

Article

[Guillaume Rongier](#) · Mai 21, 2022 16m de lecture

Maîtriser la passerelle SQL JDBC

Comme nous le savons tous, Caché est une excellente base de données qui accomplit de nombreuses tâches en son sein. Cependant, que faites-vous lorsque vous avez besoin d'accéder à une base de données externe ? Une façon de le faire est d'utiliser la passerelle Caché SQL Gateway via JDBC. Dans cet article, mon objectif est de répondre aux questions suivantes pour vous aider à vous familiariser avec cette technologie et à déboguer certains problèmes courants.

Plan de travail

- [Quels sont les paramètres de connexion dont vous avez besoin pour vous connecter à une base de données distante ?](#)
- [Qu'est-ce que la passerelle JDBC Gateway et le service de passerelle Java Gateway Service dans Ensemble ?](#)
- [Quels outils et méthodes sont disponibles pour déboguer les problèmes ?](#)
- [Quels sont les types de problèmes courants et les approches pour les résoudre ?](#)

Avant de se plonger dans ces questions, discutons rapidement de l'architecture de la passerelle JDBC SQL Gateway. Pour simplifier, vous pouvez considérer que l'architecture est la suivante : Cache établit une connexion TCP avec un processus Java, appelé processus de passerelle. Le processus de passerelle se connecte ensuite à une base de données distante, telle que Caché, Oracle ou SQL Server, en utilisant le pilote spécifié pour cette base de données. Pour plus d'informations sur l'architecture de la passerelle SQL Gateway, veuillez consulter la documentation sur [Utilisation de la passerelle Caché SQL Gateway](#).

Paramètres de connexion

Lorsque vous vous connectez à une base de données distante, vous devez fournir les paramètres suivants :

- nom d'utilisateur
- mot de passe
- nom du pilote
- URL
- chemin de classe

Connexion à la base de données Caché

Par exemple, si vous avez besoin de vous connecter à une instance de Caché en utilisant la passerelle SQL Gateway via JDBC, vous devez naviguer vers [System Administration] -> [Configuration] -> [Connectivity] -> [SQL Gateway Connections] dans le portail de gestion du système (SMP). Cliquez ensuite sur "Créer une nouvelle connexion" et spécifiez "JDBC" comme type de connexion.

Type of connection: JDBC

Connection name: cachejdbc

User: _SYSTEM

Password:

Driver name: com.intersys.jdbc.CacheDriver

URL: jdbc:Cache://127.0.0.1:1972/SAMPLES

Class path:

(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

Lors de la connexion à un système Caché, le nom du pilote doit toujours être `com.intersys.jdbc.CacheDriver`, comme indiqué dans la capture d'écran. Si vous vous connectez à une base de données tierce, vous devrez utiliser un nom de pilote différent (voir [Connexion à des bases de données tierces](#) ci-dessous).

Lorsque vous vous connectez aux bases de données Caché, vous n'avez pas besoin de spécifier un chemin de classe car le fichier JAR est téléchargé automatiquement.

Le paramètre URL varie également en fonction de la base de données à laquelle vous vous connectez. Pour les bases de données Caché, vous devez utiliser une URL de la forme suivante

```
jdbc:Cache://[server_address]:[superserver_port]/[namespace]
```

Connexion à des bases de données tierces

Une base de données tierce courante est Oracle. Un exemple de configuration est présenté ci-dessous.

Comme vous pouvez le constater, le nom du pilote et l'URL ont des caractéristiques différentes de celles que nous avons utilisées pour la connexion précédente. En outre, j'ai spécifié un chemin de classe dans cet exemple, car je dois utiliser le pilote d'Oracle pour me connecter à leur base de données.

Comme vous pouvez l'imaginer, SQL Server utilise différents modèles d'URL et de noms de pilotes.

Vous pouvez tester si les valeurs sont valides en cliquant sur le bouton " Testez la connexion ". Pour créer la connexion, cliquez sur "Enregistrer".

