



Cisco Unified Communication IOS サービスの設定

【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。

この章の内容は、次のとおりです。

- 「Cisco Unified Communication IOS サービス用のルータの設定」 (P.1)
- 「Cisco Unified Communication IOS サービスの確認とトラブルシューティング」 (P.10)
- 「コマンドリファレンス」 (P.10)

Cisco Unified Communication IOS サービス用のルータの設定

この項では、ゲートウェイ上のプロバイダーに対応するルータの設定方法を説明します。

前提条件

Cisco IOS Release 15.2(2)T

ルータへの Cisco Unified Communication IOS サービスの設定

ルータに Cisco Unified Communication IOS サービスを設定するには、次の手順を実行します。

手順の概要

1. `enable`
2. `configure terminal`
3. `ip http server`
4. `ip http max-connection value`
5. `ip http timeout-policy idle seconds life seconds requests value`
6. `http client persistent`
7. `http client connection idle timeout seconds`
8. `uc wsapi`
9. `message-exchange max-failures number`
10. `probing max-failures number`
11. `probing interval keepalive seconds`
12. `probing interval negative seconds`
13. `source-address ip-address`
14. `end`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>enable</code> 例： Router> enable	特権 EXEC モードをイネーブルにします。 プロンプトが表示されたら、パスワードを入力します。
ステップ 2	<code>configure terminal</code> 例： Router# 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	<code>ip http server</code> 例： Router(conf)# ip http server	システムで HTTP サーバ (Web サーバ) をイネーブルにします。

コマンドまたはアクション	目的
<p>ステップ4 <code>ip http max-connection value</code></p> <p>例： Router(conf)# ip http max-connection 100</p>	<p>HTTP サーバへの同時接続の最大許容数を設定します。デフォルト値は 5 です。</p>
<p>ステップ5 <code>ip http timeout-policy idle seconds life seconds requests value</code></p> <p>例： Router(conf)# ip http timeout-policy idle 600 life 86400 requests 86400</p>	<p>HTTP サーバに対する接続の維持時間を決定する特性を設定します。特性には次のものがあります。</p> <p>idle : 接続において送受信されるデータがない場合に、接続が開いたまま維持される最大時間 (秒数) です。既存の接続に対しては、新しい値は適用されないことに注意してください。サーバがビジーの場合や、life で制限された時間、または requests で制限された数に到達した場合は、設定より早く接続が閉じられる場合があります。デフォルト値は 180 秒 (3 分) です。</p> <p>life : 接続が確立されてから開いたまま維持される最大時間 (秒数) です。既存の接続に対しては、新しい値は適用されないことに注意してください。サーバがビジーの場合や、idle で制限された時間、または requests で制限された数に到達した場合は、設定より早く接続が閉じられる場合があります。また、サーバがアクティブに要求を処理している場合はサーバにより接続が閉じられることはないため、life で設定された最大秒数に達した際に処理が進行中の場合、life で設定した時間より長く接続が開かれたままになります。この場合は、処理が終了すると接続が閉じられます。デフォルト値は 180 秒 (3 分) です。最大値は 86400 秒 (24 時間) です。</p> <p>requests : 固定接続が閉じられるまでに処理する要求の最大数です。既存の接続に対しては、新しい値は適用されないことに注意してください。サーバがビジーの場合や、idle または life で制限された時間に到達した場合は、設定した最大数の要求が処理される前に接続が閉じられる場合があります。デフォルト値は 1 です。最大値は 86400 です。</p>
<p>ステップ6 <code>http client persistent</code></p> <p>例： Router(conf)# http client persistent</p>	<p>HTTP 固定接続をイネーブルにします。</p>
<p>ステップ7 <code>http client connection idle timeout seconds</code></p> <p>例： Router(conf)# http client idle timeout 600</p>	<p>クライアントが接続を閉じるまで、idle 状態を維持する秒数を設定します。</p>

	コマンドまたはアクション	目的
ステップ 8	<code>uc wsapi</code> 例： Router(config)# uc wsapi	Cisco Unified Communication IOS サービス コンフィギュレーション モードを開始します。
ステップ 9	<code>message-exchange max-failures number</code> 例： Router(config-uc-wsapi)# message-exchange max failures 2	アプリケーションとプロバイダーの間で交換されるメッセージの最大失敗数を設定します。この数を超えると、プロバイダーはアプリケーションへのメッセージ送信を停止します。範囲は 1 ～ 3 です。デフォルトは 1 です。
ステップ 10	<code>probing max-failures number</code> 例： Router(config-uc-wsapi)# probing max-failures 5	プローブ メッセージの最大失敗数を設定します。この数を超えると、ルータはアプリケーションを登録解除します。範囲は 1 ～ 5 です。デフォルトは 3 です。
ステップ 11	<code>probing interval keepalive seconds</code> 例： Router(config-uc-wsapi)# probing interval 180	プローブ メッセージの間隔 (秒数) を設定します。デフォルトは 120 秒です。
ステップ 12	<code>probing interval negative seconds</code> 例： Router(config-uc-wsapi)# probing interval negative 10	ネガティブ プローブ メッセージの間隔 (秒数) を設定します。
ステップ 13	<code>source-address ip-address</code> 例： Router(config-uc-wsapi)# source-address 172.1.12.13	UC IOS サービスの送信元 IP アドレスとして IP アドレス (ホスト名) を設定します。 (注) この送信元 IP アドレスが、プロバイダーによって NotifyProviderStatus メッセージで使用されます。
ステップ 14	<code>end</code> 例： Router(config-uc-wsapi)# end	特権 EXEC モードに戻ります。

