

Konfigurieren eines Database Call Studio-Elements - Tomcat JNDI für SQL-Datenbanken

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Konfiguration](#)

[Fehlerbehebung](#)

[Zugehörige Informationen](#)

Einleitung

In diesem Dokument wird das Configure Customer Voice Portal (CVP) Call Studio Database Element, Tomcat Java Naming and Directory Interface (JNDI), für die SQL-Datenbank beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Cisco Unified Contact Center Enterprise (UCCE) Version 12.6.2
- Cisco Package Contact Center Enterprise (PCCE) Version 12.6.2
- CVP-Version 12.6.2
- CVP Call Studio 12.6.2

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf folgenden Software-Versionen:

- Cisco PCCE Version 12.6.2
- CVP-Version 12.6.2
- CVP Call Studio 12.6.2

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Das Datenbankelement bietet die Möglichkeit, einen SQL-Befehl (Structured Query Language) auf externen Datenbanken innerhalb eines Sprachanwendungsaufflusses auszuführen. Das Element erfordert, dass JNDI im Java-Anwendungsserver konfiguriert wird, um Datenbankverbindungen zu verarbeiten. Pro Element kann nur eine einzelne SQL-Anweisung ausgeführt werden. Es gibt vier Arten von Befehlen, die ausgeführt werden können, in diesem Dokument wird jedoch nur der einzelne Befehl verwendet:

Single - Wird verwendet, um eine SQL-Abfrage auszuführen, die nur eine Zeile zurückgibt. Elementdaten werden erstellt, wobei die Variablennamen die Namen der zurückgegebenen Spalten und der Wert dieser Spalte als Elementdatenwert (als Zeichenfolge) sind. Wenn keine Zeile zurückgegeben wird, werden keine Elementdaten festgelegt.

Weitere Informationen finden Sie im [CVP VXML Element](#)-Leitfaden.

Konfiguration

In diesem Abschnitt wird erläutert, wie Sie eine neue JNDI-Datenbankverbindung in Tomcat erstellen.

Schritt 1: Bestimmen Sie die Java Runtime Environment (JRE)-Version auf Ihrem CVP-Server.

- Klicken Sie auf die Schaltfläche Start (linke Maustaste auf der Symbolleiste).
- Klicken Sie auf die Option Ausführen...
- Geben Sie CMD ein, und klicken Sie auf OK, um die DOS-Eingabeaufforderung aufzurufen.
- Geben Sie dann java-version ein

```
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>
```

Schritt 2: Installieren Sie einen kompatiblen JDBC-Treiber (Java Database Connectivity) mit der SQL-Version. Um den Datenbankzugriff auf dem Anwendungsserver zu ermöglichen, muss ein kompatibler JDBC-Treiber installiert sein. Diese Treiber, die normalerweise als JAR-Dateien verpackt werden, müssen in einem Verzeichnis abgelegt werden, auf das der Klassenpfad des Anwendungsservers zugreifen kann (auf Tomcat beispielsweise in %CVP_HOME%\VXMLServer\Tomcat\lib).

Verschiedene Treiber finden Sie [hier](#).

Ein Beispiel aus der Liste der MS-unterstützten Geräte:

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

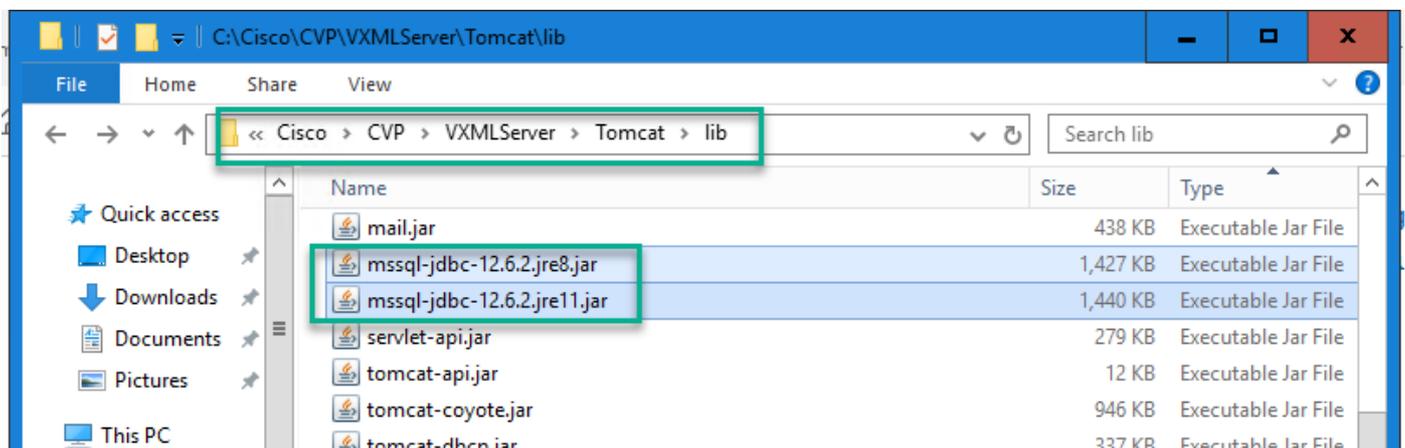
Java and JDBC specification support

 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)



Anmerkung: Damit diese Verbindung funktioniert, muss die Datenbank vorhanden sein. Der CVP Voice XML (VXML)-Server erstellt die Datenbank nicht für Sie. In diesem Dokument wird die UCCE AW-Datenbank als Beispiel verwendet. Die installierte Version des JDBC-Treibers ist Version 12.6.



