

Понимание и Устранение проблем Оперативных Данных сообщают относительно UCCX

Содержание

[Введение](#)

[Изменения в оперативных данных с SocketIO](#)

[Технический обзор](#)

[WebSocket по сравнению с длинным механизмом опроса](#)

[WebSocket \(RFC 6455\)](#)

[Долго опрос](#)

[Контрольное число Клиентов, выполняющих Долго Опрос](#)

[Оперативный Поток данных](#)

[Серверные SocketIO и Темы](#)

[Производитель](#)

[Диспетчер](#)

[Клиенты](#)

[Authentication](#)

[Оперативное отслеживание данных](#)

[Устранение проблем оперативных данных](#)

[Основной Healthcheck](#)

[Сводка](#)

[Проверьте версию браузера](#)

[Проверьте ЦП и Использование памяти](#)

[Проверьте, что Оперативный источник Поточковой передачи данных является Онлайновым](#)

[Проверьте NTP.](#)

[Проверьте DNS](#)

[Выполните диагностический тест](#)

[Проверьте имя хоста](#)

[Проверьте конфигурацию VM](#)

[Проверьте сертификаты UCCX](#)

[Проверьте поддерживаемую конфигурацию калибровки](#)

[Известные предупреждения](#)

[Оперативное отслеживание журнала данных](#)

[Журналы механизма](#)

[Журналы SocketIO \(1\)](#)

[Журналы SocketIO \(2\)](#)

[Клиентские журналы \(1\)](#)

[Журналы SocketIO](#)

[Клиентские журналы \(2\)](#)

Введение

Этот документ описывает Оперативные Данные, сообщающие в Унифицированном Contact Center Express (UCCX) решение, запускающееся с версии 11.x и структур, устраняющих неполадки контрольных точек наряду с поддержкой регистрационного анализа. Статья также выделяет различные известные предупреждения с их обходными путями.

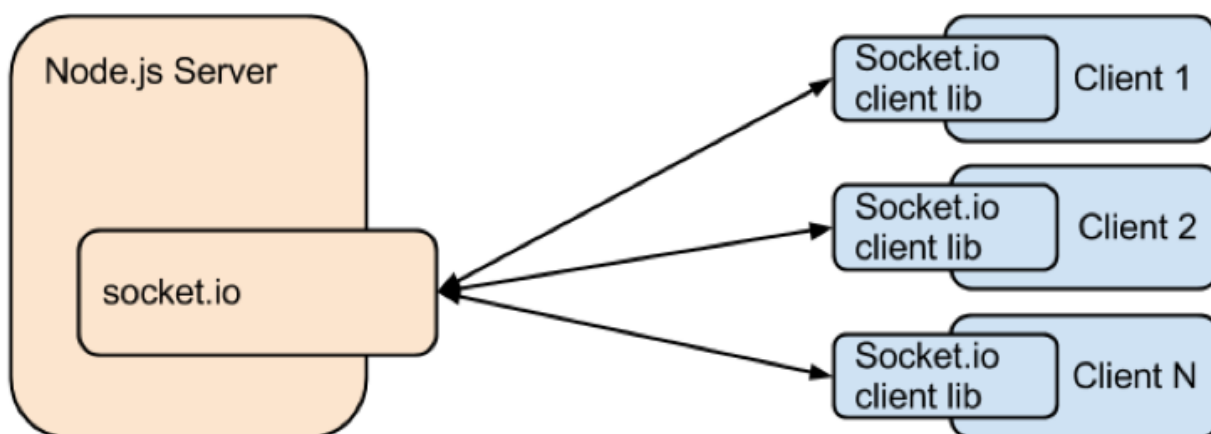
Изменения в оперативных данных с SocketIO

SocketIO базировал Оперативные замены Дизайна Данных Openfire XMPP сервер как по умолчанию, сообщая об инфраструктуре сообщения от UCCX 11.x и далее. Это - событийно-управляемое масштабируемое и эффективное решение, которое основывается на Node.js implementation с уменьшенными издержками связи при помощи протокола WebSocket Request For Comments 6455 (RFC 6455) для передачи оперативных сообщений к браузеру (клиенты).

Ранее в UCCX 10.x, был одиночный **Сервис уведомлений CCX Cisco**, на основе реализации OpenFire XMPP), который использовался и для Изящества, Чата, Почтовых уведомлений, а также Оперативных отчётов о Данных относительно гаджетов Агента Изящества и для CUIC.