JDBC Gateway vs le service Java Gateway Business Service

Tout d'abord, la passerelle JDBC et le service de passerelle Java sont complètement indépendants l'un de l'autre. La passerelle JDBC peut être utilisée sur tous les systèmes basés sur Caché, alors que le service de passerelle Java n'existe que dans le cadre d'Ensemble. En outre, le service de passerelle Java utilise un processus différent de celui utilisé par la passerelle JDBC. Pour plus de détails sur le service commercial de passerelle Java, veuillez consulter [Le service commercial de passerelle Java](#).

Méthodes et outils

Vous trouverez ci-dessous 5 outils et méthodes couramment utilisés pour résoudre des problèmes avec la passerelle JDBC SQL Gateway. Je vais d'abord parler de ces outils et vous montrer quelques exemples de leur utilisation dans [la section suivante](#).

1. Journaux

A. Journal du pilote et journal de la passerelle

Lorsque vous utilisez la passerelle JDBC, le journal correspondant est le journal de la passerelle JDBC SQL. Comme nous l'avons vu précédemment, la passerelle JDBC est utilisée lorsque Caché doit accéder à des bases de données externes, ce qui signifie que Caché est le client. Le journal du pilote, par contre, correspond à l'utilisation du pilote JDBC d'InterSystems pour accéder à une base de données Caché à partir d'une application externe, ce qui signifie que Caché est le serveur. Si vous avez une connexion d'une base de données Caché à une autre base de données Caché, les deux types de journaux peuvent être utiles.

Dans notre [documentation](#) la section relative à l'activation du journal du pilote est intitulée "Activation de la journalisation pour JDBC", et la section relative à l'activation du journal de la passerelle est intitulée "Activation de la journalisation pour la passerelle SQL JDBC".

Même si les deux journaux comportent le mot "JDBC", ils sont totalement indépendants. L'objet de cet article est la passerelle JDBC, c'est pourquoi j'aborderai plus en détail le journal de la passerelle. Pour plus d'informations sur le journal du pilote, veuillez vous reporter à la section [Activation du journal du pilote](#).

B. Activation du journal du pilote

Si vous utilisez la passerelle Caché JDBC SQL Gateway, vous devez effectuer les opérations suivantes pour activer la journalisation : dans le portail de gestion, allez dans [System Administration] > [Configuration] > [Connectivity] > [JDBC Gateway Settings]. Indiquez une valeur pour le journal de la passerelle JDBC. Ce doit être le chemin complet et le nom d'un fichier journal (par exemple, /tmp/jdbcGateway.log). Le fichier sera automatiquement créé s'il n'existe pas, mais le répertoire ne le sera pas. Caché va démarrer la passerelle JDBC SQL Gateway avec journalisation pour vous.

Si vous utilisez le service commercial Java Gateway dans Ensemble, veuillez consulter [Activation de la journalisation de la passerelle Java Gateway dans Ensemble](#) pour savoir comment activer la journalisation.

C. Analyse du journal d'une passerelle

Maintenant que vous avez collecté un journal de passerelle, vous vous posez peut-être la question suivante : quelle est la structure du journal et comment le lire ? Bonne question ! Je vais vous fournir ici quelques informations de base pour vous aider à démarrer. Malheureusement, il n'est pas toujours possible d'interpréter complètement le journal sans avoir accès au code source. Pour les situations complexes, n'hésitez pas à contacter le WRC (Centre de réponse global d'InterSystems) !

Pour démystifier la structure du journal, rappelez-vous qu'il s'agit toujours d'un morceau de données suivi d'une description de ce qu'il fait. Par exemple, voyez cette image avec une coloration syntaxique de base :

Afin de comprendre ce que Received signifie ici, vous devez vous rappeler que le journal de la passerelle enregistre les interactions entre la passerelle et la base de données descendante. Ainsi, Received signifie que la passerelle a reçu l'information de Caché/Ensemble. Dans l'exemple ci-dessus, la passerelle a reçu le texte d'une requête SELECT. Les significations des différentes valeurs de msgld peuvent être trouvées dans le code interne. Le 33 que nous voyons ici signifie " Préparer l'instruction ".