Schritt 2: Fügen Sie einen Tomcat-Kontext für die Datenbankverbindung hinzu, damit der CVP VXML-Server weiß, wie er mit Ihrer Datenbank kommunizieren soll. Weitere Informationen finden Sie unter <https://tomcat.apache.org/tomcat-9.0-doc/jndi-datasource-examples-howto.html>.

Das folgende Beispiel verwendet SQL (context.xml aus dem Ordner AUDIUM_HOME\Tomcat\conf bearbeiten, wobei AUDIUM_HOME normalerweise Cisco\CVP\VXMLServer ist):

```
<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>
```

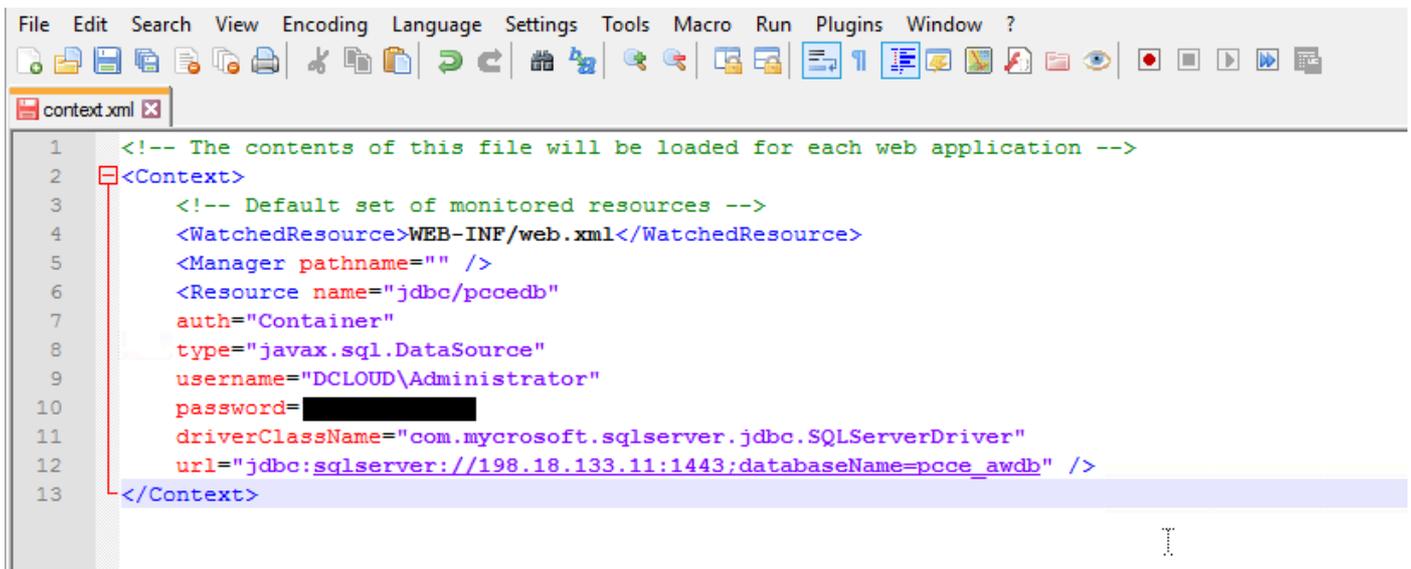
Die Standard-Portnummer für MS SQL ist 1433. Eine Beispiel-URL für den früheren Kontext ist "jdbc:sql://localhost:1433;databaseName=pcce_awdb'



Anmerkung: Alternativ kann die `<Resource>` in der Datei `server.xml` unter `<GlobalNamingResources>` konfiguriert werden, und ein `<ResourceLink>` kann in `context.xml` unter `<Context>` erstellt werden.

Zur Erhöhung der Sicherheit wird empfohlen, den Benutzernamen oder das Kennwort mithilfe des Elements festzulegen und die Felder für den Benutzernamen und das Kennwort manuell aus der Datei `context.xml` zu löschen.

Wenn der Benutzername und das Kennwort im Element angegeben werden, werden der Benutzername und das Kennwort in der Datei `context.xml` ignoriert.



```
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="DCLLOUD\Administrator"
10 password="
11 driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
12 url="jdbc:sqlserver://198.18.133.11:1443;databaseName=pccce_awdb" />
13 </Context>
```

Schritt 3: Aktivieren Sie bei starker Auslastung das Datenbankverbindungs-Pooling.

