
Artículo

[Bernardo Linarez](#) · 1 jul, 2019 Lectura de 16 min

Dominando JDBC con SQL Gateway

Como todos sabemos, IRIS Database / Caché es un motor de base de datos que efectúa muchas tareas dentro de sí misma. Sin embargo, ¿qué puede hacer cuando necesita tener acceso a una base de datos externa? Una opción es utilizar el SQL Gateway en Caché mediante un conector JDBC. En este artículo, mi objetivo es responder las siguientes preguntas con el fin de ayudarle a que se familiarice con la tecnología y que resuelva algunos de los más problemas comunes.

Resumen general

- [¿Cuáles son los parámetros de conexión que necesita para conectarse a una base de datos remota?](#)
- [¿Cuál es la diferencia de JDBC Gateway en comparación con el servicio que ofrece Java Gateway en la capa de interoperabilidad?](#)
- [¿Cuáles son las herramientas y métodos que están disponibles para resolver problemas?](#)
- [¿Cuáles son los tipos de problemas más comunes y los enfoques para resolverlos?](#)

Antes de profundizar en estas preguntas, hablemos rápidamente de la arquitectura de la Conectividad con las bases de datos de Java (JDBC) en SQL Gateway. Para que esto sea más sencillo, puede pensar en la arquitectura como una aplicación en IRIS Database / Cache realiza una conexión TCP con alguno de los procesos de Java, dicho proceso se llama Puerta de enlace. Entonces, el proceso Puerta de enlace se conecta con una base de datos remota, como Caché, Oracle o SQL Server, mediante el controlador que se especificó para esa base de datos. Para obtener más información sobre la arquitectura de SQL Gateway, consulte los documentos de apoyo sobre [el uso de SQL Gateway en Caché](#).

Parámetros de conexión

Cuando se conecte con una base de datos remota, debe proporcionar los siguientes parámetros:

- Nombre de usuario
- Contraseña
- Nombre del controlador
- URL
- Ruta para la clase

Cómo conectarse con una base de datos de Caché

Por ejemplo, si necesita conectarse con una instancia de Caché, que utilice SQL Gateway mediante la JDBC, necesita navegar hacia [System Administration] -> [Configuration] -> [Connectivity] -> [SQL Gateway Connections] en el portal de Administración del sistema (SMP). A continuación, haga clic en "Crear una nueva conexión" y especifique "JDBC" como el tipo de conexión.

The screenshot shows the 'Type of connection: JDBC' configuration window. The fields are filled as follows:

- Connection name:** cachejdbc
- User:** _SYSTEM
- Password:** [masked with dots]
- Driver name:** com.intersys.jdbc.CacheDriver
- URL:** jdbc:Cache://127.0.0.1:1972/SAMPLES
- Class path:** [empty]
- Properties:** [empty]

Below the fields, there are several options:

- Do not use delimited identifiers by default:** ☐
- Use COALESCE:** ☒
- Use NVL() instead of IFNULL():** ☐
- Conversion in composite Row IDs:**
 - ☐ Do not convert non-character values
 - ☐ Use CAST as VARCHAR
 - ☐ Use CAST as CHAR
 - ☒ Use {fn convert ...}

At the bottom right, there are three buttons: 'Test Connection' (highlighted with a blue border), 'Save', and 'Cancel'.

Cuando se conecte a un sistema IRIS Database / Caché, el nombre del controlador siempre debe ser `com.intersys.jdbc.CacheDriver`, como se muestra en la captura de pantalla. Si se conecta con una base de datos de terceros, entonces deberá utilizar un nombre diferente para el controlador (consulte la sección [Cómo conectarse con las bases de datos de terceros](#) que se encuentra más abajo).

Cuando se conecte con las bases de datos de Caché, no es necesario que especifique una ruta para la clase, porque el archivo JAR se carga automáticamente.

