

# Обратный вызов любезности CVP устранения неполадок (CCB) сбоя проверки емкости шлюза

## Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Общие сведения](#)

[Признаки](#)

[Устранение неполадок](#)

[Решение](#)

[Постоянное решение](#)

[Решение для завершения тестирования](#)

## Введение

Этот документ описывает, как устранять проблему CCB клиентского голосового портала (CVP), когда абонент не получает предложение CCB, потому что емкость магистрального шлюза превысила.

## Предварительные условия

### Требования

Компания Cisco рекомендует предварительно ознакомиться со следующими предметами:

- CVP
- Обратный вызов любезности CVP Cisco

### Используемые компоненты

Сведения, содержащиеся в этом документе, касаются следующих версий программного обеспечения:

- Сервер CVP 10.5
- Унифицированное предприятие Contact Center (UCCE) 10.5

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

## Общие сведения

Прежде чем проблема емкости шлюза диагностируется, важно понять процесс проверки данных транка в ССВ. В основном процесс сначала определяет количество вызовов от таблицы **Callback\_current** с **EventTypeID** в (21,22,23); Ожидание, Inprogress, Предварительный для определенных шлюзов и местоположений.

Во-вторых, от той же таблицы **Callback\_current**, определите, количество вызовов, завершенных со связанной причиной: **EventTypeID** = 24 (Завершенный), и **CauseID** = 27 (Связанный).

Наконец процесс добавляет эти два значения и соответствует количеству транков, настроенных под сервисом **Survivability.tcl**.

Если результат по настроенному порогу транков, процесс передает сбой обратно (возвратиться 1), иначе передает обратно ok (возвратиться 0).

Таким образом, формула для проверки транков, используемых для ССВ:

Транки ССВ <(таблица **Callback\_current** с **EventTypeID** в (21,22,23); Ожидание, Inprogress, Предварительный для определенных шлюзов) + таблица **Callback\_current** **EventTypeID** = 24 (Завершенный), и **CauseID** = 27 (Связанный)

Если значение Транков ССВ является более низкими сбоями проверки.

## Признаки

Входящий вызов не получает предложение ССВ. Вызов идет непосредственно к очереди независимо Предполагаемое время ожидания (EWT)

## Устранение неполадок

Шаг 1. Соберите Журналы операций из приложения **CallbackEntry** с Речевого Расширяемого языка разметки гипертекста (XML) (VXML) Сервер.

Шаг 2. Поиск в рамках Журналов операций для любого вызова, где проверка не ни один:

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

Что означает, что не прошла проверка. Получите GUID для этого вызова. Фильтруйте вызов теплого действия и ищите теплое как данный пример:

```
start,parameter,callid=BBBBAAAACCCDDDDDEEEFFFAAAABBBB
```

Шаг 3. Соберите журналы создания отчетов CVP для Сервера Создания отчетов. Найдите то же теплым в журналах Создания отчетов CVP.

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

Шаг 4. . Преобразуйте номер битовой маски в двоичные файлы. Используйте калькулятор программиста: **0001 00000011**

Шаг 5. . Проверьте битовую маску Руководства по созданию отчетов CVP для таблиц CCB. Необходимо видеть, что проверка отказывает из-за "EXCEED\_CAPACITY\_GW".

```
00000000 00000001
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\_SCHEDULED\_ALLOWED OK ,

Шаг 6. Сузьте проблему к определенной очереди. Проверьте CCB Servlet от сервера Создания отчетов CVP, чтобы определить, существует ли какая-либо определенная очередь (очереди), где не предлагается CCB. Откройте web-браузер и тип.

<http://{Создание отчетов о IP-адресе сервера}:8000/cvp/CallbackServlet? method=Diag>

Это - пример очереди, где предлагается CCB:

Queue Name	ICM EWT	DQRate	Reconnect Time	SLA Time	Pending Callbacks	Percent Callbacks	<rw-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	""

Это - пример очереди, где не предлагается CCB

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

Шаг 7. Проверьте, подается ли очередь (очереди) определенным шлюзом. Проверьте конфигурацию шлюза (Параметры приложения жизнеспособности).

