# CISpaces Version Control Web Service

## CISpaces Technical Overview

The CISpaces software consists of several web services running on Apache Tomcat (8.0.30) and a thin web client implemented mainly in JavaScript. The web services are deployed on the Tomcat server by placing the .war file for each web service under Tomcat’s web apps directory.

The web services communicate with an underlying Derby database using the JDBC drivers for Derby. To configure a newly installed Tomcat server to recognize the connection to a Derby database via JDBC, you should place the following script in the context.xml file located in the \*conf\* directory of Tomcat (you can place it straight after the <Context> tag):

<Resource name="jdbc/myDB"

auth="Container"

type="javax.sql.DataSource"

username="gaiandb"

password="passw0rd"

driverClassName="org.apache.derby.jdbc.ClientDriver"

url="jdbc:derby://localhost:6414/gaiandb"

maxTotal="10"

removeAbandonedOnBorrow="true"

removeAbandonedTimeout="60"

logAbandoned="true"

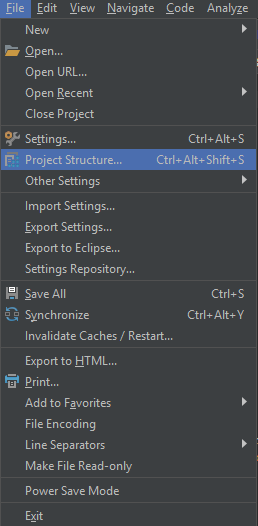
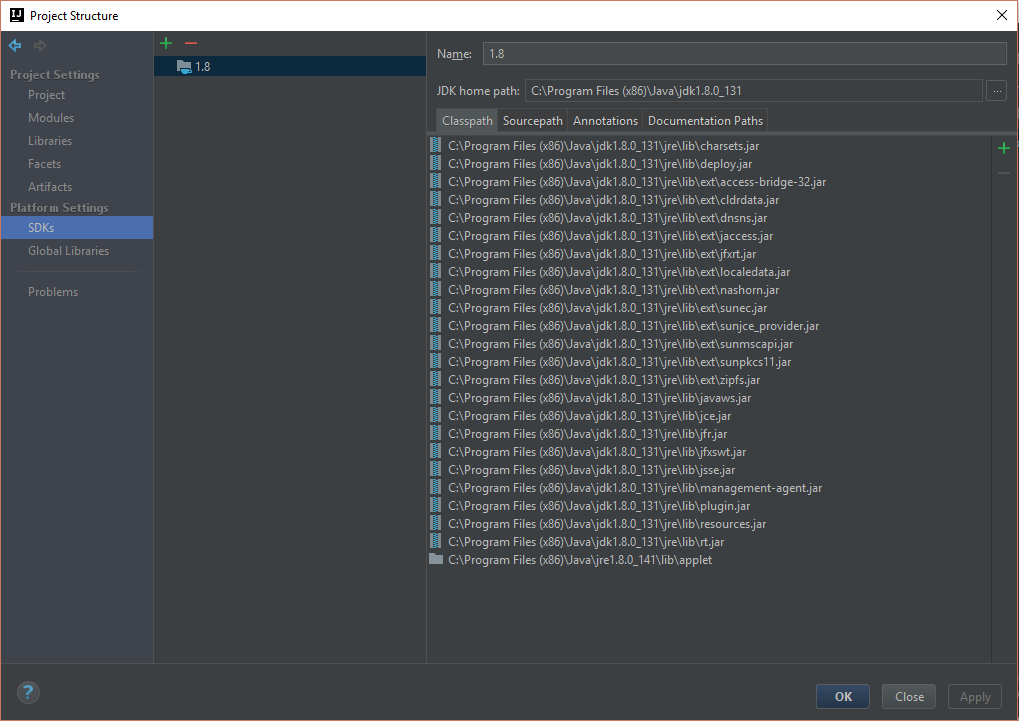
/>

The Derby database communicates with the web services on port 6414. Make sure this port is open and check your firewall settings to make sure it can be utilized by the application. If you want to get familiar with the communication mechanism established between the Derby database and the web services, check out the DBConnect.java file in the database package of the Version Control web service source code.

## Downloading the Version Control web service source

The VC web service project is hosted in a GitLab repository under the web services project. The branch is called VC\_Experimental. Simply clone the branch and open the project with your favorite IDE. The VC web service has been implemented using the JetBrains IntelliJ Idea 2017.1.4, while the rest of the web services were implemented using Eclipse.

You will need to configure the artifacts, the xml descriptor and the libraries used. If you are using IntelliJ, this is how you can configure the project:

1. Open the VC project in IntelliJ Idea.
2. Open the Project Structure:
3. Configure the SDK to Java (I am using the 32-bit jdk):

1. Select the Facets from the Project Settings and configure the xml descriptor:

A screenshot of a computer screen

Description generated with very high confidence

1. A screenshot of a cell phone screen with text

   Description generated with very high confidenceSelect Modules in the Project Structure, click on the \*+\* and select web:
2. This is what the Deployment Descriptors should look like:

A screenshot of a computer screen

Description generated with very high confidence

1. Select Artifacts and click on the \*+\*. Choose Web Application: Exploded. Rename it to VC: Web exploded and make sure the output directory is like the picture below. From the Available

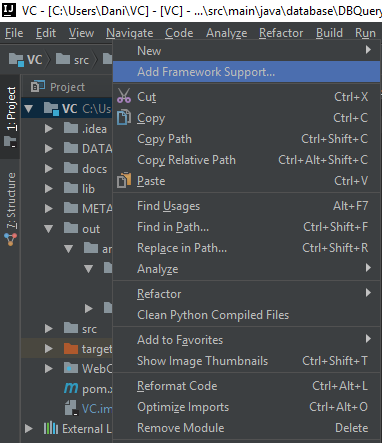
Elements, right click on the VC Compiled Output, and select \*Put into WEB-INF/classes\*. Then right-click on the web descriptor and select \*Pin into Output Root\*.A screenshot of a computer screen

Description generated with very high confidence

1. Similarly, click on the \*+\* sign and select Web Application: Archive. Select \*For VC/Web exploded\*. If it asks for META-INF, locate it under VC\WebContent\META-INF\MANIFEST.MF and select it.

A screenshot of a computer screen

Description generated with very high confidence

1. Close the Project Structure and right-click on the project, then click on ‘Add Framework Support’. Locate Maven and make sure to enable auto imports of libraries.
2. Go back to Project Structure/Libraries. Make sure Maven libraries are listed there:

A screenshot of a computer screen

Description generated with very high confidence

To configure Apache Tomcat in IntelliJ, follow the steps below.

1. Select Run/Edit Configurations:

A screenshot of text

Description generated with very high confidence

1. Click on the\*+\* and select Tomcat Server/Local. In the Application Server, click on ‘Configure’ and locate the directory of the Tomcat on your machine. Then click on the \*Deployment\* tab, and click on the \*+\* and click on Artifacts. Select both the exploded was and the .war files. Make sure the application context of the VC archive (VC.war) is pointed to / (Root), and for the VC exploded war to /VC.

A screenshot of a computer screen

Description generated with very high confidence

1. Click on OK. Now every time you start the Tomcat through IntelliJ, it will rebuild the deployment artifacts. Note that setting the application context to / will overwrite your ROOT web app in Tomcat. You can configure the application contexts yourself in a way you see fit for purpose.