

Een Studio-element voor databaseoproep configureren - Tomcat JNDI voor SQL-database

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Configuratie](#)

[Problemen oplossen](#)

[Gerelateerde informatie](#)

Inleiding

In dit document wordt het CVP-element (Customer Voice Portal) van Call Studio Database, Tomcat Java Naming en Directory Interface (JNDI) voor SQL Database beschreven.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Cisco Unified Contact Center Enterprise (UCCE), versie 12.6.2
- Cisco Package Contact Center Enterprise (PCCE), versie 12.6.2
- CVP-release 12.6.2
- CVP Call Studio 12.6.2

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende softwareversies:

- Cisco PCCE versie 12.6.2
- CVP-release 12.6.2
- CVP Call Studio 12.6.2

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

Het database-element biedt de mogelijkheid om een opdracht Structured Query Language (SQL) uit te voeren op externe databases binnen een aanroepstroom voor spraaktoepassingen. Het element vereist dat JNDI wordt geconfigureerd in de Java-toepassingsserver om databaseverbindingen te kunnen verwerken. Per element kan slechts één SQL-instructie worden uitgevoerd. Er zijn vier soorten opdrachten die kunnen worden gemaakt, maar in dit document wordt alleen de enkele opdracht gebruikt:

Single – Dit wordt gebruikt om een SQL-query uit te voeren die slechts één rij retourneert. Elementgegevens worden gemaakt met de namen van de variabelen als de namen van de geretourneerde kolommen en de waarde van die kolom als de waarde van de elementgegevens (als een tekenreeks). Als er geen rij wordt geretourneerd, worden er geen elementgegevens ingesteld.

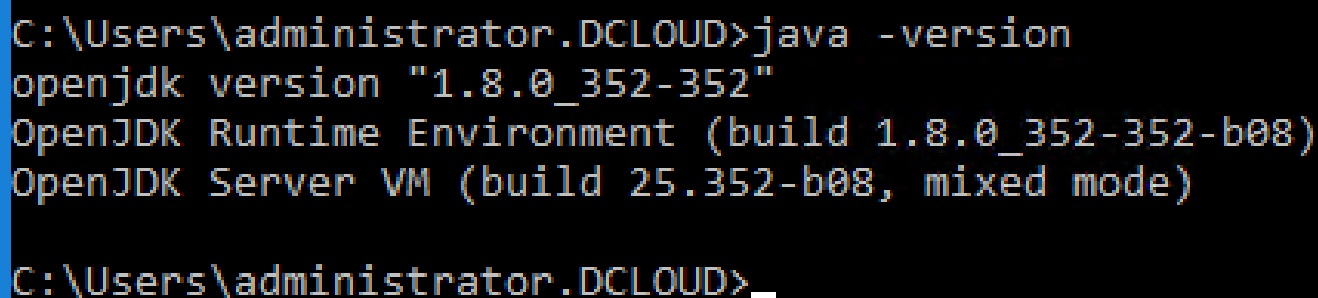
Meer informatie is te vinden in de [CVP VXML Element](#) handleiding.

Configuratie

In dit gedeelte wordt uitgelegd hoe u een nieuwe JNDI-databaseverbinding in Tomcat kunt maken.

Stap 1. Bepaal de Java Runtime Environment (JRE)-versie op uw CVP-server.

- Klik op de knop Start (linkerhoek op de bureaubladbalk)
- Klik op de optie Uitvoeren...
- Typ in CMD en klik op OK om de DOS-prompt te openen
- Typ vervolgens in Java-versie



```
C:\Users\administrator.DCLOUD>java -version
openjdk version "1.8.0_352-352"
OpenJDK Runtime Environment (build 1.8.0_352-352-b08)
OpenJDK Server VM (build 25.352-b08, mixed mode)

C:\Users\administrator.DCLOUD>
```

Stap 2. Installeer een compatibel Java Database Connectivity (JDBC)-stuurprogramma met de SQL-versie. Om toegang tot de database op uw toepassingsserver mogelijk te maken, moet een compatibel JDBC-stuurprogramma zijn geïnstalleerd. Deze stuurprogramma's, meestal verpakt als JAR-bestanden, moeten worden geplaatst in een directory die toegankelijk is voor het klassepada van de toepassingsserver (op Tomcat, bijvoorbeeld in %CVP_HOME%\VXMLServer\Tomcat\lib).

Verschillende drivers zijn [hier](#) te vinden.