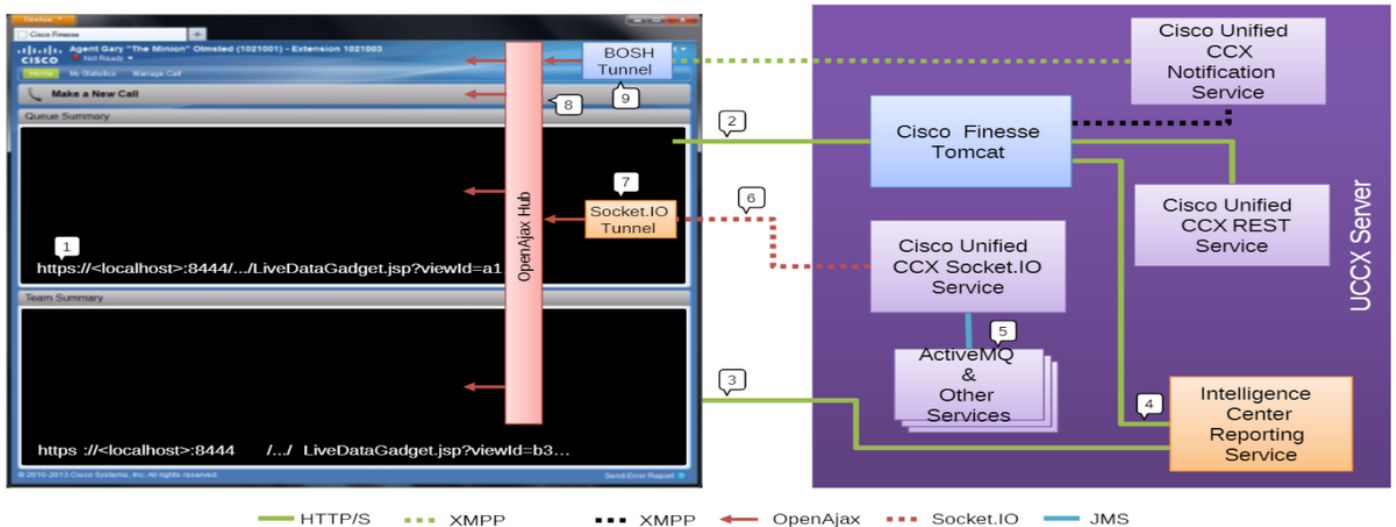
От UCCX 11.x, с введением SocketIO для Репортажей с места события - Чат Клиента Изящества и Электронная почта продолжают использовать **Сервис уведомлений CCX Cisco**, т.е. Openfire как Сервер XMPP, но Оперативные Данные, сообщая о модуле используют новую **Cisco Унифицированный Сокет CCX. Сервис Сервиса IO**.

Это уменьшает Оперативные издержки создания отчетов Данных и делает SocketIO исключительно ответственным за составление и диспетчеризацию Оперативных обновлений Данных клиентского браузера, как показано в образе сервера SocketIO.



Технический обзор

Этот образ имеет основанный Оперативный Обзор архитектуры Данных SocketIO.



С точки зрения высокого уровня данные для Оперативных Отчётов о Данных передаются в Объектной нотации JavaScript (JSON) от модуля Менеджера данных реального времени (RTDM) Механизма UCCX, через шину Службы обмена сообщениями Java (JMS) (используйте Apache ActiveMQ для реализации этого, т.е. Активной Очереди Message), к Сокету. Сервер IO, размещенный на UCCX (представленный через Унифицированный Сокет CCX. Сервис IO), который тогда получает эту информацию, процессы (использует) их в комнаты и затем распределяет (отправляет) это клиентам, использующим соединения HTTPS WebSocket по TCP.

ActiveMQ является промежуточным программным обеспечением, которое внедряет Прикладной программный интерфейс (API) JMS, и это используется для передачи сообщений между двумя компонентами, UCCX Оперативные Данные (Механизм UCCX) и сервер SocketIO.

WebSocket по сравнению с длинным механизмом опроса

Существует два способа, которыми браузеры могут подписаться для получения Оперативных обновлений данных, с помощью WebSocket (WS)/WebSocket Безопасная технология (WSS), или при помощи Длинного Механизма опроса.

WebSocket (RFC 6455)

- Протокол WebSocket (RFC 6455) является независимым политиком на основе TCP протокол. Его единственное отношение к HTTP - то, что его квитирование интерпретируется серверами HTTP как запрос Обновления и используется для установления постоянного соединения с Сервером для непрерывных обновлений.
- После успешного квитирования WebSocket клиенты и серверы передают данные назад и вперед в меньших концептуальных модулях (объекты JSON) упомянутый в этой спецификации как **сообщения**. Сообщение WebSocket не обязательно соответствует формированию кадров уровня индивидуальной сети, поскольку фрагментированное сообщение может быть объединено или разделено посредником.

- Это process start с клиентом, передающим обычный запрос HTTP к серверу.
- Заголовок Обновления включен в этот запрос, который сообщает серверу, что клиент хочет установить соединение WebSocket. т.е. **HTTP/1.1 101 Обновление Протоколов переключения: Соединение websocket: Обновление.**
- После этого комната открыта, и сообщения отправляются постоянно Клиенту/браузеру.
- WebSocket, как известно, уменьшает издержки обработки на сервере - публикаторе и предоставляет безопасные и динамические оперативные обновления.

