末尾（デフォルト）、別の要素の前または後に追加できます。
load (merge override) (terminal <file>)	<p>ファイルまたは端末から構成をロードします。</p> <ul style="list-style-type: none"> • merge : ファイルまたは端末の内容を現在の構成とマージします。 • override : 現在の構成をファイルまたは端末からの構成で置き換えます。 <p>たとえば、現在の構成が次のようであるとします。</p> <pre> devices device pl config cisco-ios-xr:interface GigabitEthernet 0/0/0/0 shutdown exit cisco-ios-xr:interface GigabitEthernet 0/0/0/1 shutdown ! !</pre> <p>エントリ <code>GigabitEthernet 0/0/0/0</code> の shutdown 値が削除されます。構成ファイルは、間にコメントが挿入された一連のコマンドであるため、構成ファイルは次のようになります。</p> <pre> devices device pl config cisco-ios-xr:interface GigabitEthernet 0/0/0/0 no shutdown exit ! !</pre> <p>その後、このファイルをコマンド load merge <i>FILENAME</i> で使用して、目的の結果を得ることができます。</p>
move <path>[first last before <key> after <key>]	既存の要素を順序付きリストの新しい位置に移動します。要素は、先頭、末尾（デフォルト）、別の要素の前または後に移動できます。
rename <instance path> <new id>	インスタンスの名前を変更します。
revert	実行構成を現在の構成にコピーし、コミットされていないすべての変更を削除します。

rload(merge override) (terminal <file>)	<p>現在のサブモードに関連するファイルをロードします。たとえば、ファイルにデバイス構成がある場合、1つのデバイスに入り、rload merge/override <file> コマンドを発行してそのデバイスの構成をロードし、次に別のデバイスに入り、rload を使用して同じ構成ファイルをロードできます。load コマンドも参照してください。</p> <ul style="list-style-type: none"> • merge : ファイルまたは端末の内容を現在の構成とマージします。 • override : 現在の構成をファイルまたは端末からの構成で置き換えます。
rollback configuration [<number>][<path>]	<p>構成を以前にコミットされた構成に戻します。wae.confファイルに保存する古い構成の数を構成できます。保存する構成がしきい値を超えると、新しい構成を作成する前に最も古い構成が削除されます。構成の変更はロールバックファイルに保存され、最新の変更は最大の番号 N を持つファイル rollbackN に保存されます。</p> <p>デルタのみがロールバックファイルに保存されます。構成をロールバック N にロールバックすると、rollback10001 ~ rollbackN に保存されているすべての変更が適用されます。オプションの path 引数を使用すると、構成ツリーの残りの部分を変更せずに、サブツリーをロールバックできます。</p> <p>このコマンドは、wae.confでロールバックが有効になっている場合にのみ使用できます。 例 :</p> <pre>user@wae (config) # rollback configuration 10005</pre>
rollback selective [<number>][<path>]	<p>rollback10001 から rollbackN までのすべての変更を元に戻す代わりに、特定のロールバックファイルに保存されている変更のみを元に戻すことができます。場合によっては、ロールバックファイルの適用に失敗したり、構成を有効にするために追加の変更が必要になったりすることがあります。</p> <p>オプションの path 引数を使用すると、構成ツリーの残りの部分を変更せずに、サブツリーをロールバックできます。</p>
show full-configuration [<pathfilter> [sort-by <idx>]]	<p>ローカルの変更を考慮して、現在の構成を表示します。パスフィルタを指定することにより、show コマンドを構成の一部に制限できます。sort-by 引数は、パスフィルタがセカンダリインデックスを持つリスト要素を指す場合に指定できます。セカンダリインデックスの名前は idx です。使用すると、テーブルはセカンダリインデックスで定義された順序で並べ替えられます。これにより、インスタンスを表示する順序を制御できます。</p>
show configuration [<pathfilter>]	<p>構成に対する現在の編集を表示します。</p>
show configuration merge [<pathfilter> [sort-by <idx>]]	<p>ローカルの変更を考慮して、現在の構成を表示します。パスフィルタを指定することにより、show コマンドを構成の一部に制限できます。sort-by 引数は、パスフィルタがセカンダリインデックスを持つリスト要素を指す場合に指定できます。セカンダリインデックスの名前は idx です。使用すると、テーブルはセカンダリインデックスで定義された順序で並べ替えられます。これにより、インスタンスを表示する順序を制御できます。</p>