Een voorbeeld uit de lijst met ondersteunde MS-systemen:

SQL version compatibility

 Expand table

Database version → ↓ Driver Version	Azure SQL Database	Azure Synapse Analytics	Azure SQL Managed Instance	SQL Server 2022	SQL Server 2019	SQL Server 2017	SQL Server 2016	SQL Server 2008 R2
12.6	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
12.4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
12.2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

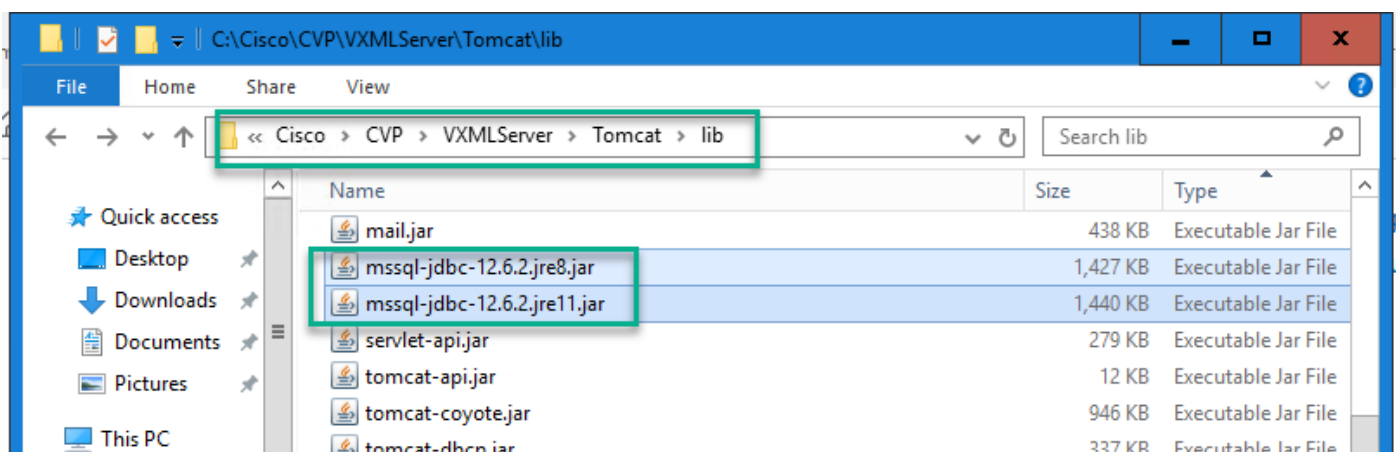
Java and JDBC specification support

 Expand table

JDBC Driver Version	JRE Versions	JDBC API Version
12.6	1.8, 11, 17, 21	4.2, 4.3 (partially)
12.4	1.8, 11, 17, 20	4.2, 4.3 (partially)
12.2	1.8, 11, 17, 19	4.2, 4.3 (partially)
11.2	1.8, 11, 17, 18	4.2, 4.3 (partially)
10.2	1.8, 11, 17	4.2, 4.3 (partially)
9.4	1.8, 11, 16	4.2, 4.3 (partially)



Opmerking: de database moet bestaan om deze verbinding te laten werken. CVP Voice XML (VXML) Server maakt de database niet voor u. In dit document wordt de UCCE AW-database als voorbeeld gebruikt en is versie 12.6 van het JDBC-stuurprogramma geïnstalleerd.



Stap 2. Voeg een Tomcat-context toe voor de databaseverbinding, zodat de CVP VXML-server weet hoe hij met uw database moet communiceren. Voor meer informatie, surf naar <https://tomcat.apache.org/tomcat-9.0-doc/jndi-datasource-examples-howto.html>.

Hier is een voorbeeld dat SQL gebruikt (bewerk context.xml vanuit AUDIUM_HOME\Tomcat\conf map waar AUDIUM_HOME gewoonlijk Cisco\CVP\VXMLServer is):

```
<Context>

<Resource name="jdbc/<LABEL_YOU_CHOOSE>"

auth="Container"

type="javax.sql.DataSource"

username="USER_NAME"

password="USER_PW"

driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"

url="jdbc:sqlserver://HOSTNAME_OR_IP:PORT;DatabaseName=< DB_NAME>" />

</Context>
```