```
application
service new-call flash:bootstrap.vxml
!
service survivability flash:survivability.tcl
param space callfeature med-inact-det enable
param ccb id:10.201.198.21;loc:CALO;trunks:512
```

Шаг 8. Если конфигурация корректна, проверьте информацию, хранившуюся в Базе данных сервера Создания отчетов (Informix) для определения количества запросов к этому определенному шлюзу и местоположению. Можно проверить идентификатор ССВ (10.201.198.21 в этом случае) или местоположение (CALO в данном примере).

Шаг 9. На сервере создания отчетов обратитесь к Базе данных Informix.

Откройте приглашение cmd и тип: **dbaccess**

Перейдите к **соединению> подключение**

Выберите **свр экземпляр**

введите имя пользователя **свр\_dbadmin**

введите пароль

выберите **callback@свр базу данных**

выйдите и перейдите к Языкам запросов

Шаг 10. Выполните запрос:

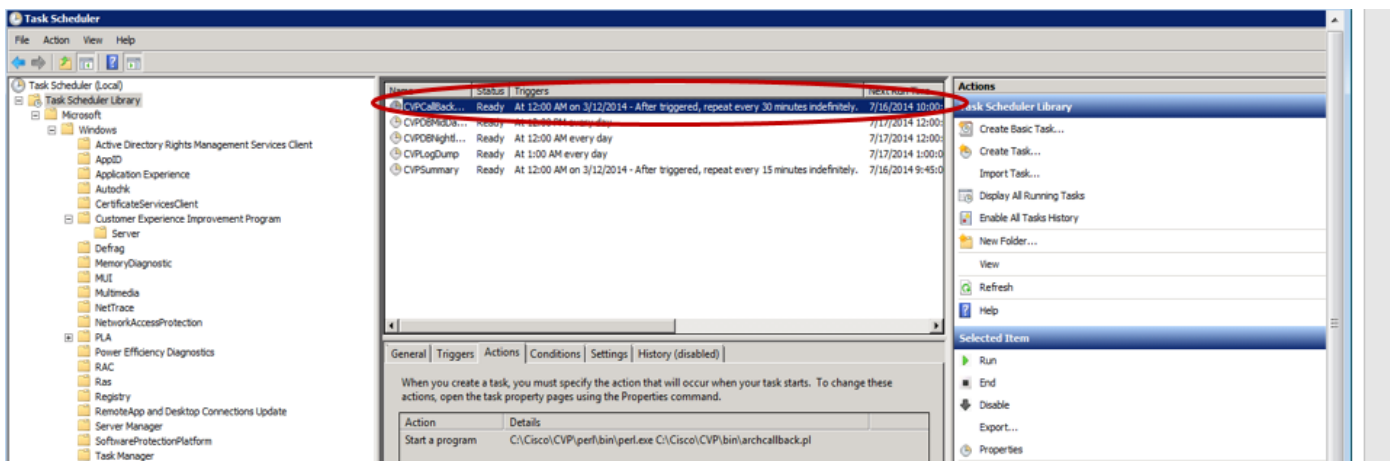
**Выберите количество (\*) от callback\_current где местоположение == "CALO";**

Шаг 11. Если значение является тем же или выше, чем значение транка, настроенное в шлюзе для местоположения (местоположений), это - причина, почему validation отказывает, так как максимальные числа позволенных транков были достигнуты в таблице Callback\_Current.

**Примечание:** Как ссылается в Руководстве по созданию отчетов CVP, таблица Обратного вызова является представлением двух таблиц: Callback\_Current и Callback\_Historical. Эти две таблицы идентичны. Каждые 30 минут данные для состоявшихся соединений вытягивают от Callback\_Pending и перемещают в Callback\_Historical.

Шаг 12. Если значение транка на местоположение достигло своих пределов в таблице Callback\_Current и нет никаких обратных вызовов в очереди, это указывает, что существует проблема в перемещении записей обратного вызова от Callback\_Current до таблицы Callback\_Historical.

Шаг 13. Гарантируйте, что CVPCallbackArchive работает под Задачами Списка (Сервер Создания отчетов CVP). Перейдите для **Начала-> Программы-> Accessories-> Системные средства-> Запланированная задача.**



Шаг 14. Если эта задача, которую выполняет CVPCallBackArchive, гарантирует, что код выхода (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

Шаг 15. Если шаг 13 и 14 прекрасен, но все еще никакие данные в таблице Callback\_Historical, необходимо будет определить, почему информация не добавлена в базе данных. Проверьте целостность информации, хранившей в токе и таблице истории. Выполните этот запрос на informix dbaccess окно CMD:

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

Шаг 16. Если количество равняется 1 или выше, это означает, что первичный ключ на текущей таблице уже существует в таблице истории, и информация не добавлена в базу данных. В большинстве этих сценариев состояние гонки заставляет двойные записи вводить в callback\_current таблицу.

GUID к сопоставлению surrogateid происходит на таблице очереди. В ситуациях, где шаги вызова от обратного вызова ждут к сценарию очереди обратного вызова, кажется, существует окно, куда архивное задание перемещает записи от текущего до истории, и приложение вводит новую запись в текущую таблицу с тем же surrogateid. Эта проблема отнесена к этому CDETS [CSCuq86400](#)

## Решение

Шаг 1. База данных Informix доступа. Откройте приглашение cmd и тип: **dbaccess**

Шаг 2. Перейдите к **соединению>**, **подключение** выбирает srv экземпляр. Введите имя пользователя srv\_dbadmin и введите пароль

Шаг 3. Выберите callback@srv выход базы данных и перейдите к **Языкам запросов**

Шаг 4. . Выполните эти команды:

удалите из callback\_current, где surrogateid в (выбирают surrogateid от callback\_historical);

Если существует ошибка временной таблицы, сделайте:

**таблица t1 отбрасывания;**

Шаг 5. . Выполните процедуру sp, которая перемещает информацию от Текущего до исторической таблицы обратного вызова из окна языка запросов dbaccess.

**ВЫПОЛНИТЕ ПРОЦЕДУРУ sp\_arch\_callback ();**

Шаг 6. Проверьте, что нет столько же записей в текущей таблице сколько прежде.

**Выберите количество (\*) от callback\_current где местоположение == "CALO";**

## **Постоянное решение**

Шаг 1. Перейдите к Cisco\CVP\informix\_frag и откройте sp\_arch\_callback.sql в текстовом редакторе.

Шаг 2. Не прокомментируйте эту линию в начале файла: - отбрасывают процедуру sp\_arch\_callback; (удалите - в начале линии).

Шаг 3. Добавьте эту линию: удалите из callback\_current, где свернутый в (выберите surrogateid от callback\_historical); после

**создайте процедуру sp\_arch\_callback () линия.**

Шаг 4. . Сохраните файл.

Шаг 5. . Это - пример о том, как первая часть файла должна быть похожей.

```
{*****
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);
```

## **Решение для завершения тестирования**

Шаг 1. Откройте приглашение cmd и выполните команду: **dbschema**

**dbschema-d обратный вызов-f sp\_arch\_callback**

**Примечание:** Если вы имеете проблему авторизации при выполнении dbschema команды, входите как svr\_dbadmin в сервер создания отчетов и попытку еще раз.

Шаг 2. От выходных данных гарантируйте, что выполняется **Удаление** из команды.

```
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;
```