show configuration commit changes [<i><number></i> [<i><path></i>]]	コミットに対して作成されたロールバック番号によって識別される、そのコミットに関連付けられた編集を表示します。変更を元に戻すためのコマンドを表示する show configuration rollback changes とは対照的に、変更は前方変更として表示されます。オプションの <i>path</i> 引数を使用すると、特定のサブツリーに関連する編集のみをリストできます。
show configuration commit list [<i><path></i>]	ロールバックファイルをリストします。オプションの <i>path</i> 引数を使用すると、特定のサブツリーに関連するロールバックファイルのみをリストできます。
show configuration rollback listed [<i><number></i>]	ロールバックファイルに関連付けられたコミットで実行された変更を元に戻すために必要な操作を表示します。これらは、構成がそのロールバック番号にロールバックされた場合に適用される変更です。
show configuration running [<i><pathfilter></i>]	コミットされていない変更を考慮せずに実行構成を表示します。オプションのパスフィルタを指定して、表示する内容を制限できます。
show configuration diff [<i><pathfilter></i>]	追加および削除された構成行の前に + と - を付けて、実行構成に対するコミットされていない変更を差分形式で表示します。
show parser dump <i><command prefix></i>	コマンドプレフィックスで始まるすべての可能なコマンドを表示します。
tag add <i><statement></i> <i><tag></i>	構成ステートメントにタグを追加します。このコマンドは、システムが属性を有効にして構成されている場合にのみ使用できます。
tag del <i><statement></i> <i><tag></i>	構成ステートメントからタグを削除します。このコマンドは、システムが属性を有効にして構成されている場合にのみ使用できます。
tag clear <i><statement></i>	構成ステートメントからすべてのタグを削除します。このコマンドは、システムが属性を有効にして構成されている場合にのみ使用できます。
timecmd <i><command></i>	コマンドの実行時間を測定して表示します。このコマンドは、CLI セッション設定で <i>devtools</i> が <i>true</i> に設定されている場合にのみ使用できます。例： <pre>user@wae# timecmd id user = admin(501), gid=20, groups=admin, gids=12,20,33,61,79,80,81,98,100 Command executed in 0.00 sec user@wae#</pre>
top [<i><command></i>]	構成の最上位に戻るか、構成の最上位でコマンドを実行します。
unhide <i><hide-group></i>	非表示グループに属するすべての要素とアクションを再表示します。パスワードが必要な場合があります。このコマンドは非表示であり、コマンドの完了時に表示されません。
validate	現在の構成を検証します。 commit check と同じ動作です。

xpath [ctx <path>] (eval must when) <expression>	<p>XPath 式を評価します。context-path は、式の評価のための現在のコンテキストとして使用できます。context-path が指定されていない場合、現在のサブモードが context-path として使用されます。パイプコマンドトレースを使用して、デバッグ情報またはトレース情報を表示できます。このコマンドは、CLI セッション設定で devtools が true に設定されている場合にのみ使用できます。</p> <ul style="list-style-type: none"> • eval : XPath 式を評価します。 • must : 式を YANG <i>must</i> 式として評価します。 • when : 式を YANG <i>when</i> 式として評価します。
--	--