ルータへの XCC プロバイダーの設定

ルータに XCC プロバイダーを設定するには、次の手順を実行します。

手順の概要

1. `enable`
2. `configure terminal`
3. `uc wsapi`
4. `provider xcc`
5. `no shutdown`
6. `remote-url url`

7. exit

8. end

手順の詳細

	コマンドまたはアクション	目的
ステップ1	<code>enable</code> 例： Router> enable	特権 EXEC モードをイネーブルにします。 プロンプトが表示されたら、パスワードを入力します。
ステップ2	<code>configure terminal</code> 例： Router# 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	<code>uc wsapi</code> 例： Router(conf)# uc wsapi	Cisco Unified Communication IOS サービス コンフィギュレーション モードを開始します。
ステップ4	<code>provider xcc</code> 例： Router(config-uc-wsapi)# provider xcc	XCC プロバイダー 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	<code>no shutdown</code> 例： Router(config-uc-wsapi-xcc)# no shutdown	XCC プロバイダーをアクティブにします。
ステップ6	<code>remote-url url</code> 例： Router(config-uc-wsapi-xcc)# remote-url http://209.133.85.47:8090/my_callcontrol	アプリケーションが XCC プロバイダーとの通信に使用する URL (IP アドレスとポート番号) を指定します。XCC プロバイダーは、IP アドレスとポートを使用して着信要求を認証します。
ステップ7	<code>exit</code> 例： Router(config-uc-wsapi-xcc)# exit	XCC コンフィギュレーション モードを終了します。
ステップ8	<code>end</code> 例： Router(config-uc-wsapi)# end	特権 EXEC モードに戻ります。

ルータへの XSVC プロバイダーの設定

ルータに XSVC プロバイダーを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **uc wsapi**
4. **provider xsvc**
5. **no shutdown**
6. **remote-url [url-number] url**
7. **exit**
8. **trunk group name**
9. **description**
10. **xsvc**
11. **exit**
12. **voip trunk group name**
13. **description**
14. **xsvc**
15. **session target ipv4:destination-address**
16. **exit**
17. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Router> enable	特権 EXEC モードをイネーブルにします。 プロンプトが表示されたら、パスワードを入力します。
ステップ 2	configure terminal 例： Router# 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	uc wsapi 例： Router(conf)# uc wsapi	Cisco Unified Communication IOS サービス コンフィギュレーション モードを開始します。
ステップ 4	provider xsvc 例： Router(config-uc-wsapi)# provider xsvc	XSVC プロバイダー 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	no shutdown 例： Router(config-uc-wsapi-xsvc)# no shutdown	XSVC プロバイダーをアクティブにします。

コマンドまたはアクション	目的
<p>ステップ 6 <code>remote-url [url-number] url</code></p> <p>例: Router(config-uc-wsapi-xsvc)# remote-url 1 http://209.133.85.47:8090/my_route_control</p>	<p>アプリケーションが XSVC プロバイダーとの通信に使用できる最大 8 つの異なる URL (IP アドレスとポート番号) を指定します。XSVC プロバイダーは、IP アドレスとポートを使用して着信要求を認証します。</p> <p><code>url-number</code> は、一意の URL を示します。範囲は 1 ~ 8 です。</p>
<p>ステップ 7 <code>exit</code></p> <p>例: Router(config-uc-wsapi-xsvc)# exit</p>	<p>XSVC コンフィギュレーション モードを終了します。</p>
<p>ステップ 8 <code>trunk group name</code></p> <p>例: Router(config)# trunk group SJ_PRI</p>	<p>トランクグループ コンフィギュレーション モードを開始して、トランク グループを定義します。</p>
<p>ステップ 9 <code>description</code></p> <p>例: Router(config)# description IN</p>	<p>トランク グループの説明を入力します。名前は、XSVC ステータスおよび XCC 接続メッセージの一部として外部アプリケーションに渡されます。</p>
<p>ステップ 10 <code>xsvc</code></p> <p>例: Router(config-trunk-group)# xsvc</p>	<p>トランク グループに対して <code>xsvc</code> のモニタリングをイネーブルにします。</p>
<p>ステップ 11 <code>exit</code></p> <p>例: Router(config-trunk-group)# exit</p>	<p>トランク グループ コンフィギュレーション モードを終了します。</p>
<p>ステップ 12 <code>voip trunk group name</code></p> <p>例: Router(config)# trunk group SJ_SIP</p>	<p>VoIP トランクグループ コンフィギュレーション モードを開始して、トランク グループを定義します。</p>
<p>ステップ 13 <code>description</code></p> <p>例: Router(config-voip-trk-gp)# description IN</p>	<p>VoIP トランク グループの説明を入力します。名前は、XSVC ステータスおよび XCC 接続メッセージの一部として外部アプリケーションに渡されます。</p>
<p>ステップ 14 <code>xsvc</code></p> <p>例: Router(config-voip-trk-gp)# xsvc</p>	<p>VoIP トランク グループに対して <code>xsvc</code> のモニタリングをイネーブルにします。</p>
<p>ステップ 15 <code>session target ipv4:destination address</code></p> <p>例: Router(config-voip-trk-gp)# session target ipv4:9.10.31.254</p>	<p>リモート ルータの IP アドレスを設定します。</p>