El parámetro URL también variará dependiendo de la base de datos a la que se esté conectando. Para las bases de datos de Caché, debe utilizar una URL en el formulario

```
jdbc:Cache://[serveraddress]:[superserverport]/[namespace]
```

Cómo conectarse con las bases de datos de terceros

Una base de datos común de terceros es Oracle. A continuación, se muestra un ejemplo de su configuración.

Type of connection: JDBC

Connection name: connectOracle

User: VICKY

Password:

Driver name: oracle.jdbc.OracleDriver

URL: jdbc:oracle:thin:@myhost:1521:orcl

Class path: /Users/mlj/tmp/ojdbc7_12g.jar
(May be a comma separated list if multiple jar files are required.)

Properties: includeSynonyms=false;restrictGetTables=true

Do not use delimited identifiers by default: ☐

Use COALESCE: ☒

Use NVL() instead of IFNULL(): ☐

Conversion in composite Row IDs: ☐ Do not convert non-character values
☐ Use CAST as VARCHAR
☒ Use CAST as CHAR
☐ Use {fn convert ...}

Test Connection Save Close

Connection successful.

Como puede ver, el nombre del controlador y la URL tienen patrones diferentes a los que utilizamos para la conexión anterior. Además, especifiqué una ruta para la clase en este ejemplo, porque necesito utilizar el controlador de Oracle para conectarme a su base de datos.

Como puede imaginarse, SQL Server utiliza diferentes URL y patrones para el nombre del controlador.

Type of connection: JDBC

Connection name: connectSQLserver

User: VICKY

Password:

Driver name: com.microsoft.sqlserver.jdbc.SQLServerDriver

URL: jdbc:sqlserver://sqlsrv/sqlsrv

Class path: /Applications/Cache20151/sqljdbc/enu/sqljdbc4.jar
(May be a comma separated list if multiple jar files are required.)

Properties:

Do not use delimited identifiers by default: ☐

Use COALESCE: ☒

Use NVL() instead of IFNULL(): ☐

Conversion in composite Row IDs: ☐ Do not convert non-character values
☐ Use CAST as VARCHAR
☐ Use CAST as CHAR
☒ Use {fn convert ...}

Test Connection Save Cancel

Puede probar si los valores son válidos al hacer clic en el botón "Test Connection". Para establecer la conexión, haga clic en "Save".

JDBC Gateway contra el Servicio empresarial de Java Gateway

En primer lugar, los servicios de JDBC Gateway y Java Gateway son completamente independientes uno del otro. JDBC Gateway puede utilizarse en todos los sistemas que se basan en Caché, mientras que el Servicio empresarial de Java Gateway únicamente existe como parte de la IRIS Interoperabilidad / Ensemble. Además, el servicio de Java Gateway utiliza un proceso diferente en comparación al que utiliza JDBC Gateway. Para obtener más información sobre el Servicio empresarial de Java Gateway, consulte [El Servicio empresarial de Java Gateway](#).

Herramientas y métodos

A continuación, se muestran 5 herramientas y métodos comunes que se utilizan para resolver problemas con JDBC SQL Gateway. Mi intención primero es discutir sobre estas herramientas y mostrarle algunos ejemplos de cuándo se utilizan en [la siguiente sección](#).

1. Registros

A. El registro del controlador contra el registro de la Puerta de enlace

Cuando se utiliza JDBC Gateway, el registro correspondiente es el registro JDBC de SQL Gateway. Como comentamos anteriormente, el JDBC Gateway se utiliza cuando Caché necesita acceder a las bases de datos externas, lo cual significa que Caché es el cliente. El registro del controlador, sin embargo, corresponde al que utiliza el controlador JDBC de InterSystems para acceder a una base de datos de Caché desde una aplicación externa, lo que significa que Caché es el servidor. Si tiene una conexión desde una base de datos de Caché con otra base de datos de Caché, ambos tipos de registro pueden ser útiles.

En nuestros [documentos de apoyo](#), la sección sobre la habilitación del registro del controlador se denomina "Habilitación del registro para la JDBC", y la sección sobre la habilitación del registro para la puerta de enlace se denomina "Habilitación del registro para la JDBC con SQL Gateway".