Долго опрос

- Это - изменение традиционного способа опроса и позволяет эмуляцию информационного толчка от сервера до клиента.
- С длинным опросом, информацией о запросах клиента от сервера похожим способом к обычному опросу. Соединение считается открытым между web - клиентом и Web-сервером так, чтобы, когда сервер имеет новую информацию, это могло выдвинуть его клиенту, и соединение тогда закрыто.
- Если сервер не имеет никакой доступной информации для клиента, вместо того, чтобы передать пустой ответ, сервер держит запрос и ждет некоторой информации, чтобы быть доступным.
- Как только информация становится доступной (или после подходящего таймаута), завершённый ответ передается клиенту.
- Клиент обычно тогда сразу повторно запрашивает информацию от сервера, так, чтобы сервер всегда имел доступный запрос ожидания, что это может использовать для отправки данных в ответ на событие.
- Долго опрашивая причины дополнительные издержки на сервере, который запрашивают на эту информацию.

Контрольное число Клиентов, выполняющих Долго Опрос

- Для USSX Оперативные Данные **WebSocket** благоприятен и рекомендован по традиционному длинному механизму опроса.
- Чтобы проверить, используют ли клиенты WebSocket OR Длинный Механизм опроса для оперативных данных, можно проанализировать отладки SocketIO и видеть сообщения.

Шаги:

- Включите Сокетные отладки IO для выравнивания, DEBUG (от по умолчанию ПРЕДУПРЕЖДАЮТ) - перешли к **Удобству обслуживания USSX> Трассировка.**

- Выследите последние журналы сервера SocketIO:

Появляется следующее:

E.g: IE compatibility mode -

TimeStamp <LOG message> (POLLING): ...

```
0000000135: 10.78.91.238: Nov 28 2016 20:19:44.297 +0530: %CCBU_nioEventLoopGroup-5-1-6-  
MessageDispatcher: %[address=/10.107.11.107:51809(WEBSOCKET)][message=Subscribe-  
{AgentCSQStats=[Email_CSQ]}][socket_io_server_type=WSS]: Subscription Request
```

E.g: Firefox/Chrome/IE Native with HTML5/websocket enabled:

TimeStamp <LOG message> (WEBSOCKET): ...

Оперативный Поток данных

Оперативный Поток данных может быть суммирован как: **Механизм CCX (ICD_RTDM, SS_RMCM) сервер JMS ActiveMQ SocketIO → Гаджеты Клиентов (Web-браузер)**

Это средства регистрации, которые должны быть включены на Механизме UCCX:

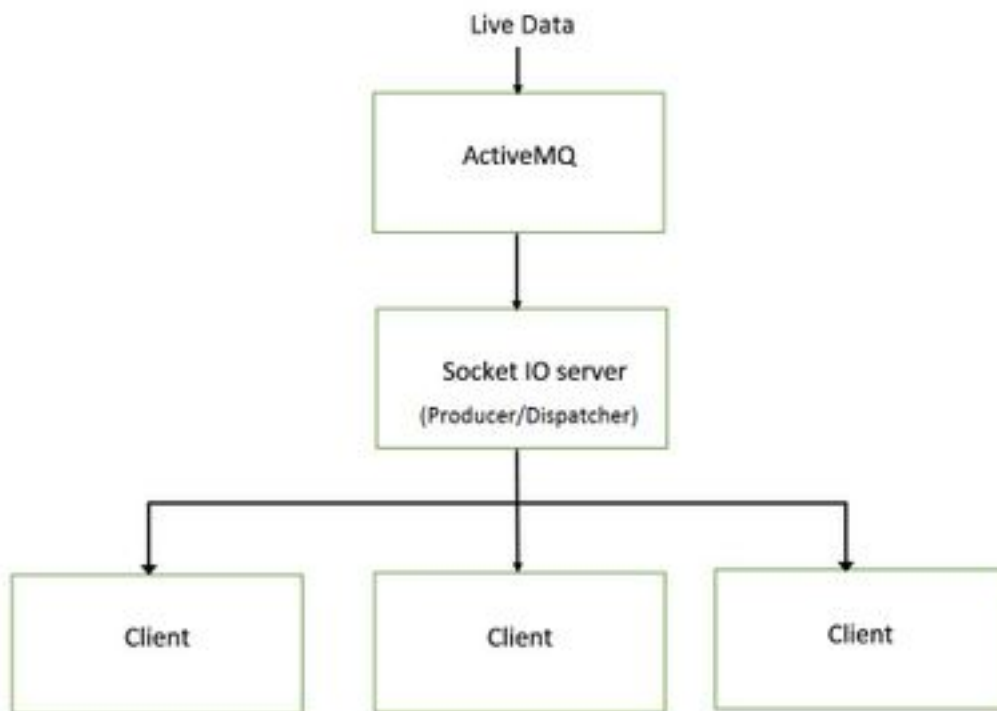
ICD_RTDM = менеджер данных реального времени распределения интеллектуального вызова

SS_RMCM = менеджер ресурсов подсистемы менеджер контакта

JMS ActiveMQ = Сервис обмена сообщениями Java Активная реализация Очереди Обмена сообщениями

Оперативные Данные сервер SocketIO разработаны прежде всего, чтобы получить информацию от активного источника данных (такого как JMS) и публиковать его, через WebSocket, Жить Клиенты Данных (как Клиентский Гаджет на Изяществе или Web - приложении Cisco Unified Intelligence Center (CUIC).

