

Configurare un elemento Studio chiamata di database - Tomcat JNDI per il database SQL

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Configurazione](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

Introduzione

Questo documento descrive l'elemento di database Call Studio di CVP (Configure Customer Voice Portal), l'interfaccia JNDI (Java Naming and Directory Interface) di Tomcat per il database SQL.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

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

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software:

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

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

L'elemento di database consente di eseguire un comando SQL (Structured Query Language) su database esterni all'interno di un flusso di chiamate dell'applicazione vocale. L'elemento richiede la configurazione di JNDI nel server applicazioni Java per gestire le connessioni al database. È possibile eseguire una sola istruzione SQL per elemento. È possibile eseguire quattro tipi di comandi, ma in questo documento viene utilizzato solo il comando singolo:

Singola: viene utilizzata per eseguire una query SQL che restituisce una sola riga. I dati dell'elemento verranno creati con i nomi delle variabili corrispondenti ai nomi delle colonne restituite e il valore di tale colonna come valore dei dati dell'elemento (come stringa). Se non viene restituita alcuna riga, non verrà impostato alcun dato elemento.

Per ulteriori informazioni, consultare la guida [CVP VXML Element](#).

Configurazione

Questa sezione spiega come creare una nuova connessione al database JNDI in Tomcat.

Passaggio 1. Determinare la versione di Java Runtime Environment (JRE) sul server CVP.

- Fare clic sul pulsante Start (pulsante sinistro sulla barra del desktop)
- Fare clic sull'opzione Esegui...
- Digitare CMD e fare clic su OK per visualizzare il prompt di DOS
- Digitare quindi in versione Java

```
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>
```

Passaggio 2. Installare un driver JDBC (Java Database Connectivity) compatibile con la versione SQL. Per abilitare l'accesso al database sul server applicazioni, è necessario installare un driver JDBC compatibile. Questi driver, generalmente collocati come file JAR, devono essere collocati in una directory accessibile al classpath del server applicazioni (su Tomcat, ad esempio, posizionati in %CVP_HOME%\VXMLServer\Tomcat\lib).

[Qui](#) sono disponibili driver diversi.

Di seguito è riportato un esempio dall'elenco degli Stati membri supportati:

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

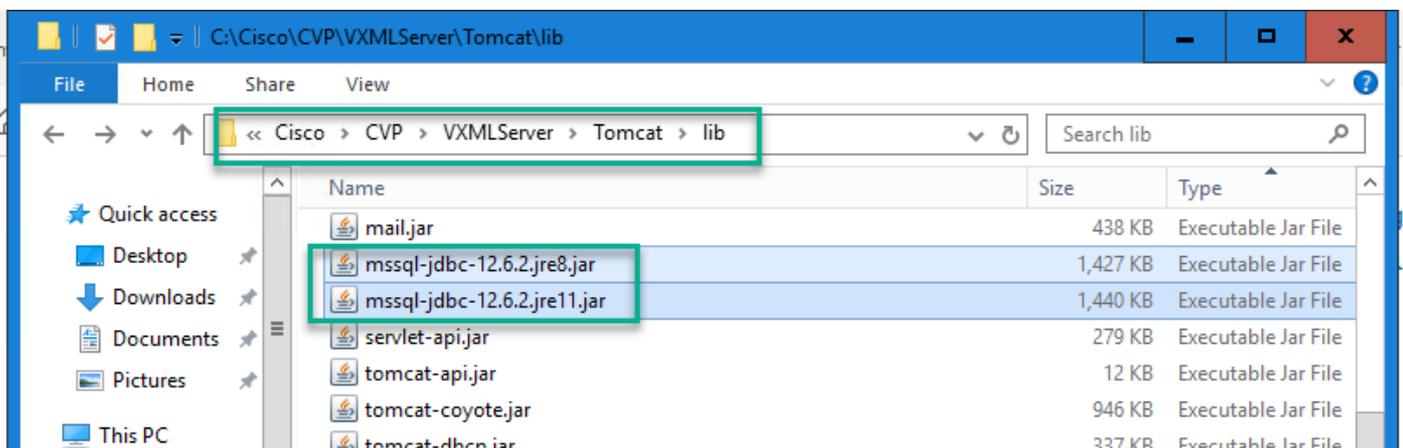
Java and JDBC specification support

 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)



Nota: Affinché la connessione funzioni, è necessario che il database esista. Il server VXML (CVP Voice XML) non creerà il database automaticamente. In questo documento viene utilizzato come esempio il database UCCE AW e viene installato il driver JDBC versione 12.6.