Le journal lui-même fournit également des informations sur le pilote, ce qui est intéressant à vérifier lors du débogage des problèmes. Voici un exemple,

Comme nous pouvons le voir, le Driver Name est com.intersys.jdbc.CacheDriver, ce qui est le nom du pilote utilisé pour se connecter au processus de passerelle. Le Jar File Name est cachejdbc.jar, ce qui est le nom du fichier jar situé dans <cacheinstalldirectory>/lib/.

2. Trouver le processus de passerelle

Pour trouver le processus de passerelle, vous pouvez exécuter la commande ps. Par exemple,

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

Cette commande ps affiche des informations sur le processus Java, notamment le numéro de port, le fichier jar, le fichier journal, l'ID du processus Java et la commande qui a lancé le processus Java.

Voici un exemple du résultat de la commande :

```
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
```

Dans Windows, vous pouvez consulter le gestionnaire des tâches pour trouver des informations sur le processus de passerelle.

3. Lancement et arrêt de la passerelle

Il y a deux façons de lancer et d'arrêter la passerelle :

- [1. Par le biais du SMP](#)
- [2. Utilisation du terminal](#)

A. Par le biais du SMP

Vous pouvez lancer et arrêter la passerelle dans le SMP en accédant à [System Administration] -> [Configuration] -> [Connectivity] -> [JDBC Gateway Server].

B. Utilisation du terminal

Sur les machines Unix, vous pouvez également démarrer la passerelle depuis le terminal. Comme nous l'avons vu dans [la section précédente](#), le résultat de ps -ef | grep java contient la commande qui a démarré le processus Java, qui dans l'exemple ci-dessus est le suivant:

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

Pour arrêter la passerelle depuis le terminal, vous pouvez tuer le processus. L'ID du processus Java est le deuxième chiffre de la ligne qui contient la commande ci-dessus, dans l'exemple ci-dessus c'est 45402. Ainsi, pour arrêter la passerelle, vous pouvez exécuter :

```
kill 45402
```

4. Écrire un programme Java

Exécuter un programme Java pour se connecter à une base de données descendante est un excellent moyen de tester la connexion, de vérifier la requête et d'aider à isoler la cause d'un problème donné. Je joins un exemple de programme Java qui établit une connexion avec SQL Server et imprime une liste de tous les tableaux. J'expliquerai pourquoi cela peut être utile dans [la section suivante](#).

```
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.*;

// Auteur : Vicky Li
// Ce programme établit une connexion avec le serveur SQL et récupère tous les tableaux.
// Le résultat est une liste de tableaux.

public class TestConnection {
    public static void main(String[] args) {
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            //please replace url, username, and password with the correct parameters
            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();
        }
    }
}
```

Pour exécuter ce programme Java (ou tout autre programme Java), vous devez d'abord compiler le fichier .java, qui dans notre cas s'appelle TestConnection.java. Ensuite, un nouveau fichier sera généré au même endroit, que vous pourrez ensuite exécuter avec la commande suivante sur un système UNIX :

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

Dans Windows, vous pouvez exécuter la commande suivante :

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

5. Suivi d'une trace de jstack

Comme son nom l'indique, jstack imprime l'arborescence des appels de procédure Java. Cet outil peut devenir pratique lorsque vous avez besoin de mieux comprendre ce que fait le processus Java. Par exemple, si vous voyez le processus de la passerelle s'accrocher à un certain message dans le journal des passerelles, vous pourriez vouloir recueillir une trace jstack. Je tiens à souligner que jstack est un outil de bas niveau qui ne devrait être utilisé que lorsque d'autres méthodes, comme l'analyse du journal des passerelles, ne résolvent pas le problème.