Aunque en ambos registros se incluye la palabra "JDBC", estos son completamente independientes. El objetivo de este artículo es conocer más sobre JDBC Gateway, de modo que analizaré con más detalle el registro para la puerta de enlace. Para obtener más información sobre el registro del controlador, consulte la [Habilitación del registro del controlador](#).

B. Habilitación del registro para la Puerta de enlace

Si utiliza el SQL Gateway en Caché mediante la JDBC, entonces debe hacer lo siguiente para habilitar el registro: en el Portal de administración, vaya a [System Administration] > [Configuration] > [Connectivity] > [JDBC Gateway Settings]. Especifique un valor para el registro de JDBC Gateway. Esta debe ser la ruta completa y el nombre de un archivo de registro (por ejemplo, /tmp/jdbcGateway.log). Sino existe, el archivo se creará automáticamente pero el directorio no existirá. Caché iniciará con el registro de la JDBC con SQL Gateway por usted.

Si utiliza el Servicio empresarial de Java Gateway en Ensemble, consulte [Habilitación del registro para Java Gateway](#) para obtener más información sobre cómo habilitar el registro.

C. Análisis del registro para la Puerta de enlace

Ahora que recopiló un registro para la puerta de enlace, quizás se pregunte: ¿cuál es la estructura del registro y cómo puedo leerlo? ¡Excelentes preguntas! Aquí le proporcionaré la información básica para que pueda comenzar. Desafortunadamente, no siempre es posible interpretar completamente el registro sin acceder al código fuente, así que para este tipo de situaciones complejas, ¡no dude en ponerse en contacto con el Centro Mundial de Respuesta de InterSystems (WRC)!

Para desmitificar la estructura del registro, recuerde que siempre son segmentos grandes de datos seguidos por una descripción de lo que hacen. Por ejemplo, observe esta imagen en la que se resalta a algunos elementos básicos de sintaxis:

```

>> MESSAGE:

msgId: 29

Sent: (13:42:08:706) [Job number = -1] [ThreadID = 12]
0000: 02 00 00 00 00 00 00 00 00 00 00 00 59 4D .....YM
000E: 02 04 .....

<< MESSAGE

Received: (13:42:08:706) [Job number = -1] [ThreadID = 12]
0000: 71 00 00 00 00 00 00 00 00 00 00 00 59 4D q.....YM
000E: 03 04 21 02 04 6C 01 73 65 6C 65 63 74 20 ...!..l.select
001C: 54 31 2E 22 54 49 4D 45 53 54 41 4D 50 54 T1."TIMESTAMP
002A: 45 53 54 22 20 41 53 20 43 31 2C 54 31 2E EST" AS C1,T1.
0038: 22 4E 55 4D 31 22 20 41 53 20 43 32 2C 54 "NUM1" AS C2,T
0046: 31 2E 22 56 41 52 43 48 41 52 31 22 20 41 1."VARCHAR1" A
0054: 53 20 43 33 20 46 52 4F 4D 20 20 22 51 55 S C3 FROM "QU
0062: 41 4C 49 54 59 22 2E 22 56 49 43 4B 59 22 ALITY"."VICKY"
0070: 20 54 31 20 4F 52 44 45 52 20 42 59 20 43 T1 ORDER BY C
007E: 32 2

>> MESSAGE:

msgId: 33

Sent: (13:42:08:783) [Job number = -1] [ThreadID = 12]
0000: 02 00 00 00 00 00 00 00 00 00 00 00 59 4D .....YM
000E: 02 04 .....

<< MESSAGE

Received: (13:42:08:783) [Job number = -1] [ThreadID = 12]
0000: 05 00 00 00 00 00 00 00 00 00 00 00 59 4D .....YM
000E: 03 04 0D 02 04 .....

>> MESSAGE:

msgId: 13

```

Para entender lo que el término Received significa aquí, es necesario recordar que en el registro de la puerta de enlace se documentan las interacciones entre la puerta de enlace y el flujo descendente de la base de datos. Por lo tanto, el término Received significa que la información proveniente de IRIS Database/Caché/Ensemble se recibió en la puerta de enlace. En el ejemplo anterior, la puerta de enlace recibió el texto de una consulta para SELECT. El significado de los diferentes valores de msgId se encuentran en el código interno. El 33 que vemos aquí significa "Preparar una sentencia".

