

CVP 礼儀コールバック (CCB) ゲートウェイ キャパシティ検証エラーを解決して下さい

目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[症状](#)

[トラブルシューティング](#)

[解決策](#)

[永続性のあるソリューション](#)

[テストの最後 ソリューション](#)

概要

トランク ゲートウェイ キャパシティが超過したので発信者が CCB オファーを得ないときこの資料に Cisco Unified Customer Voice Portal (CVP) CCB 問題を解決する方法を記述されています。

前提条件

要件

次の項目に関する知識が推奨されます。

- CVP
- Cisco CVP 礼儀コールバック

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアのバージョンに基づくものです。

- CVP サーバ 10.5
- Unified Contact Center Enterprise (UCCE) 10.5

本書の情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。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。稼働中のネットワークで作業を行う場合、コマンドの影響について十分に理解したうえで作業してください。

背景説明

ゲートウェイ キャパシティの問題は解決される前に、CCB のトランク 検証プロセスを理解することは重要です。基本的には、プロセスは最初に EventTypeID の Callback_current 表からの呼び出しの数を判別します (21,22,23); 特定のゲートウェイおよび場所のための保留中、Inprogress、一時的な。

2 番目に、Callback_current 同じ表から、接続される原因と完了する呼び出しの数判別して下さい: EventTypeID = 24 (完了する)、および CauseID = 27 (接続される)。

最後にプロセスはこれら二つの値を追加し、Survivability.tcl サービスの下で設定されるトランクの数によって比較します。

結果が設定されるトランクしきい値にある場合プロセスは失敗を送返します (1) 戻りは、他では ok (0) 戻りを送返します。

要約すると、CCB に使用するトランクを検証する数式は次のとおりです:

CCB は < トランキングします (EventTypeID の Callback_current 表 (21,22,23);、特定のゲートウェイのための Inprogress 保留中、一時的な) + EventTypeID の Callback_current 表 = 24 (完了する)、および CauseID = 27 (接続されて)

CCB トランク値がより低ければ検証は失敗します。

症状

インバウンドコールは CCB オファーを得ません。コールは推定待ち時間 (EWT) をそれにもかかわらず並べることを直接行きます

トラブルシューティング

ステップ 1. Voice Extensible Markup Language (VXML) サーバからの CallbackEntry アプリケーションからアクティビティ ログを集めて下さい。

ステップ 2. 検証がどれもではないコールをアクティビティ ログの内で捜して下さい:

```
Validate_02,data,result,none
```

検証は渡らなかったことを意味するかどれが。このコールのための GUID を得て下さい。コールをアクティビティ callid によってフィルタリングし、この例のような callid を探して下さい:

```
start,parameter,callid=BBBBAAAACCCDDDDDEEEFFFFFFFAAAAABBBB
```

ステップ 3. レポート サーバーのための CVP レポート ログを集めて下さい。CVP レポート ログの同じ callid を見つけて下さい。

```
ValidateHandler:ValidateHandler.exec: ValidateHandler GUID=BBBBAAAACCCDDDDDEEEFFFFFFFAAAAABBBB  
results:none validation status bitmask=0x00000103
```

ステップ 4. バイナリにビットマスク数を変換して下さい。プログラマ カルキュレータを使用して下さい: **0001 00000011**

ステップ 5 CCB 表があるように CVP 報告 ガイド ビットマスクを確認して下さい。検証が「EXCEED_CAPACITY_GW が理由で」失敗することがわかるはずです。

```
00000000 00000001 OK
00000000 00000010 ICM_NO_SCHEDULED_ALLOWED
00000000 00000100 ICM_NO_PREEMPTIVE_ALLOWED
00000000 00001000 NOT_IN_QUEUE
00000000 00010000 TOD
00000000 00100000 EWT
00000000 01000000 PROBE_FAILED_NO_RESPONSE
00000000 10000000 PROBE_FAILED_NO_CONFIG
00000001 00000000 EXCEED_CAPACITY_GW
00000010 00000000 EXCEED_CAPACITY_QUEUE
```

: ICM_NO_SHCEDULED_ALLOWED

ステップ 6.特定のキューに問題を狭くして下さい。CCB が提供されない特定のキューがあったかどうか確認するために CVP レポート サーバーからの CCB Servlet をチェックして下さい。Webブラウザを開き、入力して下さい。

<http:// {レポート サーバー IP Address}:8000/cvp/CallbackServlet?method=Diag>

これは CCB が提供されるキューの例です:

Queue Name	ICM EWT	DQRate	Reconnect Time	SLA Time	Pending Callbacks	Percent Callbacks	<rwt-ewt>
billing	2000	2000	30	60	0	0	2.67