Они логические схематичный описывают поток данных от Механизма UCCX (Оперативные Данные) в ActiveMQ, где Сокет. Сервер IO тогда получает информацию и распределяет, это клиентам, как показано в образе SocketIO базировало клиентские подписки :



Оперативные Данные публикуют события в группах по темам через ActiveMQ.

Сервер SocketIO подписывается на эти темы и получает группы события.

Серверные SocketIO и Темы

Сервер SocketIO настроен с различными комнатами или темами, определенными в `/opt/cisco/uccx/socketioserver/conf/socketioservice.properties` файле.

Все Оперативные отчёты о данных сопоставлены с темой JMS, которая полезна при парсинге журналов.

Обновление темы JMS происходит за каждые 3 секунды для определенных отчётов, это основанное событие (Обновление зарегистрировано только, когда событие инициировано, как Изменение состояния агента).

Они перечислены в таблице Отчёта сопоставлению Темы JMS:

Название	Название темы
отчёта	JMS
Статистический отчет CSQ агента	AgentCSQStats----> Событие Базирующа Тема
Отчёт о журнале состояния агента	AgentStateDetailStats----> Событие Базирующ Тема
Статистический отчет агента	ResourceAQStats
Сводный отчет группы агентов	ResourceAQStats
Отчёт о Состоянии команды	ResourceAQStats

Сводный отчет команды
Речевые сведения отчета CSQ
Речевая сводка CSQ

Сводный отчет CSQ чата

Статистический отчет агента чата

Пошлите по электронной почте статистический отчет агента

Почтовый сводный отчет CSQ

ResourceIAQStats
VoiceCSQDetailsStats
VoicelAQStats

ChatQueueStatistics---> Событие Базирующа
Тема

ChatAgentStats---> Событие Базирующая
Тема

EmailAgentStats---> Событие Базирующая
Тема

EmailQueueStatistics---> Событие Базирующа
Тема

Клиенты как сервер CUIС соединяются с Сокетом. Сервер IO для подписки на комнаты, упомянутые выше для получения оперативных событий данных от сервера SocketIO.

Запрос подписки отправлен от CUIС как скоро, отчёт запущен от CUIС. Например, запуск **Речевого Сводного отчета CSQ** на CUIС инициирует запрос подписки в соответствующую комнату/тему (VoicelAQStats) к серверу SIO.

Производитель

1. Получение события обрабатывает в пакетном режиме от ActiveMQ.
2. Разламывает их на их дискретные события.
3. Быстрые взгляды в каждом событии для получения идентификатора события.
4. Направляет событие в комнату на основе темы, объединенной с полученным идентификатором.

Диспетчер

Диспетчер получает дискретную комнату, названную событиями от производителя, и передает эти события всем клиентам, которые подписываются на ту комнату.

Клиенты

1. Находящиеся на web клиенты (CUIС гаджеты Агентов/Супервизора Изящества OR) соединяются с сервером SocketIO по веб-сокету.
2. После того, как связанный, эти клиенты подписываются на комнаты в Сокетном сервере IO.
3. Несколько клиентов могут подписаться на ту же комнату.

4. Клиенты получают одно событие за один раз.

Пример:

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

Сервер SocketIO тогда начинает публиковать события в клиенте через эту подписку.

Authentication

Сокет. Сервер IO должен аутентифицировать соединяющихся клиентов (как Изящество Оперативный Гаджет Данных OR CUIC) прежде, чем передать данные.

1. Сокет. Сервер IO предлагает использовать основанную аутентификацию маркера.
2. Оперативные Гаджеты Данных выполняют запрос начальной аутентификации, который вручает им Подлинный маркер.
3. Это далее передано Сокету. Сервер IO для аутентификации создания WSS (Сервер WebSocket) соединение. Для этого, Сокета. IO называет API предоставленным UCCX маркером, который это получило от гаджетов.
4. Вся аутентификация сделана с помощью строки маркера. Аутентификация клиента включена в Сокете. Сервис IO через Клиента. Authentication. Отдых. Свойство URL.
5. Все позволили Соединиться, клиенты должны предоставить эту строку в качестве параметра на URL соединения.
6. Клиенты для Снабжения сокетом. Подключение сервера IO, подпишите, отмените подписку, и разъединение. Первоначальное подключение может быть просто соединением или также включать подписку.

