

# La comprensión y resolver problemas de los datos vivos señala sobre UCCX

## Contenido

[Introducción](#)

[Cambios en los datos vivos con SocketIO](#)

[Descripción técnica general](#)

[WebSocket contra el mecanismo de sondeo largo](#)

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

[Interrogación larga](#)

[Número de control de clientes que ejecutan la interrogación larga](#)

[Flujo de datos vivo](#)

[Espacios para servidores y temas de SocketIO](#)

[Productor](#)

[Repartidor](#)

[Clientes](#)

[Autenticación](#)

[Seguimiento vivo de los datos](#)

[Resolver problemas los datos vivos](#)

[Healthcheck básico](#)

[Resumen](#)

[Verifique la versión del buscador](#)

[Verifique el CPU y el uso de la memoria](#)

[Verifique la fuente viva del flujo de datos está en línea](#)

[Verifique el NTP.](#)

[Verifique el DNS](#)

[Funcione con una prueba de diagnóstico](#)

[Verifique el nombre de host](#)

[Verifique la configuración VM](#)

[Verifique los Certificados UCCX](#)

[Verifique la configuración de clasificación soportada](#)

[Advertencias conocidas](#)

[Seguimiento vivo del registro de los datos](#)

[Registros del motor](#)

[Registros de SocketIO \(1\)](#)

[Registros de SocketIO \(2\)](#)

[Cliente Logs\(1\)](#)

[Registros de SocketIO](#)

[Registros del cliente \(2\)](#)

## Introducción

Este documento describe la información viva de los datos en la solución expresa unificada del Centro de contacto (UCCX) a partir de la versión 11.x y delinea los puntos de verificación del troubleshooting junto con soportar el análisis del registro. El artículo también delinea las diversas advertencias conocidas con sus workarounds.

## Cambios en los datos vivos con SocketIO

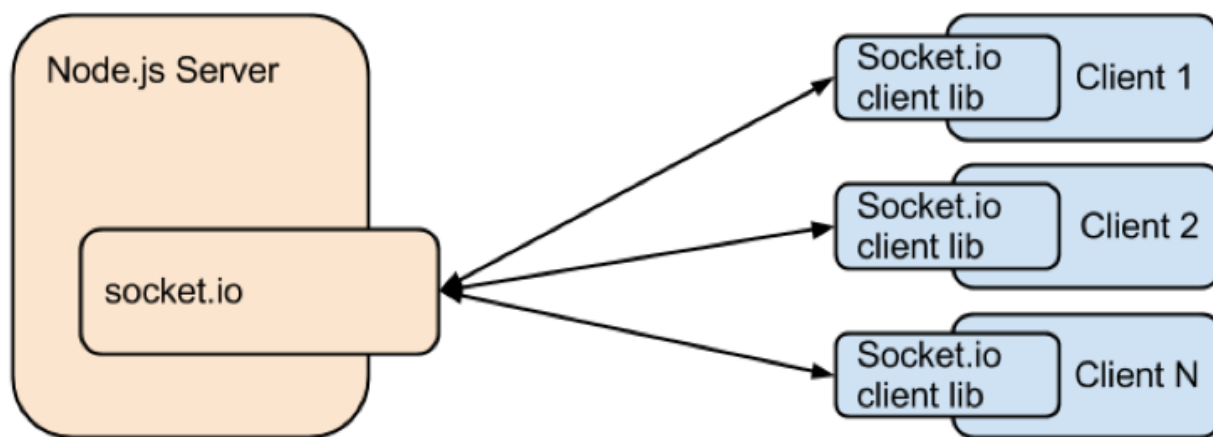
SocketIO basó el diseño de datos vivo substituye Openfire servidor XMPP como la infraestructura predeterminada del mensaje de la información de **UCCX 11.x** hacia adelante

Es una solución scalable y eficiente evento-conducida que se basa en el impementation Node.js, con los gastos indirectos reducidos de la comunicación usando el Request For Comments 6455 (RFC 6455) del protocolo de WebSocket para enviar los mensajes en tiempo real al navegador (clientes).

Anterior en UCCX 10.x, había un solo **servicio de notificación de Cisco CCX, sobre la base de OpenFire** la implementación XMPP) que fue utilizada para ambos delicadeza, charla, las notificaciones por correo electrónico así como los informes vivos de los datos sobre los gadgets del agente de la delicadeza y CUIC.