Position	Number	Callback	Remaining Wait Time	In Queue State	GUID	Entrance Date	Retry	Validation Status	Start Date	Updated	Recording
1	""	false	1986	INQUEUE	9977E700000100000000000612C6C90A	10:26:53 09/01/2017	false	0x0003	10:26:53 09/01/2017	10:26:53 09/01/2017	""

これは CCB が提供されないキューの例です

Queue Name	ICM EWT	DQRate	Reconnect Time	SLA Time	Pending Callbacks	Percent Callbacks	<rwt-ewt>
billing	2000	2000	30	60	0	0	2.67

ステップ 7 キューが特定のゲートウェイによって実行するかどうか確認して下さい。ゲートウェイコンフィギュレーション (サバイバビリティ アプリケーション パラメータ) をチェックして下さい。

```
application
service new-call flash:bootstrap.vxml
!
```

```
service survivability flash:survivability.tcl
paramspace callfeature med-inact-det enable
param ccb id:10.201.198.21;loc:CALO;trunks:512
```

ステップ 8 設定が正しい場合、この特定のゲートウェイおよび位置の呼び出しの数を判別するためにレポート サーバー データベース (Informix) の保存されている情報をチェックして下さい。CCB ID (10.201.198.21 この場合) または locattion (この例の CALO) によってチェックできません。

ステップ 9。 レポート サーバー、アクセス Informixデータベース。

CMD プロンプトを開き、入力して下さい: **dbaccess**

接続へのナビゲートは > **接続します**

cvp 例を選択して下さい

ユーザ名 **cvp_dbadmin** を入力して下さい

パスワードを入力して下さい

callback@cvp データベースを選択して下さい

照会言語に終了し、ナビゲートして下さい

ステップ 10.クエリを実行して下さい:

callback_current から数を (*) ところで位置 == 「CALO」 選択して下さい;

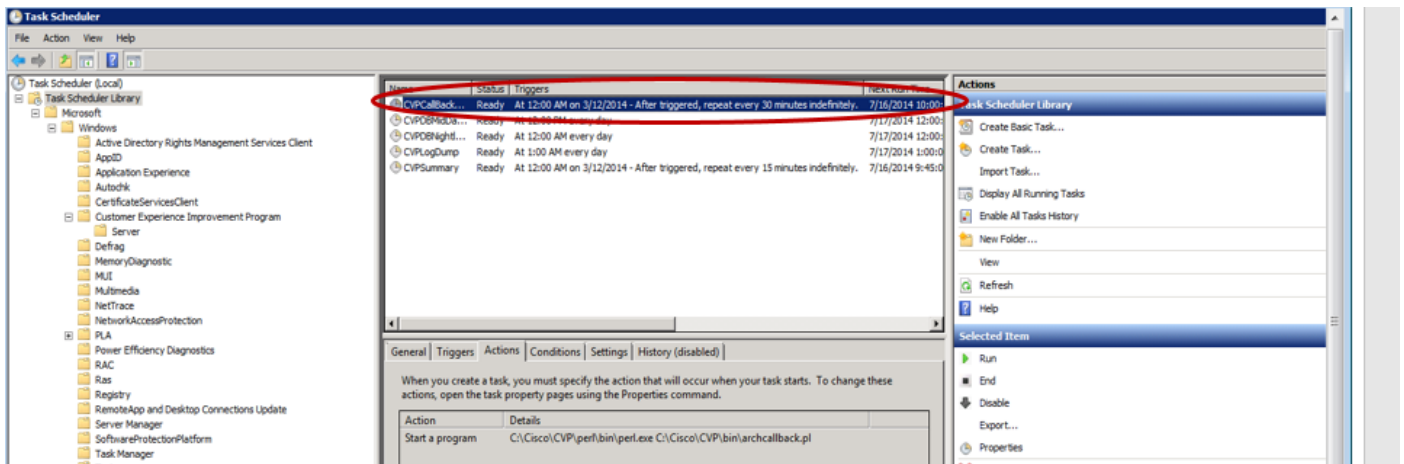
ステップ 11: 値が同じまたは位置のためのゲートウェイで設定されるトランク値より高くである場合これは許可されるトランクの最大数に Callback_Current 表で達したので validation がなぜ失敗するか原因です。

注: CVP 報告 ガイドで参照されるように、コールバック 表は 2 つの表のビューです: Callback_Current および Callback_Historical。 2 つの表は同一です。 30 分毎に、完了された呼び出しのデータは Callback_Pending から引っ張られ、Callback_Historical に移動されます。

手順 12: 位置ごとのトランク値が Callback_Current 表の制限におよび達したら Callback_Current から Callback_Historical 表ことをへコールバック レコードを移動することに問題があることをこれが示すキューにコールバックがありません。

手順 13: CVPCallbackArchive がスケジュール タスク (CVP レポート サーバー) の下で動作していることを確認して下さい。 **Start > Programs > Accessories** へのナビゲート-> **システム ツール-> 定期タスク**。

