

Jabber for Windows のコール フロー分析手順

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[Cisco Jabber for Windowsのコールフローの分析](#)

[Jabberログの用語](#)

[Csf-UNIFIED.LOG](#)

[Office統合 \(Jabberログでクリックツーコール \)](#)

[関連情報](#)

概要

このドキュメントでは、Cisco Jabber for Windowsのコールフローを分析するために使用する手順について説明します。

前提条件

要件

Cisco Jabber for Windowsバージョン9.xに関する知識があることが推奨されます。

使用するコンポーネント

このドキュメントの情報は、Cisco Jabber for Windowsバージョン9.xに基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

Cisco Jabber for Windowsのコールフローの分析

Cisco Jabber for Windowsでコールの開始と終了を確認するには、`csf-unified.log` ファイルをメモ帳++で開き、`CALL_EVENT`を検索します。コールの最初の行を検索するには、`CALL_EVENT`文

の直後にInitialを見つけます。

```
2013-02-24 18:22:07,362 INFO [0x000009e4][control\CallControlManagerImpl.cpp(2021)]
[csf.ecc.evt] [ecc::CallControlManagerImpl::notifyCallEventObservers] -
CALL_EVENT: evCreated,
0x005B1818, Initial, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_NONE, CalledPartyNumber: ,
CallingPartyNumber: ,
CallInstance: -1, Status: , GCID: , IsConference: 0, IsCallSelected: 0, CapabilitySet:
```

Jabberログの用語

Jabberのコールフローで次のキーワードを確認します。

コールの状態

- [Initial] : コールは設定中です。Jabberは着信か発信かを判別し、発信側または着信側の電話番号が準備されます。
- オフフック : Jabber電話機がオフフックになります。
- [ダイヤル(Dialing)] : 着信先電話番号の番号がダイヤルされます。
- [続行(Proceed)] : デイジットがCisco Unified Communications Manager(CUCM)に渡され、処理されます。
- RingOut : 番号がダイヤルされ、リモートエンドにリングが送信されます。
- Connected : リモートエンドが応答され、接続が確立されます。
- OnHook:Jabber電話が切断されました。

CC_CALL_TYPE : 初期化後に「NONE」から「OUTGOING」または「INCOMING」に変更されます。これは、コールが送受信されるかどうかによって異なります。

コールID : コール状態の前の16進数(0x005B1818)はJabberコールIDで、各コールの間は一貫して使用されます。次のコールが発生すると増加します。

Csf-UNIFIED.LOG

メモ帳++を使用している場合は、[Find All in Current Document] オプションを使用してコールフロー全体を表示できます。これは次のように表示されます (表示を簡単にするため、タイムスタンプと情報ブラケットは削除されています)。

```
CALL_EVENT: evCreated, 0x005B1818, Initial, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_NONE,
CalledPartyNumber: , CallingPartyNumber: , CallInstance: -1, Status: ,
GCID: , IsConference: 0,
IsCallSelected: 0, CapabilitySet:
```

```
CALL_EVENT: evInfoChanged, 0x005B1818, Initial, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
CalledPartyNumber: , CallingPartyNumber: , CallInstance: 1, Status: ,
GCID: , IsConference: 0,
IsCallSelected: 0, CapabilitySet: canOriginateCall,canDialVoicemail
```

```
CALL_EVENT: evInfoChanged, 0x005B1818, Initial, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
CalledPartyNumber: 1001, CallingPartyNumber: 1000, CallInstance: 1, Status: ,
GCID: , IsConference:
0, IsCallSelected: 0, CapabilitySet: canOriginateCall,canDialVoicemail
```

```
CALL_EVENT: evStateChanged, 0x005B1818, OffHook, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
```

CalledPartyNumber: 1001, CallingPartyNumber: 1000, CallInstance: 1, Status: ,
GCID: , IsConference:
0, IsCallSelected: 0, CapabilitySet: canEndCall,canSendDigit

CALL_EVENT: evStateChanged, 0x005B1818, **Dialing**, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
CalledPartyNumber: 1001, CallingPartyNumber: 1000, CallInstance: 1, Status: ,
GCID: , IsConference:
0, IsCallSelected: 0, CapabilitySet: canEndCall,canSendDigit