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

ws = websocket

wss = websocket безопасный

keystore, который содержит связанные сертификаты для WSS (UCCX использует по

умолчанию: сертификаты приложения Cisco Tomcat)

Местоположение пароля для WSS (по умолчанию: Tomcat Системы)

Оперативное отслеживание данных

Для регистрации существует 3 ключевых точки потока в UCCX Оперативное использование Данных архитектура SocketIO на 11.x (**Механизм - MIVR_RTDM, SS_RMCM Механизма> JMS ActiveMQ> Сервер SocketIO**)

- Журналы Механизма UCCX (MIVR с RTDM и набором RMCM к Xdebugg5) передача сообщения к JMS (Каждые 3 секунды, как замечено в Подсистеме RM отлаживает на Механизме (MIVR) SS_RMCM),
- Журналы SocketIO (регистрация сервиса SocketIO должна быть установлена в полную отладку, куда JMS передает сообщения потребителю SocketIO, который получает все сообщения, публикуемые Механизмом).
- Console log браузера (F12): SocketIO, обрабатывающий Входящие обновления, создавая соответствующие комнаты для каждой темы, и затем передавая их к Вебам - обозревателям клиентов через диспетчера сообщения. (Они замечены в Web-браузере журналы Console/F12).

Эти трассировки должны быть разрешены для наблюдения подробного, переданного в журналах.

1. Механизм UCCX: **ICD_RTDM, SS_RMCM** с уровнем 5 Xdebugging
2. Регистрация Сервиса SocketIO: Уровень отладки.
3. Console log браузера (F12) для проверки входящих Оперативных обновлений Данных.

Отладка SocketIO регистрирует, только говорят, что статус тем любит подключенный/разъединенный и подписался/отменил подписку.

Подробное отслеживание должно быть включено для наблюдения определенных событий по теме. Производитель SocketIO и журналы диспетчера доступны только после включения подробного отслеживания (Свяжитесь с Центром технической поддержки Cisco для включения подробного отслеживания).

Устранение проблем оперативных данных

Основной Healthcheck

Первый шаг к решению Оперативных проблем Данных должен установить срок поддерживаемой конфигурации и выполнить основной медицинский осмотр системы.

Проблемы, найденные в этой конфигурации контрольные точки, могут вызвать проблемы в Оперативном рендеринге Данных на Гаджете Изящества OR на веб-странице CUIC.

Необходимо исключить все эти точки, прежде чем продолжение журнала базировало устранение проблем.

Сводка

- Версия браузера должна быть совместимой.
- ЦП и Использование памяти на UCCX должны быть в определенных рамках.
- Оперативный источник данных Данных должен быть онлайн-овым на странице CUIC Datasources.
- NTP и DNS на UCCX должны быть в порядке с NTP в синхронизации и форвардом/обратными просмотрами DNS в правильном порядке.
- VM OVA должен быть правильно настроен с VRAM и vCPU требованиями по данным Cisco OVA рекомендации.
- NIC в использовании для VM должен быть VMXNET3.

Проверьте версию браузера

Проверьте поведение Оперативных Данных через другие браузеры и версии браузера.

UCCX в настоящее время поддерживает следующие версии браузера, запускающиеся 11.5:
[http://docwiki.cisco.com/wiki/Unified_CCX_Software_Compatibility_Matrix_for_11.5\(1\)#Supported_Browsers](http://docwiki.cisco.com/wiki/Unified_CCX_Software_Compatibility_Matrix_for_11.5(1)#Supported_Browsers)

Не забудьте ОТКЛЮЧАТЬ Режим совместимости на IE, так как это не поддерживается для Оперативных данных. IE переключается на Длинный Механизм опроса, когда в длинном опросе, таким образом, это - важное рассмотрение.

Проверьте ЦП и Использование памяти

Проверьте использование ЦПУ и использование памяти на сервере с помощью RTMT или команд CLI.

Например: RTMT>

CLI: покажите ЦПУ загрузки процесса

CLI: покажите память загрузки процесса

Проверьте для этих процессов при просмотре Оперативных Данных на CUIC: UCCX_Engine, socketioservice, cuicreporting.

Проверьте, что Оперативный источник Поточковой передачи данных является Онлайн-овым

Чтобы проверить, что Оперативный источник Поточковой передачи данных является онлайн-овым, перейдите к **Веб-странице CUIC> Система> Источники данных**

Проверьте NTP.

Проверьте, что NTP доступен, и UCCX синхронизируется с сервером NTP в стране <=5

CLI: статус ntp utils

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

синхронизируемый с сервером NTP (xx.xx.xx.xx) в стране 2

Проверьте DNS

Проверьте, что решает DNS.

<Hostname> сетевого хоста CLI:utils

Локальное разрешение:

host name. доменное имя решает локально к xx.xx.xx.xx

Внешнее разрешение:

host name. доменное имя имеет адрес xx.xx.xx.xx

Выполните диагностический тест

Можно выполнить диагностический тест от CLI для определения любой другой потенциальной проблемы с сервером (Например, задержка Tomcat, сетевые проблемы, и т.д.)