エキスパートモードと WAE CLI の比較

このガイドでは、エキスパートモードを使用した多くの構成について説明していますが、エキスパートモードと CLI インターフェイスを互換的に使用できることに注意することが重要です。GUI ではなくエキスパートモードを使用する利点は、構成に使用可能なすべてのフィールドが表示されることです。CLI で、使用可能なすべてのオプションを表示するには、パラメータを把握しているか、CLI コマンドのヘルプを表示する必要があります。

CLI での構成は、エキスパートモードでのナビゲートと同じパス構造に従います。次の表に、サンプルデータを使用して同等である CLI コマンドとエキスパートモード構成を示します。

設定の種類	エキスパートモード	CLI での同等コマンド
デバイスログイン情報を使用してデバイス認証グループを作成します。	<ol style="list-style-type: none"> 1. エキスパートモードから、[設定エディタ (Configuration editor)] で、[/ncs:devices] に移動し、[認証グループ (authgroups)] タブをクリックします。 2. [group] をクリックします。 3. プラス ([+]) 記号をクリックし、認証グループ名として groupABC を入力して、[追加 (Add)] をクリックします。 4. [default-map] をクリックし、次の認証パラメータを入力します。 [remote-name] : rpcl、 [remote-password] : XLydrf、 [remote-secondary-password] : XLydr。 	<pre># devices authgroups group groupABC default-map remote-name rpcl remote-password XLydrf remote-secondary-password XLydr</pre>

設定の種類	エキスパートモード	CLI での同等コマンド														
<p>XTC (topo-bgpls-xtc-nimo) を使用してネットワークを検出して、ネットワークモデルを作成します。</p> <p>(注) この例では、ネットワークアクセスと XTC エージェントが構成され、実行されていることを前提としています。</p>	<ol style="list-style-type: none"> エキスパートモードから、[設定エディタ (Configuration editor)] で、[/wae:networks] に移動し、プラス ([+]) 記号をクリックして、as54001_topo と入力します。 [追加 (Add)] をクリックします。 [nimo] タブをクリックし、NIMO タイプとして [topo-bgpls-xtc-nimo] を選択します。 次を入力します。 <table border="1" data-bbox="641 739 1073 1184"> <thead> <tr> <th>フィールド</th> <th>ユーザ入力 (User Input)</th> </tr> </thead> <tbody> <tr> <td>network-access</td> <td>as54001</td> </tr> <tr> <td>xtc-host</td> <td>xt11</td> </tr> <tr> <td>backup-xtc-host</td> <td>xt12</td> </tr> <tr> <td>asn</td> <td>54001</td> </tr> <tr> <td>igp-protocol</td> <td>isis</td> </tr> <tr> <td>extended-topology-discovery</td> <td>true</td> </tr> </tbody> </table>	フィールド	ユーザ入力 (User Input)	network-access	as54001	xtc-host	xt11	backup-xtc-host	xt12	asn	54001	igp-protocol	isis	extended-topology-discovery	true	<pre># networks network as54001_topo nimo topo-bgpls-xtc-nimo network-access as54001 xtc-host xtc11 backup-xtc-host xtc12 igp-protocol isis extended-topology-discovery true asn 54001</pre>
フィールド	ユーザ入力 (User Input)															
network-access	as54001															
xtc-host	xt11															
backup-xtc-host	xt12															
asn	54001															
igp-protocol	isis															
extended-topology-discovery	true															
<p>ネットワークモデルを統合します。</p>	<ol style="list-style-type: none"> エキスパートモードから、[設定エディタ (Configuration editor)] で、[/wae:networks] に移動し、プラス ([+]) 記号をクリックして、as54001 と入力します。 [追加 (Add)] をクリックします。 [nimo] タブをクリックし、[aggregator] を選択します。 [aggregator] > プラス ([+]) 記号をクリックし、送信元 NIMO として [as54001_topo]、[as54001_xtclsp]、[as54001_confisp]、および [as54001_snmplsp] を選択します。 	<pre># networks network as54001 nimo aggregator sources as54001_topo # networks network as54001 nimo aggregator sources as54001_xtclsp # networks network as54001 nimo aggregator sources as54001_confisp # networks network as54001 nimo aggregator sources as54001_snmplsp</pre>														