	コマンドまたはアクション	目的
ステップ 16	<code>exit</code> 例： Router(config-voip-trk-gp)# exit	VoIP トランク グループ コンフィギュレーション モードを終了します。
ステップ 17	<code>end</code> 例： Router(config-uc-wsapi)# end	特権 EXEC モードに戻ります。

ルータへの XCDR プロバイダーの設定

ルータに XCDR プロバイダーを設定するには、次の手順を実行します。

手順の概要

1. `enable`
2. `configure terminal`
3. `uc wsapi`
4. `provider xcdr`
5. `no shutdown`
6. `remote-url [url-number] url`
7. `exit`
8. `end`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>enable</code> 例： Router> enable	特権 EXEC モードをイネーブルにします。 プロンプトが表示されたら、パスワードを入力します。
ステップ 2	<code>configure terminal</code> 例： Router# 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	<code>uc wsapi</code> 例： Router(conf)# uc wsapi	Cisco Unified Communication IOS サービス コンフィギュレーション モードを開始します。
ステップ 4	<code>provider xcdr</code> 例： Router(config-uc-wsapi)# provider xcdr	XCDR プロバイダー 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 5	no shutdown 例： Router(config-uc-wsapi-xcdr)# no shutdown	XCDR プロバイダーをアクティブにします。
ステップ 6	remote-url [url-number] url 例： Router(config-uc-wsapi-xcdr)# remote-url 1 http://209.133.85.47:8090/my_route_control	アプリケーションが XCDR プロバイダーとの通信に使用できる最大 8 つの異なる URL (IP アドレスとポート番号) を指定します。XCDR プロバイダーは、IP アドレスとポートを使用して着信要求を認証します。 <i>url-number</i> は、一意の URL を示します。範囲は 1 ~ 8 です。
ステップ 7	exit 例： Router(config-uc-wsapi-xcdr)# exit	XCDR コンフィギュレーション モードを終了します。
ステップ 8	end 例： Router(config-uc-wsapi)# end	特権 EXEC モードに戻ります。

設定例

次に、Cisco Unified Communication IOS サービス用にルータを設定する例を示します。この例では、HTTP サーバと XCC、XSVC、および XCDR プロバイダーをイネーブルにします。この設定は、アプリケーションが XCC、XSVC、および XCDR プロバイダーとの通信に使用するアドレスとポートを指定します。また、XSVC がモニタするトランク グループを識別します。



(注) XSVC および XCDR は、最大 8 つの異なるリモート URL をサポートできます。

```
ip http server
!
call fallback monitor
call fallback icmp-ping count 1 interval 2 timeout 100
!
uc wsapi
  source-address 10.1.1.1
  provider xcc
    remote-url http://test.com:8090/xcc
  !
  provider xsvc
    remote-url 1 http://test.com:8090/xsvc
  !
  provider xcdr
    remote-url 1 http://test.com:8090/xcdr
  !
trunk group pri
  xsvc

voip trunk group 1
  xsvc
```

```
session target ipv4: 11.1.1.1
!  
interface Serial0/1/0:23  
isdn switch-type primary-ni  
isdn incoming-voice voice  
trunk-group pri
```

Cisco Unified Communication IOS サービスの確認とトラブルシューティング

次の show コマンドを使用して、Cisco Unified Communication IOS サービスのパフォーマンスに関する情報を収集します。

- **show wsapi registration**
- **show wsapi http client**
- **show wsapi http server**
- **show wsapi xsvc routes**

次の debug コマンドを使用して、サービス プロバイダーのトラブルシューティング情報を収集します。

- **debug wsapi xcc [CR | all | function | default | detail | error | inout | event]**
- **debug wsapi xsvc [CR | all | function | default | detail | error | inout | event]**
- **debug wsapi xcdr [CR | all | function | default | detail | error | inout | event]**
- **debug wsapi infrastructure [CR | all | function | default | detail | error | inout | event]**

コマンド リファレンス

この項では、ルータで使用される CLI コマンドを示します。

- 「[debug wsapi](#)」 (P.11)
- 「[message-exchange max-failures](#)」 (P.14)
- 「[probing interval](#)」 (P.15)
- 「[probing max-failures](#)」 (P.16)
- 「[provider](#)」 (P.17)
- 「[remote-url](#)」 (P.18)
- 「[show call media forking](#)」 (P.19)
- 「[show voip trunk group](#)」 (P.20)
- 「[show wsapi](#)」 (P.21)
- [source-address \(uc-wsapi\)](#), page 24
- 「[uc wsapi](#)」 (P.24)
- 「[voip trunk group](#)」 (P.25)
- 「[xsvc](#)」 (P.26)

debug wsapi

Cisco Unified Communication IOS サービス アプリケーション プログラミング インターフェイスのトレースを収集し、表示するには、特権 EXEC モードで **debug wsapi** コマンドを使用します。デバッグをディセーブルにするには、このコマンドの **no** 形式を使用します。

debug wsapi {infrastructure | xcc | xcdr | xsvc } [all | default | detail | error | event | function | inout | messages]

no debug wsapi {infrastructure | xcc | xcdr | xsvc } [all | default | detail | error | event | function | inout | messages]