Avant de collecter une trace jstack, vous devez vous assurer que le JDK est installé. Voici la commande pour collecter une trace jstack :

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

où le pid est l'ID du processus de la passerelle, qui peut être obtenu en exécutant la commande ps, telle que ps -ef | grep java. Pour plus d'informations sur la façon de trouver le pid, veuillez consulter [Lancement et arrêt de la passerelle](#).

Maintenant, voici quelques considérations spéciales pour les machines Red Hat. Dans le passé, il y a eu des problèmes pour attacher jstack au processus de la passerelle JDBC (ainsi qu'au processus du service métier de la passerelle Java lancé par Ensemble) sur certaines versions de Red Hat, donc la meilleure façon de collecter une trace jstack sur Red Hat est de lancer le processus de la passerelle manuellement. Pour les instructions, veuillez consulter [Collecter une trace jstack sur Red Hat](#).

Types courants de problèmes et approches pour les résoudre

1. Problème : Java n'est pas installé correctement

Dans cette situation, vérifiez la version de Java et les variables d'environnement.

Pour vérifier la version de Java, vous pouvez exécuter la commande suivante à partir d'un terminal :

```
java -version
```

Si vous obtenez l'erreur java : Command not found, cela signifie que le processus Cache ne peut pas trouver l'emplacement des exécutables Java. Cela peut généralement être résolu en plaçant les exécutables Java dans le PATH. Si vous rencontrez des problèmes, n'hésitez pas à contacter le WRC (Centre de réponse global).

2. Problème : échec de la connexion

Un bon diagnostic des échecs de connexion est la vérification du lancement du processus de la passerelle. Vous pouvez le faire en vérifiant le [journal de la passerelle](#) ou [le processus de la passerelle](#). Sur les versions modernes, vous pouvez également aller sur le SMP et visiter [System Administration] -> [Configuration] -> [Connectivity] -> [JDBC Gateway Server], et vérifier si la page affiche "JDBC Gateway is running".

Si le processus de passerelle ne s'exécute pas, il est probable que Java n'est pas installé correctement ou que vous utilisez le mauvais port ; si le processus de passerelle s'exécute, il est probable que les paramètres de connexion sont incorrects.

Dans le premier cas, veuillez vous reporter à [la section précédente](#) et vérifiez le numéro de port. Je discuterai plus en détail de la deuxième situation ici.

Il est de la responsabilité du client d'utiliser les paramètres de connexion corrects :

- nom d'utilisateur
- mot de passe
- nom du pilote
- URL
- chemin de classe

Vous pouvez vérifier si vous avez les bons paramètres de l'une des trois façons suivantes :

- Utilisez le bouton "Test Connection" après avoir sélectionné un nom de connexion dans [System Administration] -> [Configuration] -> [Connectivity] -> [SQL Gateway Connections].
Note : pour les systèmes modernes, "Test Connection" donne des messages d'erreur utiles ; pour les systèmes plus anciens, le [JDBC gateway log](#) est nécessaire pour trouver plus d'informations sur l'échec.
- Exécutez la ligne de commande suivante depuis un terminal Caché pour tester la connexion :

```
d $SYSTEM.SQLGateway.TestConnection(<connection name>)
```
- Exécutez un programme Java pour établir une connexion. Le programme que vous écrivez peut être similaire à l' [exemple](#) dont nous avons parlé précédemment.

3. Problème : décalage entre la façon dont Caché comprend JDBC et la façon dont la base de données distante comprend JDBC, par exemple :

- problèmes de type de données
- procédure stockée avec des paramètres de sortie
- flux

Pour cette catégorie, il est souvent plus utile de travailler avec le WRC (Centre de réponse global). Voici ce que nous faisons souvent pour déterminer si le problème se situe dans notre code interne ou dans la base de données distante (ou dans le pilote) :

- regarder les journaux et analyser ce qui est envoyé
- reproduire le problème en dehors de Caché en [écriture d'un programme java](#).