Passaggio 2. Aggiungere un contesto Tomcat per la connessione al database in modo che il server VXML CVP possa comunicare con il database. Per ulteriori informazioni, visitare il sito Web all'indirizzo <https://tomcat.apache.org/tomcat-9.0-doc/jndi-datasource-examples-howto.html>.

Di seguito è riportato un esempio che utilizza SQL (edit context.xml dalla cartella AUDIUM_HOME\Tomcat\conf dove AUDIUM_HOME è in genere Cisco\CVP\VXMLServer):

```
<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>
```

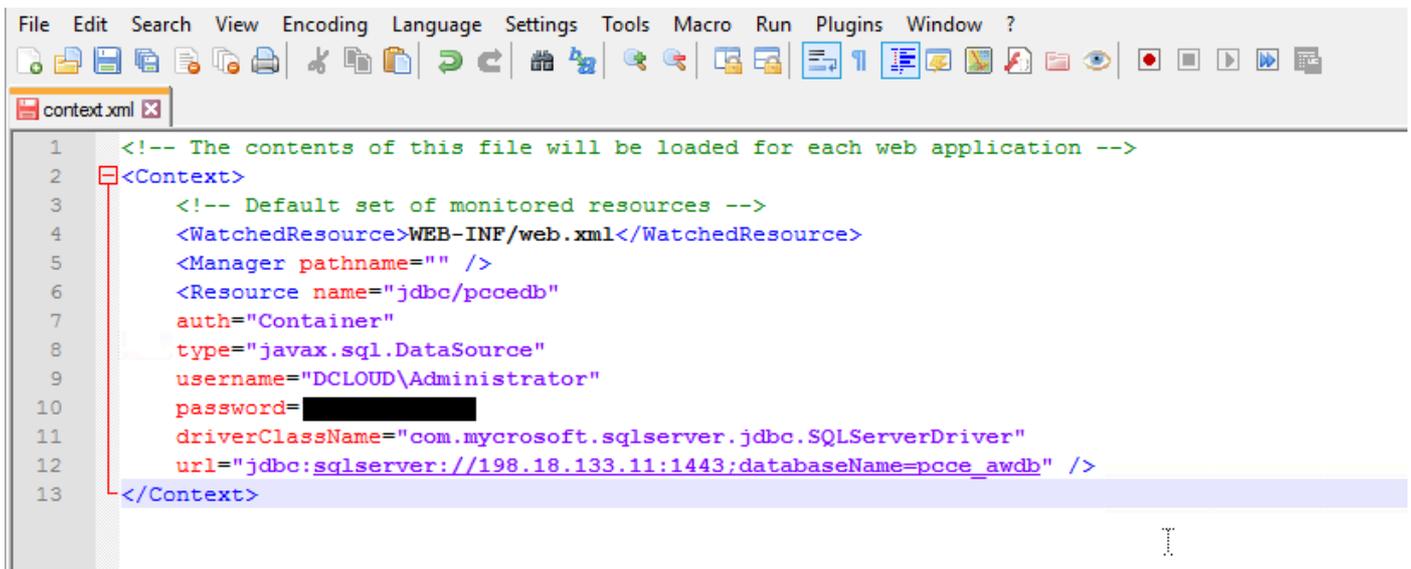
Il numero di porta predefinito per MS SQL è 1433. Un URL di esempio per il contesto precedente sarà 'jdbc:sql://localhost:1433;databaseName=pcce_awdb'.



Nota: In alternativa, è possibile configurare `<Resource>` nel file `server.xml` in `<GlobalNamingResources>` e un `<ResourceLink>` creato in `context.xml` in `<Context>`.

Per una protezione avanzata, è consigliabile impostare il nome utente o la password utilizzando l'elemento ed eliminare manualmente i campi nome utente e password dal file `context.xml`.