構文の説明

infrastructure	インフラストラクチャに対してデバッグ トレースをイネーブルにします。
xcc	XCC プロバイダーに対してデバッグ トレースをイネーブルにします。
xcdr	XCDR プロバイダーに対してデバッグ トレースをイネーブルにします。
xsvc	XSVC プロバイダーに対してデバッグ トレースをイネーブルにします。
all	すべてのデバッグ トレースをイネーブルにします。
default	デフォルトのデバッグ トレースをイネーブルにします。
detail	詳細なデバッグ トレースをイネーブルにします。
error	エラー デバッグ トレースをイネーブルにします。
event	イベント デバッグ トレースをイネーブルにします。
function	機能デバッグ トレースをイネーブルにします。
inout	入出力デバッグ トレースをイネーブルにします。
messages	API メッセージ トレースをイネーブルにします。

コマンドモード

特権 EXEC

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、Cisco Unified Communication IOS サービス サブシステムに対するデバッグ トレースをイネーブルにします。

例

次に、XCC 登録に対する **debug wsapi infrastructure** コマンドからのデバッグ出力を示します。

```
Router# debug wsapi infrastructure

23:25:09: //WSAPI/INFRA/wsapi_https_urlhook:
23:25:09: //WSAPI/INFRA: app_name cisco_xcc in url /cisco_xcc in port 8090
23:25:09: //WSAPI/INFRA/wsapi_https_urlhook: Exit
23:25:09: //WSAPI/INFRA/wsapi_https_post_action:
23:25:09: wsapi_https_data_read: <soapenv:Envelope
xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"><soapenv:Body><RequestXccRegister
xmlns="http://www.cisco.com/schema/cisco_xcc/v1_0"><applicationData><name>myapp</name><url
>http://sj22lab-as2:8090/xcc</url></applicationData><blockingEventTimeoutSec>1</blockingEv
```

```

entTimeoutSec><blockingTimeoutHandle>CONTINUE_PROCESSING</blockingTimeoutHandle><connectionEventsFilter>CREATED AUTHORIZE_CALL REDIRECTED ALERTING CONNECTED TRANSFERRED CALL_DELIVERY DISCONNECTED HANDOFFLEAVE HANDOFFJOIN</connectionEventsFilter><mediaEventsFilter>MODE_CHANGE DTMF TONE_BUSY TONE_DIAL TONE_SECOND_DIAL TONE_RINGBACK TONE_OUT_OF_SERVICE MEDIA_ACTIVITY</mediaEventsFilter><msgHeader><transactionID>txID001</transactionID></msgHeader><providerData><url>http://10.1.1.1:8090/cisco_xcc</url></providerData></RequestXccRegister></soapenv:Body></soapenv:Envelope>
23:25:09: //WSAPI/INFRA/27/0/wsapi_https_recv:
23:25:09: //WSAPI/INFRA/27/0/txID001/wsapi_ph_request_msg_handle:
23:25:09: //WSAPI/INFRA/27/0/txID001: prov_type 0 msg_type 6 prov_state 1
23:25:09: //WSAPI/INFRA/wsapi_create_common_msg:
23:25:09: //WSAPI/INFRA/wsapi_create_common_msg: Exit
23:25:09: //WSAPI/INFRA/27/0/txID001/wsapi_send_outbound_response:
23:25:09: wsapi_dump_msg: type 8
23:25:09: transactionID txID001
23:25:09: registrationID 50674FC:XCC:myapp:9
23:25:09: ResponseXccRegister:
23:25:09: providerStatus 1
23:25:09: //WSAPI/INFRA/27/0/txID001/wsapi_send_outbound_response: Exit
23:25:09: wsapi_send_ResponseRegister:mem_mgr_mempool_free: mem_refcnt(3CA18B8)=0 - mempool cleanup
23:25:09: //WSAPI/INFRA/27/0/txID001/wsapi_https_recv: Exit
23:25:09: wsapi_https_data_write: <?xml version="1.0" encoding="UTF-8"?><SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"><SOAP:Body><ResponseXccRegister xmlns="http://www.cisco.com/schema/cisco_xcc/v1_0"><msgHeader><transactionID>txID001</transactionID><registrationID>50674FC:XCC:myapp:9</registrationID></msgHeader><providerStatus>IN_SERVICE</providerStatus></ResponseXccRegister></SOAP:Body></SOAP:Envelope>
23:25:09: //WSAPI/INFRA/wsapi_https_post_action: Exit

```

次に、コールに対する **debug wsapi xcc all** コマンドからの部分的なデバッグ ログを示します。

```
Router# debug wsapi xcc all
```

```

23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_sessStore_call_add:271:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_db:145:
23:27:20: //WSAPI/XCC/xccp_session_call_add:353: xcc session successfully added
23:27:20: //WSAPI/XCC/xccp_sessStore_call_add:285: xcc call successfully added
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_create_outbound_msg_space:677:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_callData:225:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_db:145:
23:27:20: //WSAPI/XCC/xccp_session_get_callData:445:
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_notify_events:434:
23:27:20: //WSAPI/XCC/xccp_queue_events:304:
23:27:20: //WSAPI/XCC/provider_base_event_new:335:
23:27:20: //WSAPI/UNKNOWN/event_base_new:267:
23:27:20: //WSAPI/XCC: magic [0xBABE] state[EVENT_STATE_ACTIVE] owner [0x1148C178] evSize[56] debFlag[3] evHdlr[0x894D834] evHdlFree[0x894DB00]
23:27:20: //WSAPI/UNKNOWN/event_base_new:292: event base new succ
23:27:20: //WSAPI/XCC/provider_base_event_new:360: provider base eventNew success
23:27:20: //WSAPI/XCC/provider_base_add_ev_to_q:393:
23:27:20: //WSAPI/XCC/check_xccp_active:177:

```

```

23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_create_outbound_msg_space:677:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_callData:225:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_db:145:
23:27:20: //WSAPI/XCC/xccp_session_get_callData:445:
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_solicit_events:359:
23:27:20: //WSAPI/XCC/xccp_queue_events:304:
23:27:20: //WSAPI/XCC/provider_base_event_new:335:
23:27:20: //WSAPI/UNKNOWN/event_base_new:267:
23:27:20: //WSAPI/XCC: magic [0xBABE] state[EVENT_STATE_ACTIVE] owner [0x1148C178]
evSize[56] debFlag[3] evHdr[0x894D834] evHdrFree[0x894DB00]
23:27:20: //WSAPI/UNKNOWN/event_base_new:292: event base new succ
23:27:20: //WSAPI/XCC/provider_base_event_new:360: provider base eventNew success
23:27:20: //WSAPI/XCC/provider_base_add_ev_to_q:393:
23:27:20: //WSAPI/XCC/provider_base_process_events:444:
23:27:20: //WSAPI/XCC/xccp_handle_events:153:
23:27:20: //WSAPI/INFRA/wsapi_send_outbound_message:
23:27:20: //WSAPI/INFRA/wsapi_send_outbound_message_by_provider_info:
23:27:20: //WSAPI/XCC/wsapi_xcc_encode_outbound_msg:
23:27:20: //WSAPI/XCC/wsapi_xcc_encode_outbound_msg: Exit
23:27:20: //WSAPI/INFRA/0/1527/50875A4:319:out_url http://sj22lab-as2:8090/xcc
23:27:20: wsapi_send_outbound_message_by_provider_info: <?xml version="1.0"
encoding="UTF-8"?><SOAP:Envelope
xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"><SOAP:Body><NotifyXccConnectionData
xmlns="http://www.cisco.com/schema/cisco_xcc/v1_0"><msgHeader><transactionID>50875A4:319</
transactionID><registrationID>50674FC:XCC:myapp:9</registrationID></msgHeader><callData><c
allID>9</callID><state>ACTIVE</state></callData><connData><connID>1527</connID><state>IDLE
</state></connData><event><created><connDetailData><connData><connID>1527</connID><state>I
DLE</state></connData><guid>7A1E678F-8259-11E0-8FF1-D29982DCA129</guid><callingAddrData><t
ype>E164</type><addr>5522101</addr></callingAddrData><calledAddrData><type>E164</type><add
r>6001</addr></calledAddrData><origCallingAddrData><type>E164</type><addr>5522101</addr></
origCallingAddrData><origCalledAddrData><type>E164</type><addr>6001</addr></origCalledAddr
Data><connIntfType>CONN_SIP</connIntfType><mediaData><type>VOICE</type></mediaData><connIn
tf>1.3.45.2</connIntf><connDirectionType>INCOMING</connDirectionType></connDetailData></cr
eated></event></NotifyXccConnectionData></SOAP:Body></SOAP:Envelope>
23:27:20: //WSAPI/INFRA/0/1527/50875A4:319/wsapi_send_outbound_message_by_provider_info:
Exit
.
.
.

```

message-exchange max-failures

プロバイダーがアプリケーションへのメッセージ送信を停止するまで、アプリケーションとプロバイダーの間で交換されるメッセージの最大失敗数を設定するには、**message-exchange max-failures** コマンドを使用します。最大数をデフォルトの数にリセットするには、このコマンドの **no** 形式を使用します。

message-exchange max-failures *number*

no message-exchange max-failures *number*

構文の説明

number サービス プロバイダーがアプリケーションへのメッセージ送信を停止するまでに許可される、最大メッセージ数。範囲は 1 ～ 3 です。デフォルトは 1 です。

コマンド デフォルト

デフォルトは 1 です。

コマンド モード

uc wsapi コンフィギュレーション モード

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、システムがアプリケーションを到達不能と判断し、サービス プロバイダーがアプリケーションへのメッセージ送信を停止するまでに許可される、メッセージの最大失敗数を設定します。