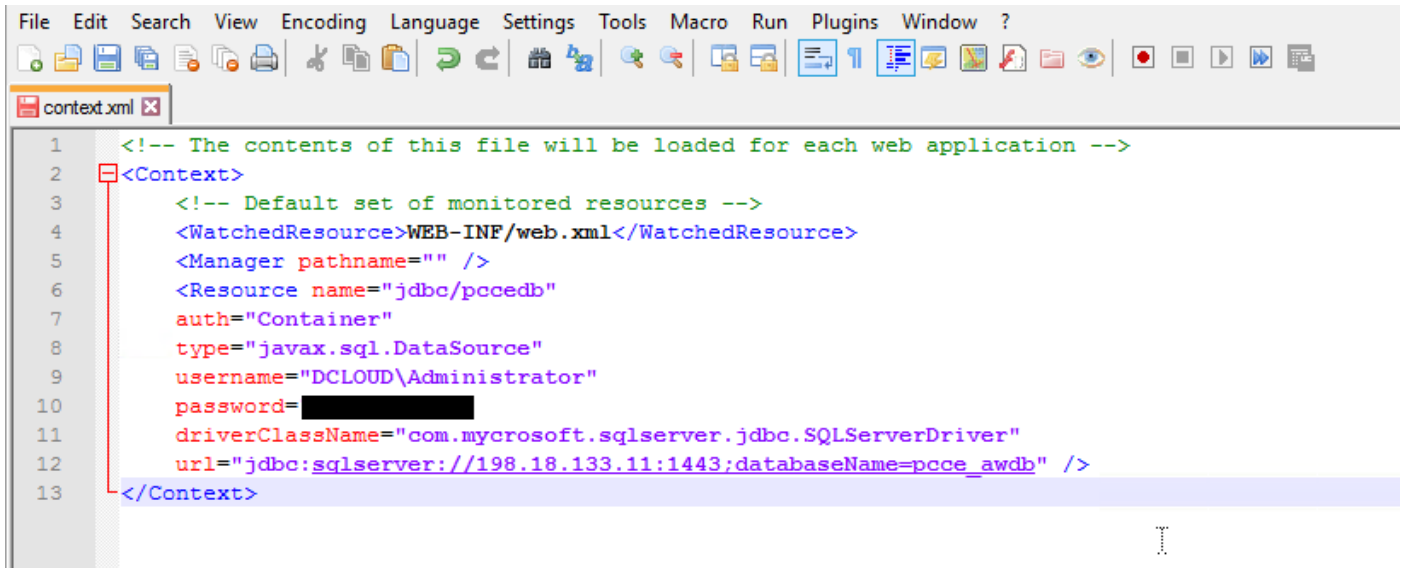
Het standaard poortnummer voor MS SQL is 1433. Een voorbeeld van een URL voor de eerdere context is 'jdbc: sql://localhost:1433;databaseName=pcce_awdb'.



Opmerking: Als alternatief kan `<Resource>` worden geconfigureerd in het bestand `server.xml` onder `<GlobalNamingResources>` en kan `<ResourceLink>` worden gemaakt in `context.xml` onder `<Context>`.

Voor een betere beveiliging is het raadzaam om de gebruikersnaam of het wachtwoord in te stellen met behulp van het element en de velden gebruikersnaam en wachtwoord handmatig te verwijderen uit het bestand `context.xml`.

Als de gebruikersnaam en het wachtwoord in het element worden opgegeven, worden de gebruikersnaam en het wachtwoord in het bestand `context.xml` genegeerd.

A screenshot of an IDE window titled 'context.xml'. The window shows an XML configuration for a database connection pool. The XML content is as follows:

```
1 <!-- The contents of this file will be loaded for each web application -->
2 <Context>
3   <!-- Default set of monitored resources -->
4   <WatchedResource>WEB-INF/web.xml</WatchedResource>
5   <Manager pathname="" />
6   <Resource name="jdbc/pccedb"
7     auth="Container"
8     type="javax.sql.DataSource"
9     username="DCLOUD\Administrator"
10    password="
11    driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
12    url="jdbc:sqlserver://198.18.133.11:1443;databaseName=pcce_awdb" />
13 </Context>
```

The IDE interface includes a menu bar (File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, ?) and a toolbar with various icons for file operations, editing, and running. A red bracket on the left side of the XML code highlights the entire configuration block from line 2 to line 13.

Step 3. Schakel onder zware belasting Database Connection Pooling in.

Een databaseverbindingspool maakt en beheert een pool van verbindingen met een database. Het recyclen en hergebruiken van reeds bestaande verbindingen met een database is efficiënter dan het openen van een nieuwe verbinding. Zie <https://tomcat.apache.org/tomcat-9.0-doc/jndi-datasource-examples-howto.html> voor meer informatie over het poolen van Tomcat-databases.