WAE CLI を使用したネットワークモデルの構成

このワークフローでは、エキスパートモードを使用してネットワークモデルを作成する構成手順の概要について説明します。

ステップ	詳細
1. デバイス認証グループと SNMP グループを構成します。	CLI を使用したデバイスアクセスの構成 (14 ページ)
2. ネットワーク アクセスプロファイルを構成します。	ネットワーク アクセスプロファイルの構成 (15 ページ)
3. エージェントを構成します。 (注) この手順は、XTC またはマルチレイヤ情報を収集する場合にのみ必要です。	<ul style="list-style-type: none"> • エキスパートモードを使用した XTC エージェントの構成 • エキスパートモードを使用した構成解析エージェントの構成
4. ネットワークを作成し、基本的なトポロジデータを収集します。	ネットワークモデルの作成 (16 ページ)
5. 追加のデータ収集または NIMO 機能を構成します。	追加 NIMO の構成 (18 ページ)
6. (オプション) スケジューラを構成します。	スケジューラ構成
7. プランアーカイブを構成して表示します。	WAE CLI を使用したアーカイブの構成 (18 ページ)

CLI を使用したデバイスアクセスの構成

WAE は、デバイスへのログインおよび SNMP アクセスに認証グループを使用します。次の手順では、構成モードで CLI を使用してデバイスアクセスを構成する方法について説明します。

ステップ 1 デバイスログイン情報を使用してデバイス認証グループを作成します。

```
# devices authgroups group <group_name>
# default-map remote-name <username>
# default-map remote-password <user_password>
# default-map remote-secondary-password <secondary_password>
# commit
```

ステップ 2 SNMP ツールを実行できるように SNMP グループを作成します。

```
# devices authgroups snmp-group <group_name>
# default-map community-name <community_name>
# commit
```

SNMPv3 の場合、次のオプションを設定できます。

```
# devices authgroups snmp-group <group_name>
# default-map community-name <community_name>
# default-map usm remote-name <remote_user>
# default-map usm security-level <auth-priv or auth-no-priv or no-auth-no-priv>
# default-map usm auth <auth_protocol> remote-password <remote_password>
# default-map usm priv <priv_protocol> remote-password <remote_password>
# commit
```

(注) no-auth-no-priv を使用して認証と暗号化を選択するオプションがありますが、これらの値はバックエンドでは使用されず、オプションです。

例

例 (デモンストレーションの目的で単純な名前とパスワードを使用) :

```
user@wae(config)# devices authgroups group ABCgroup default-map remote-name anyuser
remote-password password123 remote-secondary-password mypassword
user@wae(config)# commit
```

```
user@wae(config)# devices authgroups snmp-group snmp_v2 default-map community-name
mycompany
user@wae(config)# commit
```

```
user@wae(config)# devices authgroups snmp-group snmp_v3_01
user@wae(config)# default-map community-name mycompany
user@wae(config)# default-map usm remote-name User1
user@wae(config)# default-map usm security-level auth-priv
user@wae(config)# default-map usm auth md5 remote-password pass_a123
user@wae(config)# default-map usm priv aes remote-password pass_a123
user@wae(config)# commit
```

```
user@wae(config)# devices authgroups snmp-group snmp_v3_02
user@wae(config)# default-map community-name mycompany
user@wae(config)# default-map usm remote-name User2
user@wae(config)# default-map usm security-level auth-no-priv
user@wae(config)# default-map usm auth sha remote-password pass_b456
user@wae(config)# commit
```

```
user@wae(config)# devices authgroups snmp-group snmp_v3_03
user@wae(config)# default-map community-name mycompany
user@wae(config)# default-map usm remote-name User2
user@wae(config)# default-map usm security-level no-auth-no-priv
user@wae(config)# commit
```