例

次に、メッセージの最大失敗数を 2 回に設定する例を示します。

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# message-exchange max-failures 2
```

関連コマンド

コマンド	説明
probing interval	プローブ メッセージの間隔を設定します。
probing max-failure	応答がない状態でシステムが送信するメッセージの数を設定します。この数を超えると、システムはアプリケーションの登録を解除します。

probing interval

ルータによって送信されるプローブメッセージの間隔を設定するには、**probing interval** コマンドを使用します。間隔をデフォルト数値にリセットするには、このコマンドの **no** 形式を使用します。

probing interval [keepalive | negative] seconds

no probing interval keepalive [negative] seconds

構文の説明

keepalive	(任意) セッションがキープアライブ状態のときのプローブメッセージの間隔を設定します。範囲は 1 ~ 255 秒です。デフォルトは 5 秒です。
negative	(任意) セッションがネガティブ状態のときのプローブメッセージの間隔を設定します。範囲は 1 ~ 20 秒です。デフォルトは 5 秒です。
<i>seconds</i>	プローブメッセージ間隔の秒数。

デフォルト

デフォルトでは、セッションが正常な状態のときのプローブメッセージ間隔は 120 秒であり、セッションがネガティブ状態のときのプローブメッセージ間隔は 5 秒です。

コマンドモード

uc wsapi コンフィギュレーションモード。

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、ルータから送信されるプローブメッセージの間隔を設定します。

例

次に、正常なセッション時に 180 秒の間隔、およびセッションがネガティブ状態のときに 10 秒の間隔を設定する例を示します。

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# probing interval keepalive 180
Router(config-uc-wsapi)# probing interval negative 10
```

関連コマンド

コマンド	説明
message-exchange	メッセージ応答の最大失敗数を設定します。この数を超えると、プロバイダーはメッセージ送信を停止します。
probing max-failure	応答がない状態でシステムが送信するメッセージの数を設定します。この数を超えると、システムはアプリケーションの登録を解除します。

probing max-failures

システムがセッションを停止し、アプリケーションの登録を解除するまで、アプリケーションが応答に失敗するプローブメッセージの最大数を設定するには、**probing max-failures** コマンドを使用します。最大数をデフォルトの数にリセットするには、このコマンドの **no** 形式を使用します。

probing max-failures *number*

no probing max-failures *number*

構文の説明

<i>number</i>	システムがセッションを停止し、アプリケーションの登録を解除するまでに許可されるメッセージの最大数。範囲は 1 ～ 5 です。デフォルトは 3 です。
---------------	--

コマンド デフォルト

デフォルトは 3 です。

コマンド モード

uc wsapi コンフィギュレーション モード。

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、システムから送信されてアプリケーションが応答しないプローブメッセージの最大数を設定します。この数を超えると、システムはセッションを停止し、アプリケーションセッションの登録を解除します。

例

次に、メッセージの最大失敗数を 5 回に設定する例を示します。

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# probing max-failures 5
```

関連コマンド

コマンド	説明
message-exchange	メッセージ応答の最大失敗数を設定します。この数を超えると、プロバイダーはメッセージ送信を停止します。
probing interval	プローブメッセージの間隔を設定します。

provider

サービス プロバイダーを設定してイネーブルにするには、**provider** コマンドを使用します。プロバイダーを削除するには、このコマンドの **no** 形式を使用します。

provider [XCC | XSVC | XCDR]

no provider [XCC | XSVC | XCDR]

構文の説明

XCC	(任意) XCC サービス プロバイダーをイネーブルにします。
XSVC	(任意) XSVC サービス プロバイダーをイネーブルにします。
XCDR	(任意) XCDR サービス プロバイダーをイネーブルにします。

デフォルト

デフォルトの動作はありません。

コマンド モード

uc wsapi コンフィギュレーション モード

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、サービス プロバイダーをイネーブルにします。

例

次に、XCC サービス プロバイダーをイネーブルにする例を示します。

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# provider xcc
Router(config-uc-wsapi-xcc)# no shutdown
```

関連コマンド

コマンド	説明
remote-url	アプリケーションの URL を指定します。
source-address	プロバイダーの IP アドレスを指定します。
uc wsapi	Cisco Unified Communication IOS サービス コンフィギュレーション モードを開始します。

remote-url

サービス プロバイダーが使用するアプリケーションの URL を設定するには、**remote-url** コマンドを使用します。プロバイダーはこの URL を使用して、アプリケーションを認証し、通信します。設定された URL を削除するには、このコマンドの **no** 形式を使用します。

remote-url [*url-number*] *url*

no remote-url [*url-number*] *url*

構文の説明

<i>url-number</i>	(任意) URL 番号。範囲は 1 ~ 8 です。
<i>url</i>	サービス プロバイダーがメッセージで使用する URL を指定します。

コマンド デフォルト

なし

コマンド モード

uc wsapi コンフィギュレーション モード。

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、サービス プロバイダーがメッセージで使用するリモート URL (アプリケーション) を設定します。

例

次に、XCC サービス プロバイダーがメッセージで使用するリモート URL の設定例を示します。

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# provider xcc
Router(config-uc-wsapi-xcc)# no shutdown
Router(config-uc-wsapi-xcc)# remote-url 1 http://209.133.85.47:8090/my_route_control
```

関連コマンド

コマンド	説明
provider	プロバイダー サービスをイネーブルにします。
source-address	プロバイダーの IP アドレスを指定します。
uc wsapi	Cisco Unified Communication IOS サービス コンフィギュレーション モードを開始します。

show call media forking

現在アクティブなメディア分岐セッションを表示するには、ユーザ EXEC モードまたは特権 EXEC モードで **show call media forking** コマンドを使用します。

show call media forking

構文の説明

このコマンドには引数またはキーワードはありません。

コマンドモード

ユーザ EXEC (>)
特権 EXEC (#)

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、関連するアンカー レッグに対するメディア分岐が成功したことを確認します。

例

次に、**show call media forking** コマンドからの出力例を示します。

```
Router# show call media forking

Warning: Output may be truncated if sessions are added/removed concurrently!