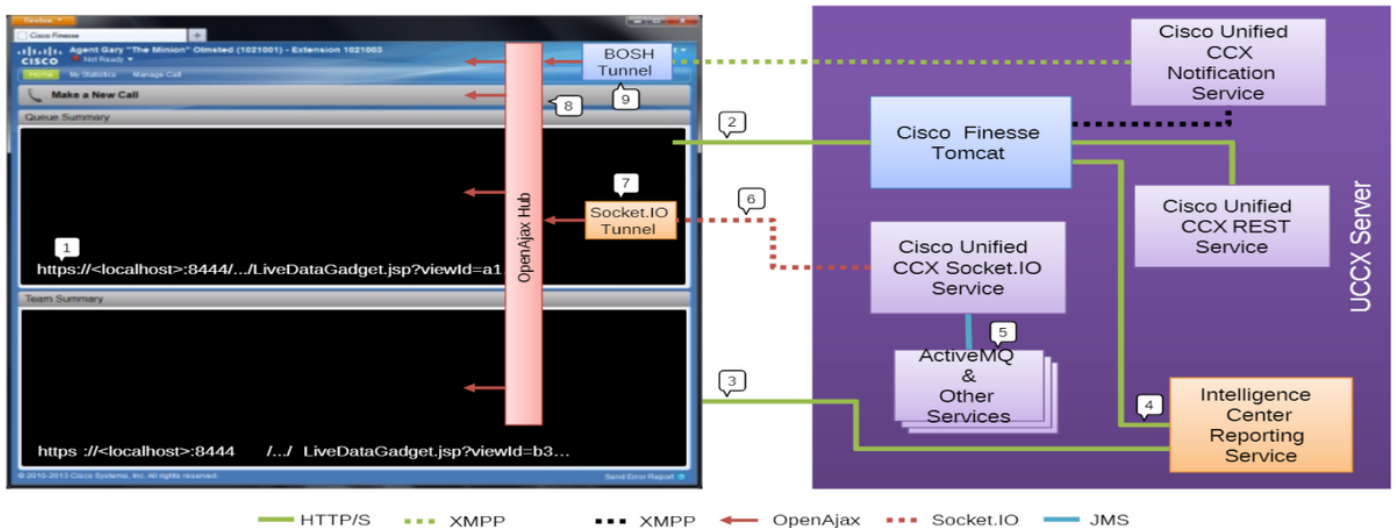
De UCCX 11.x, con la introducción de SocketIO para los informes Live - la charla y el correo electrónico del cliente de la delicadeza continúan utilizando el **servicio de notificación de Cisco CCX**, es decir Openfire como el servidor XMPP, pero el módulo vivo de la información de los datos utiliza el nuevo **servicio del servicio de CiscoUnified CCX Socket.IO**.

Esto reduce la información viva de los datos por encima, y hace SocketIO solamente responsable de elaborar y de enviar las actualizaciones vivas de los datos al buscador del cliente, tal y como se muestra en de la imagen del servidor de SocketIO.



## Descripción técnica general

Esta imagen está de SocketIO basó la Descripción general de la arquitectura viva de los datos.



De una perspectiva de alto nivel, los datos para los informes de los datos Live se envían en la notación del objeto del Javascript (JSON) del módulo del administrador de las informaciones en tiempo real del motor UCCX (RTDM), vía el bus del servicio de mensajería de las Javas (JM) (uso Apache ActiveMQ de implementar esto, es decir cola activa de Message), al servidor Socket.IO recibido en UCCX (expuesto vía el servicio unificado de CCX Socket.IO) que entonces extrae esta información, lo procesan (consume) en los cuartos y después distribuyen (los envíos) esto a los clientes que usan las conexiones HTTPS WebSocket sobre el TCP.

ActiveMQ es un middleware que implementa la aplicación JM que programa Interface(API) y se utiliza para enviar los mensajes entre dos componentes, UCCX vive servidor de los datos (motor UCCX) y de SocketIO.

## WebSocket contra el mecanismo de sondeo largo

Hay dos maneras que los navegadores pueden inscribir para recibir las actualizaciones vivas de los datos, usando la tecnología WebSocket(WS)/WebSocket Secure(WSS), o usando el mecanismo de sondeo largo.