Remarque

Le service commercial de la passerelle Java

Le nom de la classe du Service Métier d' Ensemble est `EnsLib.JavaGateway.Service`, et la classe de l'adaptateur est `EnsLib.JavaGateway.ServiceAdapter`. La session Ensemble crée d'abord une connexion avec le serveur Java Gateway, qui est un processus Java. L'architecture est similaire à celle de la passerelle JDBC SQL, sauf que le processus Java est géré par l'opération commerciale. Pour plus de détails, veuillez consulter la [documentation](#).

Activation du journal du pilote

Pour activer le journal du pilote, vous devez ajouter un nom de fichier journal à la fin de la chaîne de connexion JDBC. Par exemple, si la chaîne de connexion originale ressemble à ceci :

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

Pour activer la journalisation, ajoutez un fichier (jdbc.log) à la fin de la chaîne de connexion, de sorte qu'elle ressemble à ceci :

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

Le fichier journal sera enregistré dans le répertoire de travail de l'application Java.

Activation de la journalisation de la passerelle Java dans Ensemble

Si vous utilisez le service métier de la passerelle Java dans Ensemble pour accéder à une autre base de données, vous devez, pour activer la journalisation, spécifier le chemin et le nom d'un fichier journal (par exemple, /tmp/javaGateway.log) dans le champ "Log File" du service de la passerelle Java. Veuillez noter que le chemin d'accès doit exister.

N'oubliez pas que la connexion de la passerelle Java utilisée par la production Ensemble est distincte des connexions utilisées par les tableaux liés ou d'autres productions. Ainsi, si vous utilisez Ensemble, vous devez collecter le journal dans le service de passerelle Java. Le code qui démarre le service de passerelle Java utilise le paramètre "Log File" dans Ensemble, et n'utilise pas le paramètre dans la passerelle Caché SQL dans le SMP comme décrit [précédemment](#).

Récupération d'une trace jstack sur Red Hat

La clé ici est de lancer le processus de la passerelle manuellement, et la commande pour lancer la passerelle peut être obtenue en exécutant `ps -ef | grep java`. Vous trouverez ci-dessous les étapes complètes à suivre pour collecter une trace jstack sur Red Hat lors de l'exécution de la passerelle JDBC ou du service métier de la passerelle Java.

1. Assurez-vous que le JDK est installé.
2. Dans un terminal, exécutez `ps -ef | grep java`. Obtenez les deux informations suivantes à partir du résultat :
 - o a. Copiez la commande qui a lancé la passerelle. Cela devrait ressembler à quelque chose comme ça :

```
java -Xrs -classpath /Applications/Cache20151/lib/cachegateway.jar:/Applications/Cache20151/lib/cachejdbc.jar com.intersys.gateway.JavaGateway 62972 /Applications/Cache20151/mgr/JDBC2.log
```
 - o b. Obtenez l'ID du processus Java (pid), qui est le deuxième chiffre de la ligne qui contient la commande ci-dessus.

3. Arrêtez le processus avec `kill <pid>`.
4. Exécutez la commande que vous avez copiée à l'étape 2.a. pour lancer manuellement un processus de passerelle.
5. Jetez un coup d'oeil au journal de la passerelle (dans notre exemple, il est situé dans `/Applications/Cache20151/mgr/JDBC2.log`) et assurez-vous que vous voyez des entrées comme `>> LOADJAVACLASS: com.intersys.jdbc.CacheDriver`. Cette étape est juste pour vérifier qu'un appel à la passerelle est effectué avec succès.
6. Dans un nouveau terminal, exécutez `ps -ef | grep java` pour obtenir le pid du processus de la passerelle.
7. Rassemblez une trace jstack : `jstack -F <pid> > /tmp/jstack.txt`

[#Java](#) [#JDBC](#) [#ODBC](#) [#Service aux entreprises](#) [#SQL](#) [#Terminal](#) [#Caché](#)

URL de la source: <https://fr.community.intersystems.com/post/ma%C3%A9triser-la-passerelle-sql-jdbc>