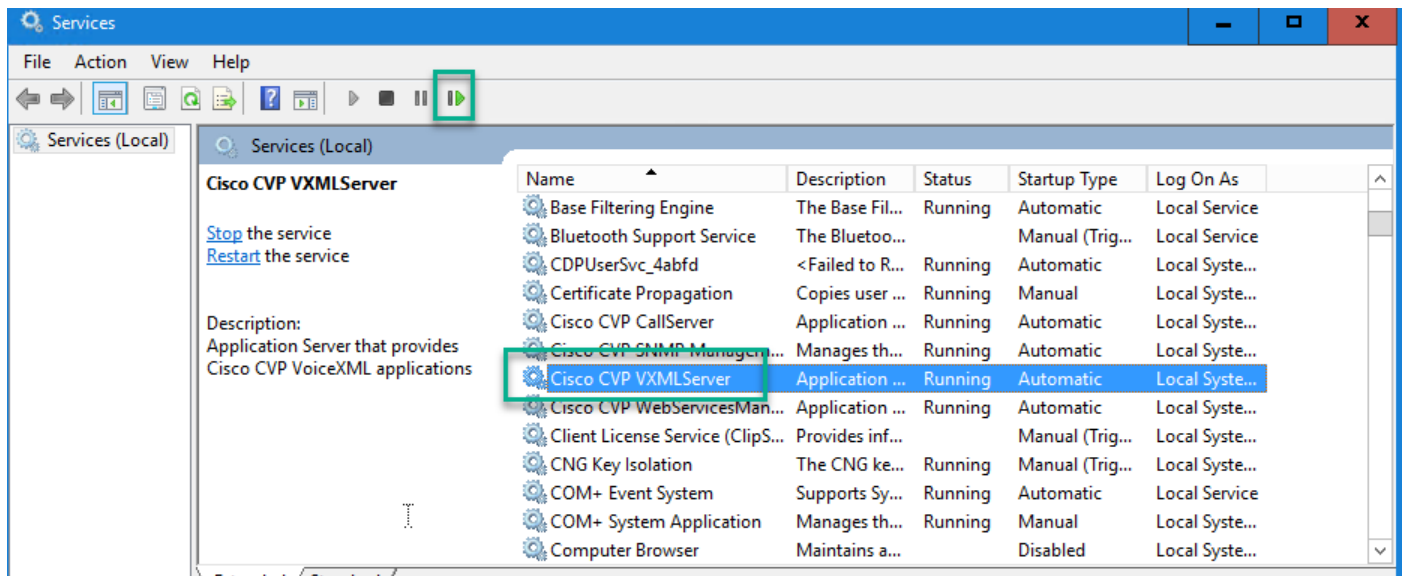
Opmerking: Tomcat 8.0 heeft twee bibliotheken voor verbindingsgroepen: commons-dbc en tomcat-jdbc-pool. Als de verbinding tussen de CVP VXML-server en de externe SQL-server uitvalt, worden de verbindingen niet automatisch hersteld als gevolg van een bekend probleem met de verbindingspoolbibliotheek van tomcat-jdbc-pool. De verbindingen kunnen alleen opnieuw tot stand worden gebracht nadat de VXMLServer tomcat-service opnieuw is gestart.

De commons-dbc verbinding pool bibliotheek heeft dit probleem niet. De commons-dbc-bibliotheek wordt standaard gebruikt en de tomcat-jdbc-pool wordt alleen gebruikt als het bestand tomcat context.xml deze regel bevat:

```
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
```

Vanwege dit probleem raadt Cisco het gebruik van de tomcat-jdbc-pool-bibliotheek niet aan.

Stap 4. Start de CVP VXML-service opnieuw op.



Stap 5. Bewerk in CVP Call Studio de configuratie van het desbetreffende database-element. Voer de tekenreeks in die u hebt ingevoerd in <LABEL_YOU_CHOICE> vanuit de Tomcat-context in de eigenschap JNDI-naam van het tabblad Instellingen van het element Database.



Opmerking: neem hier niet het jdbc / gedeelte op.

Builder - Tomcat_Element_Example/app.callflow - Cisco Unified Call Studio

File Edit Callflow Navigate Search Project Run Window Help

Navigator

- Copy of Test
- Example_Project
- Test
- Tomcat_Element_Example
 - callflow
 - deploy
 - app.callflow

Elements

- Currency
- Currency_With_Confirm
- Context
- Application_Modifier
- Date & Time
- Date

*app.callflow

Start Of Ca

Database_01

Element Configuration

Action Element - Database

General Settings Data Events

Name	Value
* Type	
* JNDI Name	<LABEL_YOU_CHOOSE>
* SQL Query	
* Result-Set XML	true
* Max Rows in XML	50
Username	
Password	

Stap 6. De toepassing opslaan, implementeren en bijwerken op de VXML-server.

Problemen oplossen

Er is momenteel geen specifieke troubleshooting-informatie beschikbaar voor deze configuratie.

Gerelateerde informatie

- Microsoft JDBC Driver voor SQL Server-ondersteuningsmatrix: [Microsoft JDBC-stuurprogramma's](#)
- VXML-elementengids: [CVP VXML-elementengids](#)
- [Technische ondersteuning en documentatie – Cisco Systems](#)

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.