CLI:utils диагностируют тест

Проверьте имя хоста

Имя хоста UCCX должно меняться в нижнем регистре.

Обратитесь Установку/Руководство по обновлению и этот дефект.

<http://cdets.cisco.com/apps/dumpcr?identifier=CSCva75058&content=summary&format=html>

Проверьте конфигурацию VM

Проверьте, что корректный шаблон Открытого устройства виртуализации (OVA)

используется согласно этим ссылкам:

https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/virtualization/virtualization-cisco-unified-contact-center-express.html#11.5

https://программное_обеспечение.cisco.com/download/type.html?mdfid=286287033&flowid=76362

Адаптер сети VM UCCX (NIC) установка должен быть **VMXNET3**.

Если корректный адаптер не выбран тогда, UCCX не публикует данные реального времени на устойчивой скорости.

Если значение Адаптера сети VM не является VMXNET3, тогда идут по этой ссылке для исправления параметров настройки адаптера: http://docwiki.cisco.com/wiki/Virtualization_for_Cisco_Unified_Contact_Center_Express

Проверьте сертификаты UCCX

Несоответствие сертификата или когда сертификаты не приняты, может заставить оперативный источник потоковой передачи данных идти разъединенный/офлайнный.

- Примите сертификат, когда предложено.
- Гарантируйте, что не истекают сертификаты третьей стороны.
- Проверьте, что имя хоста в сертификате корректно. В противном случае придерживитесь ниже процедуры.

На стороне ССХ:

1. Отредактируйте этот файл/`usr/local/platform/conf/platformConfig.xml` и исправьте имя хоста и удостоверьтесь, что Записи DNS согласно названию в этом файле.
2. Удалите и восстановите сертификаты после проверки вышеупомянутых подробных данных о кластере UCCX использование команд CLI:
CLI: свидетельство набора regen tomcat

3. Перезапустите Сервис Cisco Tomcat на UCCX.
CLI: сервис utils перезапускает Tomcat Cisco

Больше на сертификатах:

<http://www.cisco.com/c/en/us/support/docs/customer-collaboration/unified-contact-center-express/118855-configure-uccx-00.html>

Проверьте поддерживаемую конфигурацию калибровки

Гарантируйте, что пределы максимальной конфигурации придерживаются к на настройке. Для этого обратитесь Руководство по дизайну Уровня Решения UCCX для максимальной конфигурации сервера и пределов:

UCCX 11.5: http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cust_contact/contact_center/crs/express_11_5/desi...

UCCX 11.0: http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cust_contact/contact_center/crs/express_11_0/desi...

Для подтверждения этого, можно выполнить эти команды CLI, проверив калибровку рекомендации, на настройке для получения точного номера.

ТРЕБУЕМАЯ ИНФОРМАЦИЯ

Общее количество нет. из Супервизоров, назначенных на Команды

Общее количество нет. из настроенных Супервизоров

Общее количество нет. из настроенных Агентов

Общее количество нет. из настроенных Команд

Нет. из агентов на команду

Нет. из супервизоров на команду

Количество пользователей ролью (CUIC)

Количество настроенных CSQ

ЗАПРОС

выполните `uccx sql, db_cra` выбирают количество (*) от (выберите отдельный `resourceloginid` от супервизора где активный = 't'),

выполните `uccx sql, db_cra` ВЫБИРАЮТ количество (*) от ресурса, где активный = 't' и `resourcetype=2`

выполните `uccx sql, db_cra` ВЫБИРАЮТ количество (*) от ресурса, где активный = 't' и `resourcetype=1`

выполните `uccx sql, db_cra` ВЫБИРАЮТ количество (*) от Команды, активный = 't'

выполните `uccx sql, db_cra` ВЫБИРАЮТ `team.teamname`, количество (resource.resourceid) команда ресурс AS NumOfAgents FROM ОБЪЕДИНЕНИЯ ВНУТРЕННЕГО ON (включено) команда `teamid=resource.assignedTeamID` где `resource.active = 't'` и `team.active` GROUP BY `teamname`

выполните `uccx sql, db_cra` ВЫБИРАЮТ `team.teamName`, количество (Supervisor.recordID) супервизор команда AS NumOfSupervisors FROM ОБЪЕДИНЕНИЯ ВНУТРЕННЕГО ON (включено) Супервизор `managedTeamID=team.teamid` где `Supervisor.active = 't'` и `team.active` GROUP BY `teamName`

работайте `sql` выбирают `g.name` как имя группы, количество (*) как `groupsize` от `cuic_data:cuicgroup g, cuic_data:cuicgroupmember gm` где `gm.groupid` группа `g.name`

выполните `uccx sql, db_cra` выбирают количество (*) от `contactservicequeue`, где активный = 't'

Известные предупреждения

[CSCva13838](#) : Оперативная задержка данных на CUIC

[CSCus17605](#) : Задержитесь из-за чрезмерных разрешений создания отчетов к Супервизорам (высокая нагрузка)

[CSCux33949](#) : Оперативные Данные выходят из-за Расширенных Лицензий.