### WebSocket (RFC 6455)

- El protocolo de WebSocket (RFC 6455) es un protocolo TCP basado independiente. Su solamente relación al HTTP es que su apretón de manos es interpretado por los servidores HTTP pues una petición de la actualización, y utilizado para establecer una conexión persistente con el servidor para las actualizaciones continuas.
- Después de un apretón de manos acertado de WebSocket, los clientes y servidores transfieren los datos hacia adelante y hacia atrás en unidades conceptuales más pequeñas (objetos JSON) designadas en esta especificación los **mensajes**. El mensaje de WebSocket no corresponde necesariamente a una capa de red determinada que enmarca, pues un mensaje hecho fragmentos se puede unir o partir por un intermediario.
- Este proceso comienza con el cliente que envía un pedido de HTTP regular al servidor.
- Una encabezado de la actualización se incluye en esta petición que informe al servidor que el cliente desea establecer una conexión de WebSocket. es decir **actualización de los**

## protocolos de la transferencia HTTP/1.1101: conexión del websocket: Actualización.

- Después de ese, se abre un cuarto y los mensajes se envían continuamente al cliente/al navegador.
- WebSocket se conoce para reducir el procesar de los gastos indirectos en el servidor de publicación, y proporciona las actualizaciones vivas seguras y dinámicas.

### Interrogación larga

- Es una variación de la técnica tradicional de la interrogación y permite la emulación de un empuje de la información de un servidor a un cliente.
- Con la interrogación larga, la información de los pedidos de cliente del servidor de una manera similar a una encuesta normal. Una conexión se celebra abierta entre el cliente de Web y el servidor Web de modo que cuando el servidor tiene nueva información pueda avanzarla al cliente y la conexión entonces esté cerrada.
- Si el servidor no tiene ninguna información disponible para el cliente, en vez del envío una respuesta vacía, el servidor lleva a cabo la petición y espera una cierta información para estar disponible.
- Una vez que la información está disponible (o después de un descanso conveniente), una respuesta completa se envía al cliente.
- Las re-peticiones del cliente normalmente entonces inmediatamente la información del servidor, de modo que el servidor tenga siempre una petición que espera disponible que pueda utilizar para entregar los datos en respuesta a un evento.
- La interrogación larga causa los gastos indirectos adicionales en el servidor que petición esta información.

### Número de control de clientes que ejecutan la interrogación larga

- Para UCCX viven los datos, **WebSocket es favorable y recomendado** sobre el mecanismo tradicional de la largo-interrogación.
- Para marcar si los clientes están utilizando WebSocket O el mecanismo de sondeo largo para los datos vivos, usted puede analizar los debugs de SocketIO y ver los mensajes.

### Pasos:

- Dé vuelta encima de los debugs IO del socket para nivelar el DEBUG (del valor por defecto ADVIERTA) - navegan a la **utilidad > a la traza UCCX**.
- Ate los últimos registros del servidor de SocketIO:

Verá esto:

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): ...

## Flujo de datos vivo

El flujo de datos vivo se puede resumir como: **CCX gadgets de los clientes del → del servidor de SocketIO del → del → JM ActiveMQ del motor (ICD\_RTDM, SS\_RMCM) (buscador Web)**

Éstos son logging facilities esa necesidad de ser habilitado en el motor UCCX:

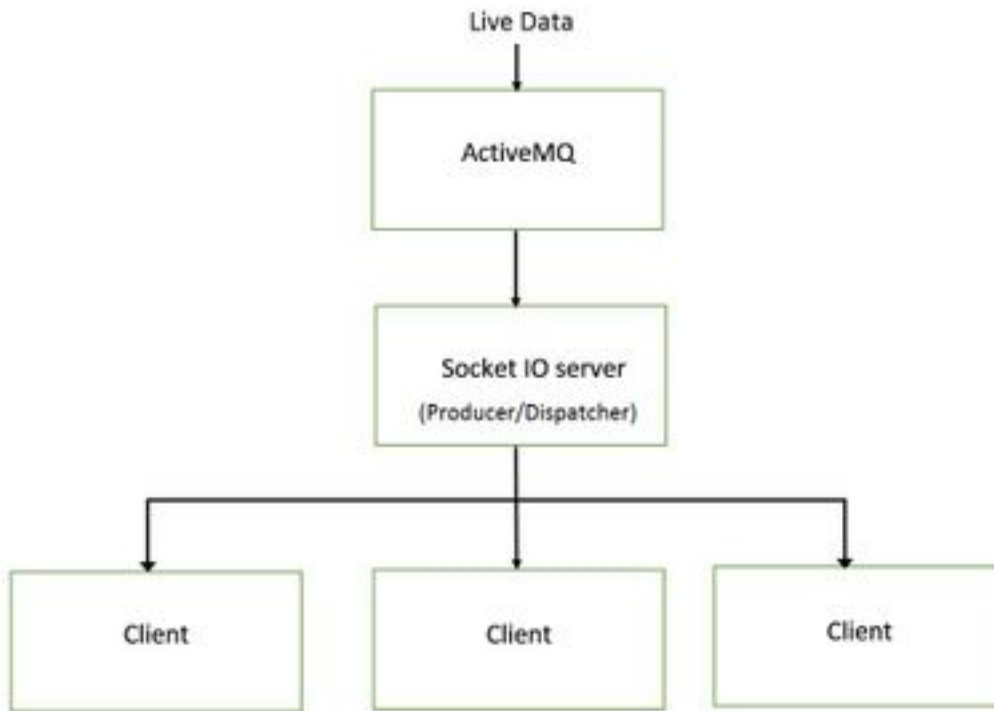
**ICD\_RTDM** = administrador de las informaciones en tiempo real de la distribución de llamada inteligente

**SS\_RMCM** = administrador de contactos del administrador de recursos del subsistema

**JM ActiveMQ** = implementación activa de la cola de la Mensajería del servicio de mensajería de las Javas

El servidor vivo de SocketIO de los datos se diseña sobre todo para recibir la información de la fuente de datos viva (tal como JM) y para publicarla, vía WebSocket, para vivir los clientes de los datos (como un gadget del cliente en la delicadeza o Cisco unificó la aplicación de Web del centro de la inteligencia (CUIC)).

Este el diagrama esquemático lógico describe el flujo de datos del motor UCCX (datos vivos) en ActiveMQ, en donde el servidor Socket.IO después extrae la información y la distribuye a los clientes, tal y como se muestra en de la imagen de SocketIO basó las suscripciones del cliente:



Los datos vivos publican los eventos en los lotes sobre los temas vía ActiveMQ.

El servidor de SocketIO inscribe a estos temas y recibe los lotes del evento.

## Espacios para servidores y temas de SocketIO

El servidor de SocketIO se configura con los diversos cuartos o los temas definidos en `/opt/cisco/uccx/socketioserver/conf/socketioservice.properties` clasifican.

Todos los informes vivos de los datos se asocian a un tema JM, que es útil mientras que analiza los registros.

La actualización del tema JM sucede en cada 3 segundos, con certeza señala que éstos son evento basado (una actualización se fija solamente cuando se acciona un evento, como un cambio del estado de agente).

Se enumeran en la tabla de informe a la asignación del tema JM:

nombre	Señale el	JM	Nombre del tema
Informe de estadísticas del agente CSQ		AgentCSQStats	---> el evento basó el tem
Informe del registro del estado de agente		AgentStateDetailStats	---> el evento basó el tem
Informe de los datos estadísticos de agentes		ResourceIAQStats	
Informe del resumen del equipo de agentes		ResourceIAQStats	
Informe del estado del equipo		ResourceIAQStats	
Informe del resumen del equipo		ResourceIAQStats	

Informe de la reunión de la Voz CSQ  
Resumen de la Voz CSQ  
Informe del resumen de la charla CSQ  
Informe de los datos estadísticos de agentes de la charla  
Informe de los datos estadísticos de agentes del correo electrónico  
Informe del resumen del correo electrónico CSQ

VoiceCSQDetailsStats  
VoicelAQStats  
ChatQueueStatistics ---> el evento basó el tem  
ChatAgentStats ---> el evento basó el tema  
EmailAgentStats ---> el evento basó el tema  
EmailQueueStatistics ---> el evento basó el tem

Los clientes como el servidor CUIIC conectan con el servidor Socket.IO para inscribir a los cuartos enumerados arriba para recibir los eventos vivos de los datos del servidor de SocketIO.