CALL_EVENT: evStateChanged, 0x005B1818, **Proceed**, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
CalledPartyNumber: 1001, CallingPartyNumber: 1000, CallInstance: 1, Status: ,
GCID: , IsConference:
0, IsCallSelected: 0, CapabilitySet: canEndCall

CALL_EVENT: evStateChanged, 0x005B1818, **RingOut**, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
CalledPartyNumber: 1001, CallingPartyNumber: 1000, CallInstance: 1, Status: ,
GCID: , IsConference:
0, IsCallSelected: 0, CapabilitySet: canEndCall,canDirectTransfer

CALL_EVENT: evCallStarted, 0x005B1818, **Connected**, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
CalledPartyNumber: 1001, CallingPartyNumber: 1000, CallInstance: 1, Status: ,
GCID: , IsConference:
0, IsCallSelected: 0, CapabilitySet: canHold,canEndCall,canSendDigit,canCallPark,
canDirectTransfer,
canJoinAcrossLine

CALL_EVENT: evStateChanged, 0x005B1818, **OnHook**, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
CalledPartyNumber: 1001, CallingPartyNumber: 1000, CallInstance: 1, Status: ,
GCID: , IsConference:
0, IsCallSelected: 0, CapabilitySet:

CALL_EVENT: evDestroyed, 0x005B1818, OnHook, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_OUTGOING,
CalledPartyNumber: , CallingPartyNumber: , CallInstance: 1, Status: ,
GCID: , IsConference:
0, IsCallSelected: 0, CapabilitySet:

出力は、Jabberが最初にオフフックになり、ダイヤルした時点、および発信側番号と着信側番号を表示します。着信コールの場合、同様のコールフローパターンが表示されます。ただし、**CC_CALL_TYPE**は異なります。

```
2013-02-24 18:22:07,130 INFO [0x00000ec0] [control\CallControlManagerImpl.cpp(2001)]  
[csf.ecc.evt][ecc::CallControlManagerImpl::notifyCallEventObservers] -  
CALL_EVENT: evCreated, 0x10003, Initial,  
CC_ATTR_NORMAL, CC_CALL_TYPE_INCOMING, CalledPartyNumber: 1001,  
CallingPartyNumber: 1000,  
CallInstance: 1, Status: , GCID: , IsConference: 0, IsCallSelected: 0, CapabilitySet:
```

Office統合 (Jabberログでクリックツーコール)

Microsoft Office Integrationから発信されたコールの場合、コールフローの前に1行が追加されます。「Starting voice call with」を検索すると、**Click to Call**機能が実行されるすべてのポイントがログに表示されます。たとえば、連絡先に対するMicrosoft Outlookからのコール(右クリックしてコール)は、次のようになります。

```
2013-02-24 18:42:18,912 INFO [0x000009e0] [\ExtensionProviderFunctionsImpl.cpp(235)]  
[cuc-extension-provider] [StartVoiceCallImpl2] - Starting voice call with 1001  
数ミリ秒後、同じ「Initial」キーワードが開始されます。
```

```
2013-02-24 18:42:18,915 INFO [0x0000053c] [control\CallControlManagerImpl.cpp(2021)]
[csf.ecc.evt][ecc::CallControlManagerImpl::notifyCallEventObservers] -
CALL_EVENT: evCreated, 0x00EF3BA0, Initial, CC_ATTR_NOT_DEFINED, CC_CALL_TYPE_NONE,
CalledPartyNumber: , CallingPartyNumber: ,
CallInstance: -1, Status: , GCID: , IsConference: 0, IsCallSelected: 0, CapabilitySet:
```

[Click to Call] 機能の[cuc-extension-provider] セクションに注目してください。Jabberログでこれを検索すると、Microsoft Office Integrationから取得した結果も表示されます。

関連情報

- [Cisco Jabber for Windows のトラブルシューティング](#)
- [テクニカル サポートとドキュメント – Cisco Systems](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。