[CSCvb67761](#) : Отчёт о Журнале Состояния агента, причиняющий Высокую загрузку CPU, Влияющую на Оперативные Данные.

[CSCva95411](#) : Высокая загрузка CPU сервиса SocketIO из-за объектов Пространства имен, берущих чрезмерную память кучи.

[CSCvb75279](#) : Оперативный источник Потокowej передачи данных, идущий оффлайн во время к из Синхронизирующего MIVR пароль Keystore

[CSCvc45189](#) : Оперативный источник Потокowej передачи данных, оффлайн должный Резервировать и Восстановить операцию

[CSCut04158](#) : Оперативные Данные выходят из-за неправильного NIC - адаптера

Примечание: Чтобы подтвердить или обойти вышеупомянутые предупреждения, включите Центр технической поддержки Cisco путем открытия запроса на обслуживание.

Оперативное отслеживание журнала данных

Журналы механизма

- Оперативные Данные публикуют события в группах по темам через Активный MQ.
- Сервер SocketIO подписывается на эти темы и получает группы события.

Журналы MIVR: Оперативные Данные передаются от механизма до сообщения JMS по темам (Например: VoiceCSQDetailsStats, VoiceIAQStats, ResourceIAQStats и т.д.)

Для тем события Based (ChatAgentStats, ChatQueueStatistics AgentCSQStats, AgentStateDetailStats) механизм передает пустые сообщения JMS после каждых 3 секунд для поддержания комнаты SocketIO.

Эти трассировки ageseen только после разрешения этих трассировок: [SocketIO Подробное Отслеживание](#)

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType": "VALUELIST", "name": "VoiceIAQStats.esd  
Name", "operator": "SetValues", "value": [{"key": "cssCsq", "desc": "cssCsq"}], "valuelistId": "01FB2C011  
0000133771FC3C33F57F543", "isKeyField": true}] report-layer.js:1414:11
```

Журналы SocketIO (1)

Получение события SocketIO обрабатывает в пакетном режиме от Активного MQ

Когда группы событий поступают, сервер SocketIO делает это:

- Разламывает их на их дискретные события.
- Быстрые взгляды в каждом событии для получения идентификатора события.
- Направляет событие в комнату на основе темы, объединенной с полученным идентификатором

Дискретные события с Названиями Помещения ставятся в очередь на Диспетчере

Производитель **Примечание:** SocketIO, который имеет слушателя/потребителя JMS , получает данные от шины JMS по темам через ActiveMQ и передает его к комнатам SocketIO, которые ставятся в очередь диспетчеру.

Эти трассировки замечены только после разрешения подробных трассировок на SocketIO (свяжитесь с Центром технической поддержки Cisco для включения того же).

Тема Пример 1: VoiceIAQStats.

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

Тема Пример 2: VoiceCSQDetailsStats.

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

Тема Пример 3: ResourceIAQStats

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

Журналы SocketIO (2)

Сокетный диспетчер IO: получает события от производителя и предоставляет клиентам

Диспетчер получает дискретную комнату, названную событиями от производителя, и передает эти события всем клиентам, которые подписываются на ту комнату.

Эти трассировки замечены только после разрешения подробных трассировок на SocketIO (свяжитесь с Центром технической поддержки Cisco для включения того же).

Тема Пример 1: VoiceIAQStats.

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

Тема Пример 2: VoiceCSQDetailsStats.

1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= **Subscribing for rooms:**
<VoiceIAQStats=cssCsq> report-layer.js:1414:11

1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString
[{"fieldId":"720298FB10000140000000A10A4E5E6F","fieldType":"VALUELIST","name":"VoiceIAQStats.esd
Name","operator":"SetValues","value":[{"key":"cssCsq","desc":"cssCsq"}],"valuelistId":"01FB2C011
0000133771FC3C33F57F543","isKeyField":true}] report-layer.js:1414:11

Тема Пример 3: ResourceIAQStats

1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= **Subscribing for rooms:**
<VoiceIAQStats=cssCsq> report-layer.js:1414:11

1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString
[{"fieldId":"720298FB10000140000000A10A4E5E6F","fieldType":"VALUELIST","name":"VoiceIAQStats.esd
Name","operator":"SetValues","value":[{"key":"cssCsq","desc":"cssCsq"}],"valuelistId":"01FB2C011
0000133771FC3C33F57F543","isKeyField":true}] report-layer.js:1414:11

Клиентские журналы (1)

Клиентская Регистрация показывает Соединение, события Subscribing и Receiving.

1. Находящиеся на web клиенты соединяются с сервером как WebSocket или Долго
Опрос клиентов.
2. После того, как связанный, или как часть подключения этих клиентов подписываются
на комнаты в Сокетном сервере IO.
 - Несколько клиентов могут подписаться на ту же комнату.
 - Каждая комната только получит/получит события того типа
3. Клиенты получают 1 событие за один раз.

Выполнение отчёты означает подписываться на события от сервера SIO.

Запуск отчёт CUIС инициирует запрос подписки к теме, соответствующей отчёту серверу
SocketIO.

Пример 1: CUIС отправляющий запроса подписки после рабочего Речевого Сводного отчета CSQ (Тема VoiceIAQStats)

1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= **Subscribing for rooms:**
<VoiceIAQStats=cssCsq> report-layer.js:1414:11

1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString
[{"fieldId":"720298FB10000140000000A10A4E5E6F","fieldType":"VALUELIST","name":"VoiceIAQStats.esd
Name","operator":"SetValues","value":[{"key":"cssCsq","desc":"cssCsq"}],"valuelistId":"01FB2C011
0000133771FC3C33F57F543","isKeyField":true}] report-layer.js:1414:11

Пример 2: CUIС отправляющий запроса подписки после рабочих Речевых Сведений отчета Агента CSQ (тема VoiceCSQDetailsStats)

1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= **Subscribing for rooms:**
<VoiceIAQStats=cssCsq> report-layer.js:1414:11

1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString


```
[{"fieldId":"720298FB10000140000000A10A4E5E6F","fieldType":"VALUELIST","name":"VoiceIAQStats.esd Name","operator":"SetValues","value":[{"key":"cssCsq","desc":"cssCsq"}],"valuelistId":"01FB2C0110000133771FC3C33F57F543","isKeyField":true}] report-layer.js:1414:11
```

Пример 3: CUIС отправляющий запроса подписки после рабочего Статистического отчета Агента (Topic:ResourceIAQStats)

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F","fieldType":"VALUELIST","name":"VoiceIAQStats.esd Name","operator":"SetValues","value":[{"key":"cssCsq","desc":"cssCsq"}],"valuelistId":"01FB2C0110000133771FC3C33F57F543","isKeyField":true}] report-layer.js:1414:11
```

Сервер SocketIO получил запрос подписки, и ПРИСОЕДИНИТЕСЬ к комнате.

Журналы SocketIO

Пример 1: Сервер SIO ПРИСОЕДИНЯЕТСЯ после получения запроса подписки (тема VoiceIAQStats)

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F","fieldType":"VALUELIST","name":"VoiceIAQStats.esd Name","operator":"SetValues","value":[{"key":"cssCsq","desc":"cssCsq"}],"valuelistId":"01FB2C0110000133771FC3C33F57F543","isKeyField":true}] report-layer.js:1414:11
```

Пример 2: Сервер SIO ПРИСОЕДИНЯЕТСЯ после получения запроса подписки (тема VoiceCSQDetailsStats)

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F","fieldType":"VALUELIST","name":"VoiceIAQStats.esd Name","operator":"SetValues","value":[{"key":"cssCsq","desc":"cssCsq"}],"valuelistId":"01FB2C0110000133771FC3C33F57F543","isKeyField":true}] report-layer.js:1414:11
```

Пример 3: Сервер SIO ПРИСОЕДИНЯЕТСЯ после получения запроса подписки (тема ResourceIAQStats)

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F","fieldType":"VALUELIST","name":"VoiceIAQStats.esd Name","operator":"SetValues","value":[{"key":"cssCsq","desc":"cssCsq"}],"valuelistId":"01FB2C0110000133771FC3C33F57F543","isKeyField":true}] report-layer.js:1414:11
```

Клиентские журналы (2)

Сервер SocketIO после получения запроса подписки, начинает передавать события через диспетчера сообщения. Ниже обновления события, которые получены на браузере.

Журнал браузера: Обновление получено на браузере (нажмите F12 на браузере для наблюдения обновления),

Пример 1: Браузер CUIС получает сообщения на браузере (тема VoiceIAQStats)

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

Пример 2: Браузер CUIС получает сообщения на браузере (тема VoiceCSQDetailsStats)

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

Пример 2: Браузер CUIС получает сообщения на браузере (тема ResourceIAQStats)

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```

Изящество получает то же обновление на агенте/Supervisor Desktop. (Нажмите F12 на браузере для наблюдения обновлений),

Пример 1:

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= Subscribing for rooms:  
<VoiceIAQStats=cssCsq> report-layer.js:1414:11
```

```
1/18/2017, 10:14:07 PM=INFOC8E2DB0C10000140000000A40A4E5E6B= ReportManager : filterString  
[{"fieldId":"720298FB10000140000000A10A4E5E6F", "fieldType":"VALUELIST", "name":"VoiceIAQStats.esd  
Name", "operator":"SetValues", "value":[{"key":"cssCsq", "desc":"cssCsq"}], "valuelistId":"01FB2C011  
0000133771FC3C33F57F543", "isKeyField":true}] report-layer.js:1414:11
```