La petición de la suscripción se envía del CUIIC mientras que pronto un informe se inicia del CUIIC. Por ejemplo iniciar el **informe del resumen de la Voz CSQ** en el CUIIC acciona una petición de la suscripción al cuarto/al tema correspondientes (VoicelAQStats) al servidor SIO.

## Productor

1. Recibe eventos lotes de ActiveMQ.
2. Los rompe abajo en sus eventos discretos.
3. Ojeadas en cada evento para obtener la identificación del evento.
4. Rutea el evento al cuarto basado en el tema combinado con la identificación obtenida.

## Repartidor

El repartidor recibe el cuarto discreto nombrado los eventos del productor y envía estos eventos a todos los clientes que inscriban a ese sitio.

## Clientes

1. Los clientes basados en web (CUIIC O los gadgets de los agentes/del supervisor de la delicadeza) conectan con el servidor de SocketIO sobre el socket de la red.
2. Una vez que están conectados, estos clientes inscriben a los cuartos en el servidor IO del socket.
3. Varios clientes pueden inscribir al mismo cuarto.
4. Los clientes reciben un en un momento del evento.

Ejemplo:

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
```

El servidor de SocketIO entonces comienza a publicar los eventos al cliente con esta suscripción.

## Autenticación

El servidor Socket.IO necesita autenticar a los clientes de conexión (como el gadget de los datos de la delicadeza O el CUIC vivo) antes de enviar los datos.

1. El servidor Socket.IO propone utilizar la autenticación basada token.
2. Los gadgets vivos de los datos realizan un pedido de autenticación inicial que les dé un token del auth.
3. Esto se pasa más a fondo encendido al servidor Socket.IO para autenticar la creación de una conexión WSS (servidor de WebSocket). Para esto, Socket.IO llama el API proporcionado por UCCX el token que consiguió de los gadgets.
4. Toda la autenticación se hace usando una cadena de símbolos. La autenticación de cliente se habilita en el servicio Socket.IO vía la propiedad Client.Authentication.Rest.URL.
5. Todos conexión de los clientes habilitados deben proporcionar esta cadena como parámetro en la conexión URL.
6. Los clientes al servidor Socket.IO conectan, inscriben, desinscriben, y desconectan. La conexión inicial puede ser apenas una conexión o también incluir la suscripción.

```
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 seguro

El keystore que contiene los Certificados asociados para WSS (UCCX utiliza el valor por defecto: Certificados de la aplicación de Cisco Tomcat)

La ubicación del passphrase para WSS (valor por defecto: Tomcat del sistema)



# Seguimiento vivo de los datos

Para registrar, hay 3 puntas dominantes del flujo en los datos vivos UCCX usando la arquitectura de SocketIO en 11.x (**SS\_RMCM del motor MIVR\_RTDM, del motor > JM ActiveMQ > el servidor de SocketIO**)

- El motor UCCX registra (MIVR con el conjunto RTDM y RMCM a Xdebug5) el envío del mensaje a JM (cada 3 segundos como se ve en los debugs del subsistema RM en el SS\_RMCM del motor (MIVR))
- Registros de SocketIO (el registro del servicio de SocketIO se debe fijar al debugging completo donde los JM envían los mensajes al consumidor de SocketIO, que recibe todos los mensajes que son publicados por el motor).
- Registros de la consola del navegador (F12): SocketIO que procesa las actualizaciones entrantes, creando los cuartos apropiados para cada tema, y después enviando éstos a los buscadores Web del cliente vía el repartidor del mensaje. (Éstos se ven en los registros del buscador Web Console/F12).

Estas trazas tienen que ser habilitadas para considerar el detallado messaged en los registros.

1. Motor UCCX: **ICD\_RTDM, SS\_RMCM** con el nivel 5 de Xdebugging
2. Registro del servicio de SocketIO: Nivel de debug.
3. Registros de la consola del navegador (F12) para marcar las actualizaciones vivas entrantes de los datos.

SocketIO que los registros del debug dicen solamente el estatus de los temas como conectado/que lo desconectó y que inscribió/desinscrito.

El seguimiento detallado tiene que ser habilitado para considerar los eventos específicos en un tema. Los registros del productor y del repartidor de SocketIO están disponibles sólo después de habilitar el seguimiento detallado (TAC de Cisco del contacto para habilitar el seguimiento detallado).