ネットワーク アクセス プロファイルの構成

始める前に

認証およびSNMPグループが構成されていることを確認します。詳細については、「[CLIを使用したデバイスアクセスの構成 \(14 ページ\)](#)」を参照してください。

ステップ 1 次のコマンドを入力します。

```
# wae nimos network-access network-access <network-access-ID> auth-group <auth-group-ID>
# wae nimos network-access network-access <network-access-ID> snmp-group <snmp-group-ID>
```

ステップ 2 次のコマンドを繰り返して、各管理 IP アドレスを入力します。

```
# wae nimos network-access network-access <network-access-IP> node-access <node-access-ID-1> auth-group
<auth_group_ID> default-snmp-group <snmp-group-ID>
ip-manage <ip-address-1>
```

(注) ノードフィルタは ip-manage では機能しません。

ステップ 3 設定をコミットします。

```
# commit
```

例

次に例を示します。

```
# wae nimos network-access network-access as64001 auth-group ABCgroup
# wae nimos network-access network-access as64001 snmp-group snmp_v3_01
# wae nimos network-access network-access as64001 node-access 10.1.1.1 ip-manage
10.18.20.121
# wae nimos network-access network-access as64001 node-access 10.2.2.2 ip-manage
10.18.20.122
# commit

# wae nimos network-access network-access netaccess_01 auth-group ABCgroup
# wae nimos network-access network-access netaccess_01 snmp-group snmp_v2
node-access 122.168.200.2 ip-manage 192.18.20.2
```

ネットワークモデルの作成

ネットワークを作成するときは、`topo-igp-nimo` または `topo-bgpls-xtc-nimo` を使用して基本的なトポロジ収集も構成する必要があります。詳細については、[IGP データベースを使用したトポロジ収集](#) および [XTC を使用したトポロジ収集](#) を参照してください。



(注) 既存のプランファイルをロードしてネットワークモデルを作成するオプションがあります。[プランファイルのロード \(17 ページ\)](#) を参照してください。

始める前に

- デバイスアクセスとネットワークアクセスを構成する必要があります。
- XTC を実行するネットワークを作成する場合は、XTC エージェントが構成されていることを確認します。詳細については、[エキスパートモードを使用した XTC エージェントの構成](#) を参照してください。

次のコマンドを入力します。

```
# networks network <topo-network-model-name> nimo <NIMO-name>
network-access <network-access> <parameter-1>
<parameter-1-option> <parameter-2> <parameter-2-option>
<parameter-x> <parameter-x-option>
# commit
```

例

次の例は、`topo-bgpls-xtc-nimo` を設定する 2 つの方法を示しています。

例 1 :

```
# networks network NetworkABC_topo-bgpls-xtc-nimo nimo topo-bgpls-xtc-nimo network-access
TTE_lab_access
# networks network NetworkABC_topo-bgpls-xtc-nimo nimo topo-bgpls-xtc-nimo xtc-host
TTE-xtc11
# networks network NetworkABC_topo-bgpls-xtc-nimo nimo topo-bgpls-xtc-nimo backup-xtc-host
xtc12
# networks network NetworkABC_topo-bgpls-xtc-nimo nimo topo-bgpls-xtc-nimo asn 62001
# networks network NetworkABC_topo-bgpls-xtc-nimo nimo topo-bgpls-xtc-nimo igp-protocol
isis
# networks network NetworkABC_topo-bgpls-xtc-nimo nimo topo-bgpls-xtc-nimo
extended-topology-discovery true
```

例 2 :

```
# networks network NetworkABC_topo-bgpls-xtc-nimo nimo topo-bgpls-xtc-nimo
network-access TTE_lab xtc-host TTE-xtc11
backup-xtc-host TTE-xtc12 igp-protocol isis
extended-topology-discovery true asn 62001
```