Session      Call      n/f  Destination (port address)
-----
7             6         far  1234 1.5.35.254
8             6         near 5678 1.5.35.254
```

表 1 では、表示されるフィールドについて説明します。

表 1 Show Call Media Forking のフィールドの説明

フィールド	説明
Session	セッション ID。
Call	16 進数のコール レッグ ID。show call leg active コマンドからのコール ID と一致する必要があります。
n/f	分岐された音声ストリームの方向（近端または遠端）。
Destination (port address)	分岐されたパケットの宛先。次の要素で構成されます。 <ul style="list-style-type: none"> RTP ポート IP アドレス

show voip trunk group

VoIP トランク グループの内部リストを表示するには、ユーザ EXEC または特権 EXEC モードで **show voip trunk group** コマンドを使用します。

show voip trunk group

構文の説明

このコマンドには引数またはキーワードはありません。

コマンド モード

ユーザ EXEC (>)

特権 EXEC (#)

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、VoIP トランク グループを表示します。

例

次に、**show voip trunk group** コマンドの出力例を示します。

```
Router# show voip trunk group
=====
name:          1
protocol:      cisco
ip:            1.3.45.2
xsvc:          TRUE
```

show wsapi

登録情報、統計情報、およびルート情報など、Cisco Unified Communication IOS サービスの情報を表示するには、ユーザ EXEC モードまたは特権 EXEC モードで **show wsapi** コマンドを使用します。

show wsapi {http-client | http-server | registration {all | xcc | xcdr | xsvc } | xsvc route }

構文の説明

http-client	HTTP クライアント インターフェイスで収集された統計情報を表示します。
http-server	HTTP サーバ インターフェイスで収集された統計情報を表示します。
registration	WSAPI サブシステムに現在登録されているアプリケーションを表示します。
all	登録されているすべてのアプリケーションを表示します。
xcc	XCC プロバイダーに登録されているアプリケーションを表示します。
xcdr	XCDR プロバイダーに登録されているアプリケーションを表示します。
xsvc	XSVC プロバイダーに登録されているアプリケーションを表示します。
xsvc route	XSVC プロバイダーの内部ルート情報を表示します。

コマンドモード

ユーザ EXEC
特権 EXEC

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、Cisco Unified Communication IOS サービスの情報を表示します。

例

次に、**show wsapi http-client** コマンドの出力例を示します。

```
Router# show wsapi http-client

WSAPI Outgoing Notify/Solicit Message Statistics
=====

wsapi_show_httpc_callback_context_invalid: 0
wsapi_show_httpc_callback_context_error: 0
wsapi_show_httpc_callback_no_reg: 5
wsapi_show_httpc_callback_notify_OK: 85
wsapi_show_httpc_callback_notify_error: 0
wsapi_show_httpc_callback_client_error: 0
wsapi_show_httpc_callback_error: 7
wsapi_show_httpc_callback_client_error: 0
wsapi_show_httpc_callback_decode_error: 28
wsapi_show_httpc_callback_no_txID: 0
wsapi_show_httpc_callback_OK: 655
wsapi_show_httpc_create_msg_error: 0
wsapi_show_httpc_context_active: 0
wsapi_tx_context_freeq depth: 4
```

次に、**show wsapi http-server** コマンドの出力例を示します。

```
Router# show wsapi http-server

WSAPI Incoming Request Message Statistics
=====

wsapi_show_https_urlhook: 23
wsapi_show_https_post_action: 23
wsapi_show_https_post_action_fail: 0
wsapi_show_https_xml_fault: 0
wsapi_show_https_post_action_done: 23
wsapi_show_https_service_timeout: 0
wsapi_show_https_send_error: 0
wsapi_show_https_invalid_context: 0
wsapi_show_https_data_active: 0
wsapi_https_data_q depth: 1
wsapi_show_https_internal_service_error: 0
wsapi_show_https_service_unavailable_503: 0
wsapi_show_https_not_found_404: 0
wsapi_show_https_registration_success: 9
wsapi_show_https_not_registered: 0
wsapi_show_https_registration_auth_fail: 1
wsapi_show_https_registration_fail: 0
wsapi_show_https_un_registered: 0
```

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

```
Router# show wsapi registration all

Provider XCC
=====
registration
  id: 4FA11CC:XCC:myapp:5
  appUrl:http://sj22lab-as2:8090/xcc
  appName: myapp
  provUrl: http://10.1.1.1:8090/cisco_xcc
  prober state: STEADY
  connEventsFilter:
  CREATED|AUTHORIZE_CALL|ADDRESS_ANALYZE|REDIRECTED|ALERTING|CONNECTED|TRANSFERRED|CALL_DELI
  VERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_LEAVE
  mediaEventsFilter:
  DTMF|MEDIA_ACTIVITY|MODE_CHANGE||TONE_DIAL|TONE_OUT_OF_SERVICE|TONE_RINGBACK|TONE_SECOND_D
  IAL
  blockingEventTimeoutSec: 1
  blockingTimeoutHandle: CONTINUE_PROCESSING