## Resolver problemas los datos vivos

### Healthcheck básico

El primer paso hacia los problemas vivos de los datos del troubleshooting es establecer una línea de fondo de la configuración admitida y realizar una revisión médica básica del sistema.

Los problemas encontrados en los puntos de verificación de esta configuración pueden causar los problemas en la representación viva de los datos en el gadget de la delicadeza O en la página web CUIC.

Usted debe eliminar todas estas puntas antes de proceder con el troubleshooting basado registro.

### Resumen

- La versión del buscador debe ser compatible.

- El CPU y el uso de la memoria en el UCCX deben estar dentro de los límites.
- El datasource vivo de los datos debe estar en línea en la página de los Datasources CUIIC.
- El NTP y el DNS en el UCCX deben ser en servicio, con el NTP adentro sincronizan, y DNS adelante/las búsquedas inversas en la orden correcta.
- Los HUEVOS VM deben ser correctamente aprovisionado con los requisitos del vRAM y del vCPU según las guías de consulta de los HUEVOS de Cisco.
- El NIC funcionando para el VM debe ser VMXNET3.

### **Verifique la versión del buscador**

Verifique el comportamiento de los datos Live a través de los diversos navegadores y versiones del buscador.

UCCX soporta actualmente las versiones del buscador siguientes que comienzan 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)

Recuerde INHABILITAR al modo de compatibilidad en el IE puesto que eso no se soporta para los datos Live. El IE recurre al mecanismo de sondeo largo cuando en largo sondear tan esto es una consideración importante.

### **Verifique el CPU y el uso de la memoria**

Marque el USO de la CPU y el uso de la memoria en el servidor usando RTMT o los comandos CLI.

E.g: RTMT >

**CLI: muestre la CPU de la carga de proceso**

**CLI: muestre la memoria de la carga de proceso**

Marque para saber si hay estos procesos si ve los datos vivos sobre CUIIC: UCCX\_Engine, socketioservice, cuicreporting.

### **Verifique la fuente viva del flujo de datos está en línea**

Para verificar que la fuente viva del flujo de datos esté en línea, navegue a la **página web CUIIC > al sistema > a los Datasources**

### **Verifique el NTP.**

Verifique que el NTP sea accesible y UCCX está sincronizado al servidor NTP en el estrato <=5