次のタスク

このネットワークモデルをソースネットワークとして使用して、追加のネットワークコレクションを構成します。詳細については、[NIMO の説明](#)を参照してください。

プランファイルのロード

プランファイルを読み込んでネットワークモデルを作成できます。これは、たとえば、収集されたトポロジとデマンドの情報を含むプランファイルがすでにある場合に役立ちます。基本的なトポロジ収集を使用して新しいネットワークモデルを作成してデマンドで拡張することによってゼロから始める代わりに、既存のプランファイルをロードできます。

次のコマンドを使用して、プランファイルからネットワークモデルを作成します。

```
# wae components load-plan run plan-file <plan-file-location> network-name
<network-model-name>
```

次に例を示します。

```
# wae components load-plan run plan-file /home/tommy/us_atlanta_wan1.txt network-name
NetworkABC_topo_demands
```

追加 NIMO の構成

このトピックでは、さまざまなタイプの高度なネットワークデータ収集を構成するための一般的な手順について説明します。NIMOは、さまざまなタイプのデータを収集するために使用されます。一部のNIMOでは、エージェントの構成が必要です。詳細については、「[NIMOの説明](#)」を参照してください。

始める前に

送信元ネットワークとして使用するには、基本的な収集を含むネットワークモデルが必要です。

次のコマンドを入力します。

```
# networks network <network-model-name> nimo <NIMO-name> source-network <source-network>
# networks network <network-model-name> nimo <NIMO-name> <parameter-x> <parameter-x-option>
```

WAE CLI を使用したアーカイブの構成

[ネットワーク モデル コンポーザ](#) を使用してアーカイブを設定します。も可能です。

ステップ 1 WAE CLI を起動し、コンフィギュレーションモードに移行します。

```
# wae_cli -C
# config
(config)#
```

ステップ 2 アーカイブディレクトリを設定し、ファイルからアーカイブデータを取得するかどうかを選択します。

```
(config)# networks network <network_model_name> plan-archive archive-dir <archive_directory>
(config)# networks network <network_model_name> plan-archive source <file>
(config)# commit
```

次に例を示します。

```
(config)# networks network Network_123 plan-archive archive-dir /archive/planfiles/Network_123
(config)# networks network <network_model_name> plan-archive source filename
(config)# commit
```

ステップ3 アーカイブを実行します。アーカイブを実行すると、現在のネットワークモデルがプランファイル (.pln フォーマット) で指定したアーカイブディレクトリに保存されます。

```
(config)# networks network <network_model_name> plan-archive run
```

次に例を示します。

```
(config)# networks network Network_123 plan-archive run
status true
message Successfully archived plan file 20170131_1919_UTC.pln for network Network_123
```

ステップ4 アーカイブディレクトリに移動して、プランファイルが保存されたことを確認します。アーカイブディレクトリは、年、月、日といったサブフォルダに分かれています。

次に例を示します。

```
(config)# ls /Network_123/2017/01/31
20170131_0100_UTC.pln 20170131_0330_UTC.pln 20170131_1012_UTC.pln
20170131_1312_UTC.pln 20170131_1919_UTC.pln
```

次のタスク

プランファイルをアーカイブに保存する頻度をスケジュールします。

アーカイブ内のプランファイルの管理

アーカイブが構成され、アーカイブディレクトリが作成されたことを確認します。詳細については、[WAE CLI を使用したアーカイブの構成 \(18 ページ\)](#) を参照してください。

アーカイブでは、次のタスクを実行できます。

- すべてのプランファイルを一覧表示するには :

```
(config)# networks network <network_model_name> plan-archive list
```

- ネットワークモデルをアーカイブに保存するには :

```
(config)# networks network <network_model_name> plan-archive run
```

- プランファイルを取得するには :

```
(config)# networks network <network_model_name> plan-archive get
```



(注) 既存のプランファイルを使用してネットワークモデルを作成できます。[プランファイルのロード \(17 ページ\)](#) を参照してください。

翻訳について

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