El registro por sí mismo también proporciona información sobre el controlador, la cual se puede verificar cuando se depuran los errores. Por ejemplo:

```
-----  
Started At: Jan 14, 2016 4:05:45 PM  
Driver Name: com.intersys.jdbc.CacheDriver  
Jar File Name: cachejdbc.jar  
Client Version: 2015.2.0.653  
Compiled At: Sun 06/21/2015 19:00:45.12  
JDBC Version: 4.2  
-----
```

Como podemos ver, el Driver Name es `com.intersys.jdbc.CacheDriver`, que es el nombre del controlador que se utilizó para conectarse con el proceso para la puerta de enlace. El Jar File Name es `cachejdbc.jar`, que es el nombre del archivo jar que se encuentra en `<cacheinstalldirectory>cacheinstalldirectory>/lib/`.

2. Cómo encontrar el proceso para la puerta de enlace

Para encontrar el proceso para la puerta de enlace, puede ejecutar el comando `ps`. Por ejemplo:

```
ps -ef | grep java
```

Con el comando `ps` se despliega la información sobre el proceso de Java, que incluye el número de puerto, el archivo jar, el archivo de registro, el ID del proceso Java y el comando que comenzó el proceso de Java.

Este es un ejemplo del resultado del comando:

```
mlimbpr15:-mli$ ps -ef | grep java
```

```
17182 45402 26852 0 12:12PM ?? 0:00.00 sh -c java -Xrs -classpath  
/Applications/Cache20151/lib/cachegateway.jar:/Applications/Cache20151/lib/cachejdbc.jar  
com.intersys.gateway.JavaGateway 62972 /Applications/Cache20151/mgr/JDBC.log 2>&1
```

```
17182 45403 45402 0 12:12PM ?? 0:00.22 /usr/bin/java -Xrs -classpath  
/Applications/Cache20151/lib/cachegateway.jar:/Applications/Cache20151/lib/cachejdbc.jar  
com.intersys.gateway.JavaGateway 62972 /Applications/Cache20151/mgr/JDBC.log
```

```
502 45412 45365 0 12:12PM ttys000 0:00.00 grep java
```

En Windows, puede verificar el administrador de tareas para encontrar más información sobre el proceso para la puerta de enlace.

3. Cómo iniciar y detener la puerta de enlace

Existen dos formas para iniciar y detener la puerta de enlace:

1. [Mediante el SMP](#)

2. [Utilizando el Terminal](#)

A. Mediante el SMP

Puede iniciar y detener la puerta de enlace en el SMP al ingresar en [System Administration] -> [Configuration] -> [Connectivity] -> [JDBC Gateway Server].

The screenshot shows the JDBC Gateway Server configuration page. The left sidebar contains settings for the gateway, including Port Number (62972), Log File, Java Home Directory, JVM Arguments, and JAVA_HOME Environment Variable. The main content area shows Status Information, including a Refresh button, a Refresh interval (30 seconds), and a status indicator that the gateway is running. Below the status information is a table of Recent Activity, which shows a list of events with columns for ID, Date Time, Record Type, Job, and Text. The 'Start' button in the Status Information section is highlighted with a red box.