## CLI: estatus NTP del 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
```

sincronizado al servidor NTP (xx.xx.xx.xx) en el estrato 2

## Verifique el DNS

Verifique que el DNS esté resolviendo.

### CLI: <hostname> del Host de red del utils

Resolución local:

resoluciones hostname.domainname localmente a xx.xx.xx.xx

Resolución externa:

hostname.domainname tiene direccionamiento xx.xx.xx.xx

## Funcione con una prueba de diagnóstico

Usted puede funcionar con una prueba de diagnósticos del CLI para identificar cualquier otro problema potencial con el servidor (E.g tiempo de espera de Tomcat, problemas de red, los etc)

### CLI: el utils diagnostica la prueba

## Verifique el nombre de host

El nombre de host UCCX debe yo en minúsculas.

Refiérase por favor instalan/guía de actualización y este defecto.

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

## Verifique la configuración VM

Verifique eso que la plantilla abierta correcta del dispositivo de la virtualización (HUEVOS) se utiliza según estos links:

[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://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://software.cisco.com/download/type.html?mdfid=286287033&flowid=76362>

La configuración del adaptador de red UCCX VM (NIC) debe ser **VMXNET3**.

Si el adaptador correcto entonces no se selecciona UCCX no publica las informaciones en tiempo real en constante valora.

Si no es la configuración del adaptador de red VM VMXNET3 después siguen este link para corregir las configuraciones del adaptador:

[http://docwiki.cisco.com/wiki/Virtualization\\_for\\_Cisco\\_Unified\\_Contact\\_Center\\_Express](http://docwiki.cisco.com/wiki/Virtualization_for_Cisco_Unified_Contact_Center_Express)

## Verifique los Certificados UCCX

Certifique la discordancia o cuando los Certificados no se validan puede hacer la fuente viva del flujo de datos ir disconnected/off-liné.

- Valide el certificado cuando está indicado.
- Asegúrese de que no estén expirados los Certificados del otro vendedor.
- Verifique que el nombre de host en el certificado esté correcto. Si no, siga el procedimiento abajo.

En CCX el lado:

1. Edite este archivo `/usr/local/platform/conf/platformConfig.xml` y corrija el nombre de host y asegúrese que las entradas DNS están según el nombre en este archivo.
2. Borre y regenere los Certificados después de verificar los detalles antedichos en el cluster UCCX usando los comandos CLI:  
**CLI: fije el tomcat CERT regen**
3. Recomiencie servicio Tomcat de Cisco encendido el UCCX.  
**CLI: utils service restart Cisco Tomcat**

Más en los Certificados:

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

## Verifique la configuración de clasificación soportada

Asegúrese de que los límites de la configuración máxima estén adheridos en a la configuración. Para esto, refiera la guía de diseño del nivel de la solución UCCX para la Configuración del servidor y los límites máximos:

### UCCX

**11.5:** [http://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/cust\\_contact/contact\\_center/crs/express\\_11\\_5/desi..](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..](http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cust_contact/contact_center/crs/express_11_0/desi..)

Para confirmar esto usted puede funcionar con estos comandos CLI, verificando la directriz del apresto, sobre la configuración para conseguir la cantidad exacta.

### INFORMACIÓN REQUERIDA

Total no de los supervisores

### CONSULTA

funcione con la cuenta selecta del db\_cra sql del uccx (\*) de (resource)

asignados a los equipos	distinto selecto del supervisor donde active = "t ")
Sume no de los supervisores configurados	funcione con la cuenta SELECTA del db_cra sql del uccx (*) del recurso donde active='t y resourcetype=2
Sume no de los agentes configurados	funcione con la cuenta SELECTA del db_cra sql del uccx (*) del recurso donde active='t y resourcetype=1
Sume no de los equipos configurados	funcione con la cuenta SELECTA del db_cra sql del uccx (*) del equipo donde active='t
No de los agentes por el equipo	ejecute el db_cra team.teamname SELECTO sql del uccx, count(resource.resourceid) COMO NumOfAgents del recurso del UNID INTERNO del equipo EN team.teamid=resource.assignedTeamID donde GRUPO resource.active='t y team.active='t POR el teamname
No de los supervisores por el equipo	ejecute el db_cra team.teamName SELECTO sql del uccx, count(Supervisor.recordID) COMO NumOfSupervisors del equipo del U INTERNO del supervisor EN Supervisor.managedTeamID=team.teamid donde GRUPO Supervisor.active='t y team.active='t POR el teamName
Número de usuarios por el papel (CUIC)	ejecute sql g.name selecto como nombre de grupo, cuenta (*) como groupsize del cuic_data: cuicgroup g, cuic_data: gm del cuicgroupmemb donde grupo g.id = gm.groupid por g.name
Número de CSQ configurados	funcione con la cuenta selecta del db_cra sql del uccx (*) del contactservicequeue donde active='t

## Advertencias conocidas

[CSCva13838](#) : Retardo vivo de los datos en CUIC

[CSCus17605](#) : Retardo debido a los permisos excesivos de la información a los supervisores (mucha carga)

[CSCux33949](#) : Problemas vivos de los datos debido a las licencias aumentadas.

[CSCvb67761](#) : Informe del registro del estado de agente que causa CPU elevada los datos vivos de afectación.

[CSCva95411](#) : Servicio de SocketIO CPU elevada debido a los objetos namespace que toman la memoria excesiva del montón.

[CSCvb75279](#) : La fuente viva del flujo de datos que va off-liné durante a de sincroniza la contraseña MIVR Keystore

[CSCvc45189](#): Fuente viva del flujo de datos off-liné debido a la operación de reserva y del Restore

[CSCut04158](#): Problemas vivos de los datos debido al adaptador NIC incorrecto

Nota: Para confirmar o la solución alternativa las advertencias antedichas, implica por favor el TAC de Cisco abriendo una solicitud de servicio.

## Los datos vivos registran el seguimiento

## Registros del motor

- Los datos vivos publican los eventos en los lotes sobre los temas vía MQ activo.
- El servidor de SocketIO inscribe a estos temas y recibe los lotes del evento.

**Registros MIVR:** Los datos vivos se envían del motor a través de un mensaje JM sobre los temas (Eg.: VoiceCSQDetailsStats, VoiceIAQStats, ResourceIAQStats etc...)

Para el motor basado evento de los temas (ChatAgentStats, ChatQueueStatistics AgentCSQStats, AgentStateDetailStats) envía los mensajes vacíos JM después de cada 3 segundos para mantener el cuarto de SocketIO vivo.

Estas trazas areseen sólo después de habilitar estas trazas: [SocketIO detalló el seguimiento](#)