を探します。



手順 14：このタスク CVPCallbackArchive が完了したら終了コードをあります確認して下さい (0x0)。

Name	Status	Triggers	Next Run...	Last Run Ti...	Last Run Result	Author
CVPCallBack...	Ready	At 12:00 AM on 1/31/2017 - ...	8/30/20...	8/30/2017 4...	The operation completed successfully. (0x0)	Administrator
CVPDBMidD...	Ready	At 12:00 PM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPDBNight...	Ready	At 12:00 AM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPLogDump	Ready	At 1:00 AM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPSummary	Ready	At 12:00 AM on 1/31/2017 - ...	8/30/20...	8/30/2017 4...	The operation completed successfully. (0x0)	Administrator

手順 15：ステップ 13 および 14 が、まだ Callback_Historical 表のデータうまくないが、情報がデータベースになぜ追加されないか判別する必要があります。電流およびヒストリテーブルの保存されている情報の統合をチェックして下さい。Informix dbaccess CMD ウィンドウのこのクエリを実行して下さい：

```
Select count (*) from callback_current where surrogateid in (select surrogateid from callback_historical);
```

ステップ 16。数が 1 またはより高い場合、現在のテーブルの主キーがヒストリテーブルに既にあって、情報がデータベースに追加されないことを意味します。これらのシナリオのほとんどでは、競合状態により重複したレコードは callback_current 表に入ります。

surrogateid マッピングへの GUID はキュー表で起こります。コールがコールバック待機からコールバックキュースクリプトに移る状況では、そこにアーカイブジョブが電流から履歴にレコードを移動するアプリケーションが同じ surrogateid が付いている現在のテーブルで新記録を入力するウィンドウのようで。この問題はこの CDETS [CSCuq86400](#) と関連しています

解決策

ステップ 1. アクセス Informix データベース。CMD プロンプトを開き、入力して下さい: **dbaccess**

ステップ 2. 接続へのナビゲートは > 選定された cvp 例を接続します。型ユーザ名 cvp_dbadmin および型パスワード

ステップ 3. 照会言語への選定された callback@cvp データベース終了およびナビゲート

ステップ 4. これらのコマンドを実行して下さい:

callback_current からの削除と surrogateid (callback_historical からの選定された surrogateid);

一時テーブル エラーがあったら:

表 T1 を廃棄して下さい;

ステップ 5. sp プロシージャを実行して下さい現在のから照会言語 ウィンドウ dbaccess からの歴史的コールバック 表に情報を移動する。

プロシージャ sp_arch_callback() を実行して下さい;

ステップ 6 現在のテーブルに同様に多くのレコードが前にないことを確認して下さい。

callback_current から数を (*) ところで位置 == 「CALO」選択して下さい;

永続性のあるソリューション

ステップ 1. Cisco \ CVP \ informix_frag にナビゲート し、テキストエディタの sp_arch_callback.sql を開いて下さい。

ステップ 2. ファイルの始めにこの行のコメントアウトを解除して下さい: ---プロシージャ sp_arch_callback を廃棄して下さい; (取除いて下さい -- 行の始まり)。

ステップ 3. この行を追加して下さい: 代理をされるところ callback_current からの削除 (callback_historical からの選定された surrogateid); 後

プロシージャ sp_arch_callback() 行を作成して下さい。

ステップ 4. ファイルを保存して下さい。

ステップ 5 これはファイルの最初の一部がどのようにののように見える必要があるか例です。

```
{*****
Stored procedure to move completed calls out of the active table into the
historical table.
*****}
drop procedure sp_arch_callback;
create procedure sp_arch_callback()

DEFINE p_ageoff INTEGER;

-- delete any duplicates found in current table.

delete from callback_current where surrogateid in (select surrogateid from callback_historical);
```

テストの最後 ソリューション

ステップ 1. CMD プロンプトを開き、コマンドを実行して下さい: dbschema

dbschema - d コールバック - f sp_arch_callback

注: dbschema コマンド、ログオンおよび試みをレポート サーバーに cvp_dbadmin として
もう一度実行する場合の許可問題があれば。

呼び出します。出力から、コマンドからの削除が実行されるようにして下さい。

```
C:\Users\Administrator>dbschema -d callback -f sp_arch_callback
DBSCHEMA Schema Utility          INFORMIX-SQL Version 12.10.FC3

create procedure "Administrator".sp_arch_callback()
DEFINE p_ageoff INTEGER;
-- delete any duplicates found in current table.
delete from callback_current where surrogateid in (select surrogateid from callb
ack_historical);
SELECT surrogateid
FROM Callback_current
WHERE EventTypeID in (24,29) -- Completed, Too many callbacks
AND CauseID in (27,28) -- Connected, Cancelled
INTO TEMP t1 WITH NO LOG;
```