B. Utilizando el Terminal

En equipos que cuentan con el sistema operativo Unix, también puede iniciar la puerta de enlace desde el terminal. Como comentamos en [la sección anterior](#), el resultado de `ps -ef | grep java` contiene el comando que inició el proceso para Java, que en el ejemplo anterior es:

```
java -Xrs -classpath /Applications/Cache20151/lib/cachegateway.jar:/Applications/Cache20151/lib/cachejdbc.jar
com.intersys.gateway.JavaGateway 62972 /Applications/Cache20151/mgr/JDBC.log
```

Para detener la puerta de enlace desde el terminal, puede finalizar el proceso. El ID del proceso en Java es el segundo número de la línea que contiene el comando anterior, que en el ejemplo anterior es 45402. Por lo tanto, para detener la puerta de enlace, puede ejecutar:

```
kill 45402
```

4. Cómo escribir un programa en Java

Para ejecutar un programa en Java debe conectarse con una base de datos de flujo descendente, ya que es una excelente manera de probar la conexión, verificar las consultas y aislar la causa de un problema determinado. Adjunté un ejemplo de un programa de Java, el cual establece una conexión con SQL Server e imprime una lista de todas las tablas. En [la siguiente sección](#) explicaré por qué esto puede ser útil.

```
import java.sql.*;

import java.sql.Date;

import java.util.*;

import java.lang.reflect.Method;

import java.io.InputStream;

import java.io.ByteArrayInputStream;

import java.math.BigDecimal;

import javax.sql.*;
```

// Autor: Vicky Li

// Este programa establece una conexión con SQL Server y recupera todas las tablas. El resultado es una lista de tablas.

```
public class TestConnection {

    public static void main(String[] args) {

        try {

            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

            //reemplace la URL, el nombre de usuario y la contraseña con los parámetros correctos

            Connection conn = DriverManager.getConnection(url,username,password);

            System.out.println("connected");

            DatabaseMetaData meta = conn.getMetaData();

            ResultSet res = meta.getTables(null, null, null, new String[] {"TABLE"});

            System.out.println("List of tables: ");

            while (res.next()) {

                System.out.println(

                    " " + res.getString("TABLE_CAT") +

                    ", " + res.getString("TABLE_SCHEM") +

                    ", " + res.getString("TABLE_NAME") +

                    ", " + res.getString("TABLE_TYPE")

                );

            } catch (Exception e) {

                e.printStackTrace();

            }

        }

    }

}
```

Para ejecutar este programa de Java (o cualquier otro programa de Java), primero debe compilar el archivo .java , que en nuestro caso se llama TestConnection.java. A continuación, se generará un archivo nuevo en la misma ubicación, el cual podrá ejecutar con el siguiente comando en un sistema UNIX:

```
java -cp "<path to driver>/sqljdbc4.jar;lib/*:." TestConnection
```

En Windows, puede ejecutar el siguiente comando:

```
java -cp "<path to driver>/sqljdbc4.jar;lib/*;" TestConnection
```

5. Cómo realizar un seguimiento del jstack

Como su nombre lo indica, jstack imprime un registro de seguimiento (stack traces) de Java en los subprocesos de Java. Esta herramienta puede ser útil cuando se necesita comprender mejor lo que está haciendo alguno de los procesos de Java. Por ejemplo, si observa que el proceso para la puerta de enlace reproduce un mensaje determinado en su registro, es posible que desee recopilar un seguimiento del jstack. Quiero señalar que jstack es una herramienta de bajo nivel que solo debería utilizarse cuando otros métodos, como el análisis del registro para la puerta de enlace, no resuelven el problema.

Antes de que recopile un seguimiento del jstack, debe asegurarse de que el Java Development Kit (JDK) está instalado. Este es el comando para recopilar un seguimiento del jstack:

```
jstack -F <pid> > /<path to file>/jstack.txt
```

donde pid es el ID del proceso para la puerta de enlace, el cual puede obtenerse cuando se ejecuta el comando ps, por ejemplo, en ps -ef | grep java. Para obtener más información sobre cómo encontrar el pid, consulte de nuevo [Cómo iniciar y detener la puerta de enlace](#).

Ahora bien, aquí puede consultar algunas consideraciones especiales para los equipos que cuentan con Red Hat. En el pasado, hubo problemas para adjuntar el jstack en el proceso de JDBC Gateway (al igual que en el proceso del Servicio empresarial de Java Gateway que comenzó en Ensemble) con algunas versiones de Red Hat, por lo que la mejor manera para recopilar un seguimiento del jstack en Red Hat es iniciar el proceso para la puerta de enlace de forma manual. Para obtener las instrucciones, consulte la sección [Cómo recopilar un seguimiento del jstack en Red Hat](#).