```
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
```

## Registros de SocketIO (1)

### Lotes de SocketIO recibe eventos de MQ activo

Mientras que llegan los lotes de eventos, el servidor de SocketIO hace esto:

- Los rompe abajo en sus eventos discretos.
- Ojeadas en cada evento para obtener la identificación del evento.
- Rutea el evento al cuarto basado en el tema combinado con la identificación obtenida

Los eventos discretos con los nombres del sitio se envían a la cola en el repartidor

Nota: El productor de SocketIO que tiene el módulo de escucha/el consumidor JM consigue los datos del bus JM en los temas vía ActiveMQ y los envía a los cuartos de SocketIO que se envían a la cola al repartidor.

Estas trazas se consideran sólo después de habilitar las trazas detalladas en SocketIO (TAC de Cisco del contacto para habilitar lo mismo).

### Ejemplo 1: Tema de 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
```

## Ejemplo 2: Tema de 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
```

## Ejemplo 3: Tema de 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
```

## Registros de SocketIO (2)

Repartidor IO del socket: Consigue los eventos del productor y los entrega a los clientes

El repartidor recibe el cuarto discreto nombrado los eventos del productor y envía estos eventos a todos los clientes que inscriban a ese sitio.

Estas trazas se consideran sólo después de habilitar las trazas detalladas en SocketIO (TAC de Cisco del contacto para habilitar lo mismo).

## Ejemplo 1: Tema de 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
```

## Ejemplo 2: Tema de 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
```

## Ejemplo 3: Tema de 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
```

## Cliente Logs(1)

Demostraciones del registro del cliente que conectan, inscribiendo y recibiendo los eventos.

1. Los clientes basados en web conectan con el servidor como WebSocket o clientes largos de la interrogación.
2. Una vez que está conectado, o como parte de conectar a estos clientes inscriba a los cuartos en el servidor IO del socket.
  - Varios clientes pueden inscribir al mismo cuarto.
  - Cada sitio conseguirá/recibe solamente los eventos de ese tipo
3. Los clientes recibirán 1 en un momento del evento.

El ejecutarse los informes significa inscribir a los eventos del servidor SIO.

Iniciar el informe CUIC acciona una petición de la suscripción al tema correspondiente al informe al servidor de SocketIO.

**Example1: CUIC que envía la petición de la suscripción después de funcionar con el informe del resumen de la Voz CSQ (tema 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
```

**Example2: CUIC que envía la petición de la suscripción después de funcionar con el informe de la reunión del agente de la Voz CSQ (tema 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
```

**Example3: CUIC que envía la petición de la suscripción después del informe corriente de los datos estadísticos de agentes (tema: 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
```

El servidor de SocketIO recibió la petición de la suscripción y SE UNE Al cuarto.



## Registros de SocketIO

Example1: El servidor SIO SE UNE A después de recibir la petición de la suscripción (tema 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
```

Example2: El servidor SIO SE UNE A después de recibir la petición de la suscripción (tema 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
```

Example3: El servidor SIO SE UNE A después de recibir la petición de la suscripción (tema 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
```

## Registros del cliente (2)

Servidor de SocketIO después de recibir la petición de la suscripción, comienzo que envía los eventos a través del repartidor del mensaje. Abajo están las actualizaciones del evento que se reciben en el navegador.

Registro del navegador: La actualización se recibe en el navegador (prensa F12 en el navegador para ver la actualización)

Example1: El navegador CUIIC recibe los mensajes en el navegador (tema 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
```

Example2: El navegador CUIIC recibe los mensajes en el navegador (tema

## 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
```

## Example2: El navegador CUIC recibe los mensajes en el navegador (tema 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
```

La delicadeza recibe la misma actualización en el agente/el Supervisor de escritorio. (Prensa F12 en el navegador para ver las actualizaciones)

## Example1:

```
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
```