Provider XSVC
=====
registration index: 2
  id: 4FA0F8C:XSVC:myapp:3
  appUrl:http://sj22lab-as2:8090/xsvc
  appName: myapp
  provUrl: http://10.1.1.1:8090/cisco_xsvc
  prober state: STEADY
  route filter:
  event filter: off

Provider XCDR
=====
registration index: 1
  id: 4FA10A0:XCDR:myapp:1
  appUrl:http://sj22lab-as2:8090/xcdr
```

```

appName: myapp
provUrl: http://10.1.1.1:8090/cisco_xcdr
prober state: STEADY
cdr format: COMPACT
event filter: off

```

次に、**show wsapi xsvc route** コマンドの出力例を示します。

```
Router# show wsapi xsvc route
```

```
Route SANJOSE_SIP
```

```

=====
Type: VOIP
Description: OUT
Filter:
Trunk:
    Trunk Name:      1.3.45.2
    Trunk Type:      SIPV2
    Trunk Status:    UP

```

```
Route SANJOSE_PRI
```

```

=====
Type: PSTN
Description: IN
Filter:
Trunk:
    Trunk Name:      Se0/1/0:23
    Trunk Type:      ISDN PRI
    Trunk Status:    UP
    Total channels   2
    Channel bitmap   0x01FFFFFFE 1-24
    Link bitmap      0x00000006
    Alarm            0x00000001
    Time elapsed     516
    Interval         92
    CurrentData
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
    TotalData
49 Line Code Violations, 7 Path Code Violations,
0 Slip Secs, 1 Fr Loss Secs, 1 Line Err Secs, 0 Degraded Mins,
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 2 Unavail Secs

    Trunk Name:      Se0/1/1:23
    Trunk Type:      ISDN PRI
    Trunk Status:    UP
    Total channels   2
    Channel bitmap   0x01FFFFFFE 1-24
    Link bitmap      0x00000006
    Alarm            0x00000001
    Time elapsed     516
    Interval         92
    CurrentData
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
    TotalData
42 Line Code Violations, 4 Path Code Violations,
0 Slip Secs, 1 Fr Loss Secs, 1 Line Err Secs, 0 Degraded Mins,
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 2 Unavail Secs

```

uc wsapi

特定のアプリケーションに Cisco Unified Communication IOS サービス環境を設定するには、**uc wsapi** コマンドを使用します。

uc wsapi

構文の説明

このコマンドには引数またはキーワードはありません。

コマンド デフォルト

なし

コマンド モード

EXEC モード。

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、Cisco Unified Communication IOS サービスの設定環境を開始します。

例

次に、Cisco Unified Communication IOS サービスの設定環境を開始する例を示します。

```
Router(config)# uc wsapi
Router(config-uc-wsapi)#
```

関連コマンド

コマンド	説明
provider	プロバイダー サービスをイネーブルにします。

voip trunk group

VoIP トランク グループを定義または変更したり、トランク グループ コンフィギュレーション モードを開始したりするには、グローバル コンフィギュレーション モードで **voip trunk group** コマンドを使用します。VoIP トランク グループを削除するには、このコマンドの **no** 形式を使用します。

voip trunk group name

no voip trunk group name

構文の説明

<i>name</i>	VoIP トランク グループの名前。有効な名前は、63 文字以内の英数字で構成されます。
-------------	--

コマンドデフォルト

VoIP トランク グループが定義されていません。

コマンドモード

グローバル コンフィギュレーション

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

voip trunk group コマンドを使用して、VoIP トランクを定義し、トランクまでサービスアビリティを拡張します。デフォルトでは、IP トランクのセッションプロトコルは **h323** です。ゲートウェイにプロファイルを格納するための十分なメモリがある場合、最大 1000 個のトランク グループをゲートウェイに設定できます。

例

次に、VoIP トランク グループを作成し、モニタリングをイネーブルにする例を示します。

```
Router(config)# voip trunk group siptrk1
Router(config-voip-trk)# session protocol sipv2
Router(config-voip-trk)# target ipv4: 10.1.1.15
Router(config-voip-trk)# xsvc
```

コマンド	説明
show voip trunk group	VoIP トランク グループの内部リストを表示します。
xsvc	トランクのモニタリングをイネーブルにします。

X SVC

トランク グループとして定義されている TDM (ISDN-PRI/BRI、DS0-group、アナログ音声ポート) 音声インターフェイスでの拡張サービスアビリティ (xsvc) のサポートを追加するには、**xsvc** コマンドを使用します。拡張サービスアビリティのサポートをディセーブルにするには、このコマンドの **no** 形式を使用します。

xsvc

no xsvc

構文の説明

このコマンドには引数またはキーワードはありません。

コマンドデフォルト

トランク グループでは、拡張サービスアビリティはディセーブルです。

コマンドモード

トランク グループ コンフィギュレーション

コマンド履歴

リリース	変更内容
15.2(2)T	このコマンドが追加されました。

使用上のガイドライン

このコマンドを使用して、トランク グループとして定義されている音声インターフェイスでの拡張サービスアビリティのサポートを追加します。

例

次に、トランク グループでモニタリングをイネーブルにする例を示します。

```
Router(config)# trunk group tdm-tg1
Router(config-trunk-group)# xsvc
```

関連コマンド

コマンド	説明
provider	プロバイダー サービスをイネーブルにします。