Los tipos de problemas más comunes y los enfoques para resolverlos

1. Problema: Java no se instaló correctamente

En esta situación, verifique la versión de Java y las variables de entorno.

Para verificar la versión de Java, puede ejecutar la siguiente línea de comando desde un terminal:

```
java -version
```

Si recibe el error java: Command not found, entonces el proceso de Cache no puede encontrar la ubicación de los ejecutables de Java. Esto normalmente se puede arreglar con la instalación de los ejecutables de Java en PATH. Si tiene problemas para realizar esto, no dude en ponerse en contacto con el Centro Mundial de Respuesta de InterSystems (WRC).

2. Problema: Hubo un fallo en la conexión

Un buen método para diagnosticar los fallos en la conexión es verificar si el proceso para la puerta de enlace se inicia correctamente. Puede hacerlo, ya sea, al verificar el [registro para la puerta de enlace](#) o [el proceso para la puerta de enlace](#). En las versiones modernas, también puede ir al SMP y consultar [System Administration] -> [Configuration] -> [Connectivity] -> [JDBC Gateway Server], para comprobar si en la página se muestra el mensaje "JDBC Gateway está en ejecución".

Si el proceso para la puerta de enlace no está en ejecución, es posible que Java no se instaló correctamente o que utiliza el puerto equivocado, si el proceso para la puerta de enlace está en ejecución, entonces es probable que los parámetros de conexión no sean los correctos.

Si ocurre la primera situación, consulte [la sección anterior](#) y verifique nuevamente el número del puerto. Aquí analizaré con más detalle esta última situación.

Es responsabilidad del cliente utilizar los parámetros de conexión correctos:

- nombre de usuario

- contraseña
- nombre del controlador
- URL
- ruta de la clase

Puede verificar si tiene los parámetros correctos mediante cualquiera de las siguientes tres formas:

- Utilice el botón "Test Connection" después de seleccionar un nombre para la conexión en [System Administration] -> [Configuration] -> [Connectivity] -> [SQL Gateway Connections].
Nota: en los sistemas modernos, el procedimiento "Test Connection" proporciona mensajes de error que son útiles, en los sistemas más antiguos es necesario el [registro de JDBC Gateway](#) para encontrar más información sobre el fallo.
- Ejecute la siguiente línea de comandos desde un terminal de Caché para probar la conexión:

```
d $SYSTEM.SQLGateway.TestConnection(<connection name>)
```

- Ejecute un programa de Java para establecer una conexión. El programa que escriba puede ser similar al [ejemplo](#) que discutimos anteriormente.

3. Problema: falta de correspondencia entre la forma en que Caché interpreta al JDBC y la forma en que la base de datos remota interpreta al JDBC, como en:

- los problemas con el tipo de datos
- el procedimiento para almacenar con parámetros de salida
- las transmisiones

Para esta categoría, con frecuencia es más útil trabajar con ayuda del Centro Mundial de Respuesta de InterSystems (WRC). Este es el procedimiento que realizamos con frecuencia para determinar si el problema está dentro de nuestro código interno o en la base de datos remota (o con el controlador):

- Se revisan los registros y se analiza lo que se envió
- Se reproduce el problema que se encuentra fuera del Caché al [escribir un programa en Java](#).

Notas al pie de página

El Bussiness Service JavaGateway

El nombre de la clase del Servicio empresarial en IRIS Interoperabilidad/Ensemble es `EnsLib.JavaGateway.Service`, y la clase del adaptador es `EnsLib.JavaGateway.ServiceAdapter`. En la sesión de Ensemble primero se crea una conexión con Java Gateway Server, el cual es un proceso de Java. La arquitectura es similar a la que se encuentra en la JDBC con SQL Gateway, excepto que el proceso de Java lo administra la Actividad empresarial. Para obtener más detalles, consulte los [documentos de apoyo](#).