Se il nome utente e la password vengono specificati nell'elemento, il nome utente e la password nel file `context.xml` verranno ignorati.



```
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=pcc_e_awdb" />
13 </Context>
```

Passaggio 3. In condizioni di carico elevato, abilitare il connection pooling del database.

Un connection pool di database crea e gestisce un pool di connessioni a un database. Il riciclo e il riutilizzo delle connessioni già esistenti in un database è più efficiente rispetto all'apertura di una nuova connessione. Per ulteriori informazioni sul pooling del database Tomcat, vedere <https://tomcat.apache.org/tomcat-9.0-doc/jndi-datasource-examples-howto.html>.



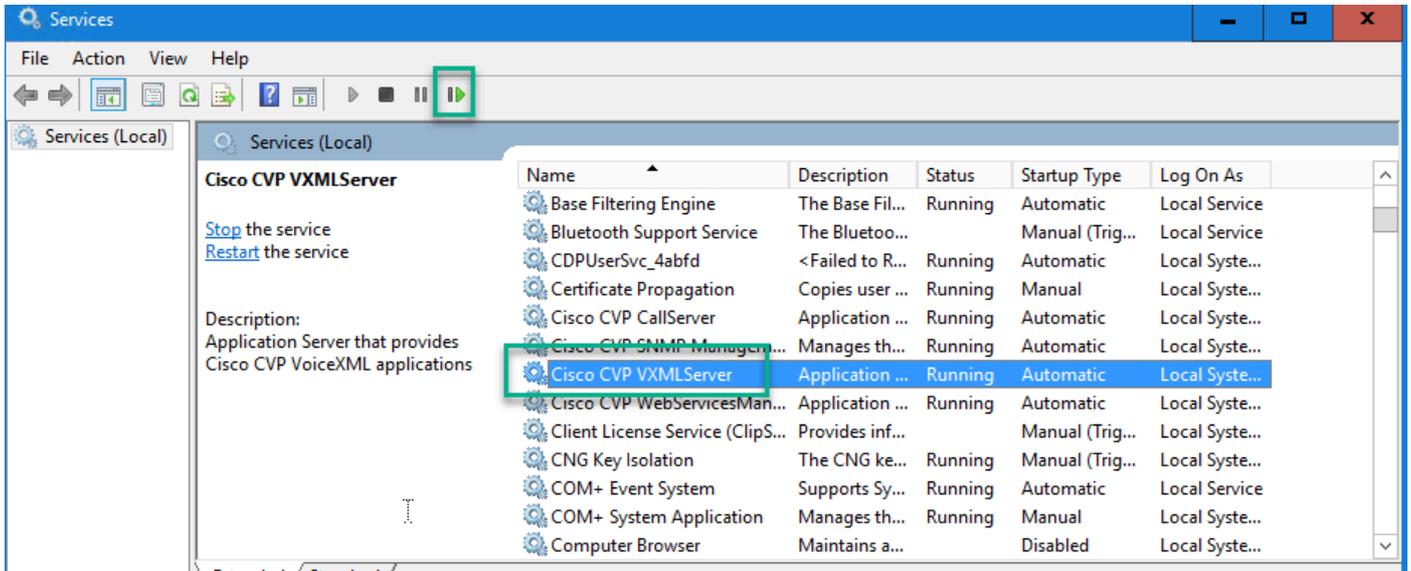
Nota: Tomcat 8.0 dispone di due librerie di connection pool: commons-dbcp e tomcat-jdbc-pool. A causa di un problema noto con la libreria del pool di connessioni tomcat-jdbc, se la connessione tra il server VXML CVP e il server SQL remoto non è attiva, le connessioni non vengono ristabilite automaticamente. Le connessioni possono essere ristabilite solo dopo il riavvio del servizio VXMLServer Tomcat.