Ein Datenbankverbindungs-pool erstellt und verwaltet einen Pool von Verbindungen mit einer Datenbank. Das Wiederverwenden und Wiederverwenden bereits vorhandener Verbindungen zu einer Datenbank ist effizienter als das Öffnen einer neuen Verbindung. Weitere Informationen zu Tomcat Database Pooling finden Sie unter <https://tomcat.apache.org/tomcat-9.0-doc/jndi-datasource-examples-howto.html>.



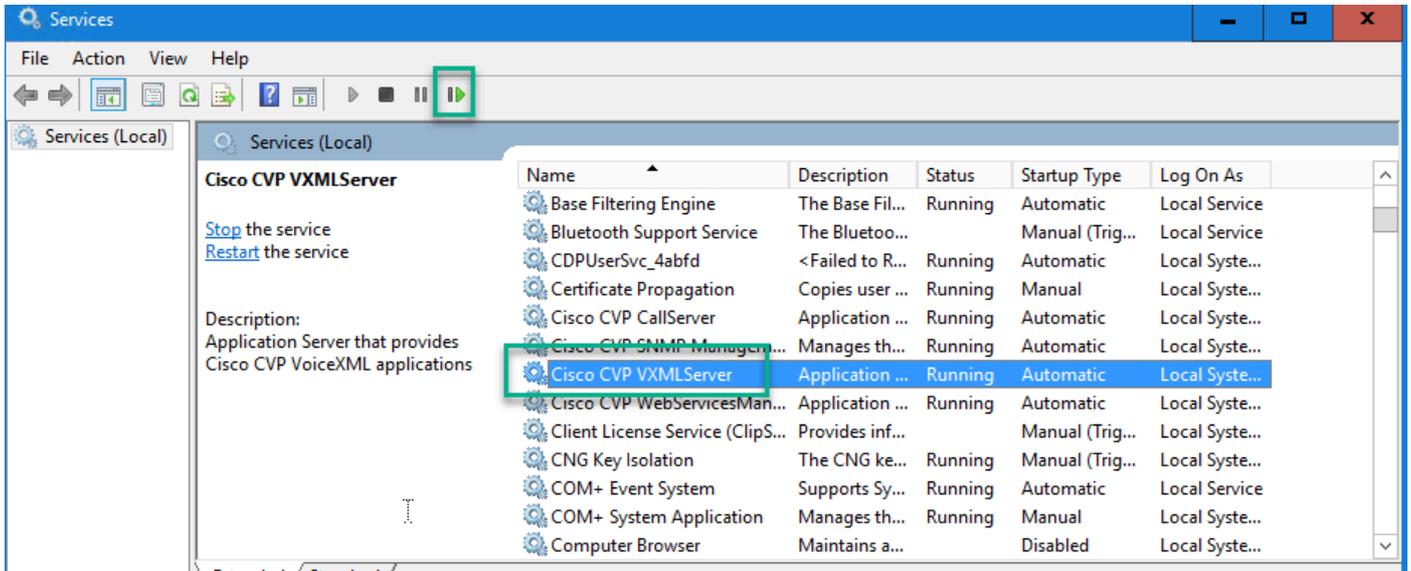
Anmerkung: Tomcat 8.0 verfügt über zwei Verbindungspoolbibliotheken: commons-dbcp und tomcat-jdbc-pool. Aufgrund eines bekannten Problems mit der Verbindungspoolbibliothek tomcat-jdbc-pool werden die Verbindungen nicht automatisch wiederhergestellt, wenn die Verbindung zwischen dem CVP VXML-Server und dem Remote-SQL-Server ausfällt. Die Verbindungen können erst wiederhergestellt werden, nachdem der Dienst VXMLServer tomcat neu gestartet wurde.

Die Verbindungsbibliothek commons-dbcp weist dieses Problem nicht auf. Die Bibliothek commons-dbcp wird standardmäßig verwendet, und der Pool tomcat-jdbc wird nur verwendet, wenn die Datei tomcat context.xml folgende Zeile enthält:

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

Aufgrund dieses Problems empfiehlt Cisco nicht, die Bibliothek tomcat-jdbc-pool zu verwenden.

Schritt 4: Starten Sie den CVP VXML-Service neu.



Schritt 5: Bearbeiten Sie in CVP Call Studio die Konfiguration des betreffenden Database-Elements. Geben Sie die Zeichenfolge, die Sie in <LABEL_YOU_CHOOSE> aus dem Tomcat-Kontext eingegeben haben, in die JNDI Name-Eigenschaft auf der Registerkarte Einstellungen des Database-Elements ein.



Anmerkung: Fügen Sie den jdbc/part hier nicht ein.

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	

Schritt 6: Speichern, Bereitstellen und Aktualisieren der Anwendung auf dem VXML-Server.

Fehlerbehebung

Für diese Konfiguration sind derzeit keine spezifischen Informationen zur Fehlerbehebung verfügbar.

Zugehörige Informationen

- Microsoft JDBC-Treiber für SQL Server-Unterstützungsmatrix: [Microsoft JDBC-Treiber](#)
- VXML Elementführung: [CVP VXML-Elementleitfaden](#)
- [Technischer Support und Dokumentation für Cisco Systeme](#)

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.