Cómo habilitar el registro del controlador

Para habilitar el registro del controlador, debe añadir un nombre de archivo para el registro al final de la cadena de conexión de la JDBC. Por ejemplo, si la cadena de conexión original tiene el siguiente aspecto:

```
jdbc:Cache://127.0.0.1:1972/USER
```

Para habilitar el registro, agregue un archivo (`jdbc.log`) al final de la cadena de conexión, de modo que se vea como lo siguiente:

```
jdbc:Cache://127.0.0.1:1972/USER/jdbc.log
```



El archivo de registro se guardará en el directorio de trabajo de la aplicación Java.

Habilitación del registro de Java Gateway en Ensemble

Si utiliza el Business Service (BS) de Java Gateway en Ensemble para ingresar a otra base de datos, entonces para habilitar el registro debe especificar la ruta y el nombre de un archivo de registro (por ejemplo, `/tmp/javaGateway.log`) en el campo "Log File" del servicio Java Gateway. Tenga en cuenta que la ruta ya debe existir.


EnsLib.JavaGateway.Service

| | | | | | |
|----------|-------|-----|----------|------|---------|
| Settings | Queue | Log | Messages | Jobs | Actions |
|----------|-------|-----|----------|------|---------|

  Search:

Heartbeat Failure Timeout

Heartbeat Failure Action




Heartbeat Failure Retry

JavaHome


Class Path

JVM Args

JDK Version



Log File




JavaDebug

☐

Java Debug Port

Java Debug Suspend



Throttle Delay

Recuerde, la conexión de Java Gateway que se utiliza en la producción de Ensemble es independiente de las conexiones que utilizan las tablas vinculadas u otras producciones. Por lo tanto, si utiliza Ensemble, debe recopilar el registro en el servicio de Java Gateway. El código que inicia el servicio Java Gateway utiliza el parámetro "Log File" en Ensemble, y no utiliza la configuración del Caché SQL Gateway en el SMP como se describió [anteriormente](#).

Cómo recopilar un seguimiento del jstack en Red Hat

La clave aquí es iniciar el proceso para la puerta de enlace manualmente, y el comando para iniciar la puerta de enlace se puede obtener con la ejecución de `ps -ef | grep java`. Más adelante se muestran todos los pasos que deben seguirse cuando se recopila un seguimiento del jstack en Red Hat para ejecutar el JDBC Gateway o el Servicio empresarial de Java Gateway.

1. Asegúrese de que el JDK está instalado.
2. En un terminal, ejecute el comando `ps -ef | grep java`. Obtenga las siguientes dos piezas de información a partir del resultado:
 - Copie el comando que inició la puerta de enlace. Debería verse algo como esto: `java -Xrs -classpath /Applications/Cache20151/lib/cachegateway.jar:/Applications/Cache20151/lib/cachejdbc.jar com.intersys.gateway.JavaGateway 62972 /Applications/Cache20151/mgr/JDBC2.log`
 - Obtenga el ID del proceso de Java (pid), es decir, el segundo número de la línea que contiene el comando anterior
3. Finalice el proceso con `kill <pid>`.
4. Ejecute el comando que copió del Paso 2.a. para comenzar un proceso para la puerta de enlace de forma manual.
5. Eche un vistazo al registro para la puerta de enlace (en nuestro ejemplo, se encuentra en `/Applications/Cache20151/mgr/JDBC2.log`) y asegúrese de ver una entrada parecida a `>> LOADJAVACLASS: com.intersys.jdbc.CacheDriver`. Este paso es solo para verificar que la llamada para la puerta de enlace se realizó correctamente.
6. En un terminal nuevo, ejecute el comando `ps -ef | grep java` para obtener el pid del proceso para la puerta de enlace.
7. Recopile un seguimiento del jstack: `jstack -F <pid> > /tmp/jstack.txt`

[#Operación empresarial](#) [#Servicio empresarial](#) [#Java](#) [#JDBC](#) [#SQL](#) [#Terminal](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

URL de fuente: <https://es.community.intersystems.com/post/dominando-jdbc-con-sql-gateway>