Il problema non si verifica nella libreria del pool di connessioni commons-dbcp. La libreria commons-dbcp viene utilizzata per impostazione predefinita e il pool tomcat-jdbc viene utilizzato solo se il file context.xml tomcat contiene la riga seguente:

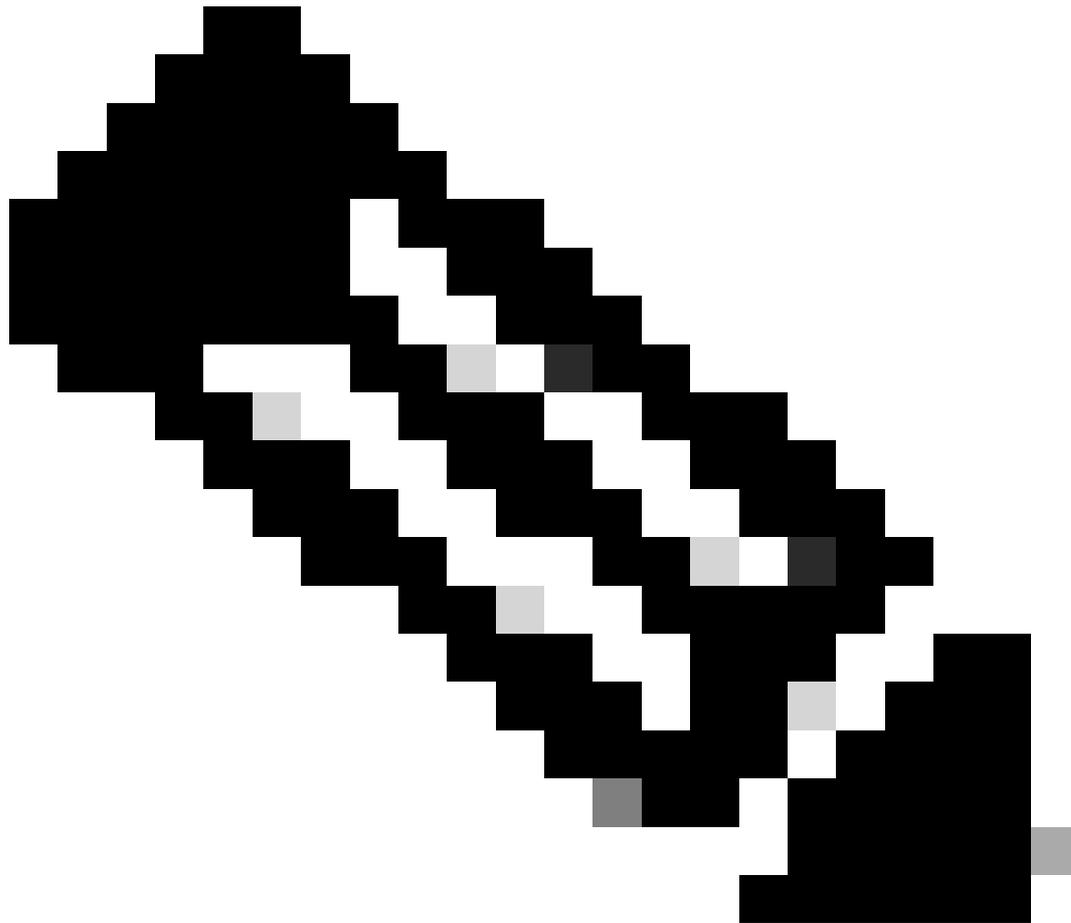
```
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
```

A causa di questo problema, Cisco sconsiglia di utilizzare la libreria tomcat-jdbc-pool.

Passaggio 4. Riavviare il servizio VXML CVP.



Passaggio 5. In CVP Call Studio, modificare la configurazione dell'elemento del database in questione. Immettere la stringa immessa in <LABEL_YOU_CHOOSE> dal contesto Tomcat nella proprietà JNDI Name della scheda Settings dell'elemento Database.



Nota: Non includere la parte jdbc/hdbc in questo punto.

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	

Passaggio 6. Salvare, distribuire e aggiornare l'applicazione nel server VXML.

Risoluzione dei problemi

Al momento non sono disponibili informazioni specifiche per la risoluzione dei problemi di questa configurazione.

Informazioni correlate

- Matrice di supporto del driver JDBC Microsoft per SQL Server: [Driver JDBC Microsoft](#)
- Guida agli elementi VXML: [Guida agli elementi VXML CVP](#)
- [Documentazione e supporto tecnico – Cisco Systems](#)

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).