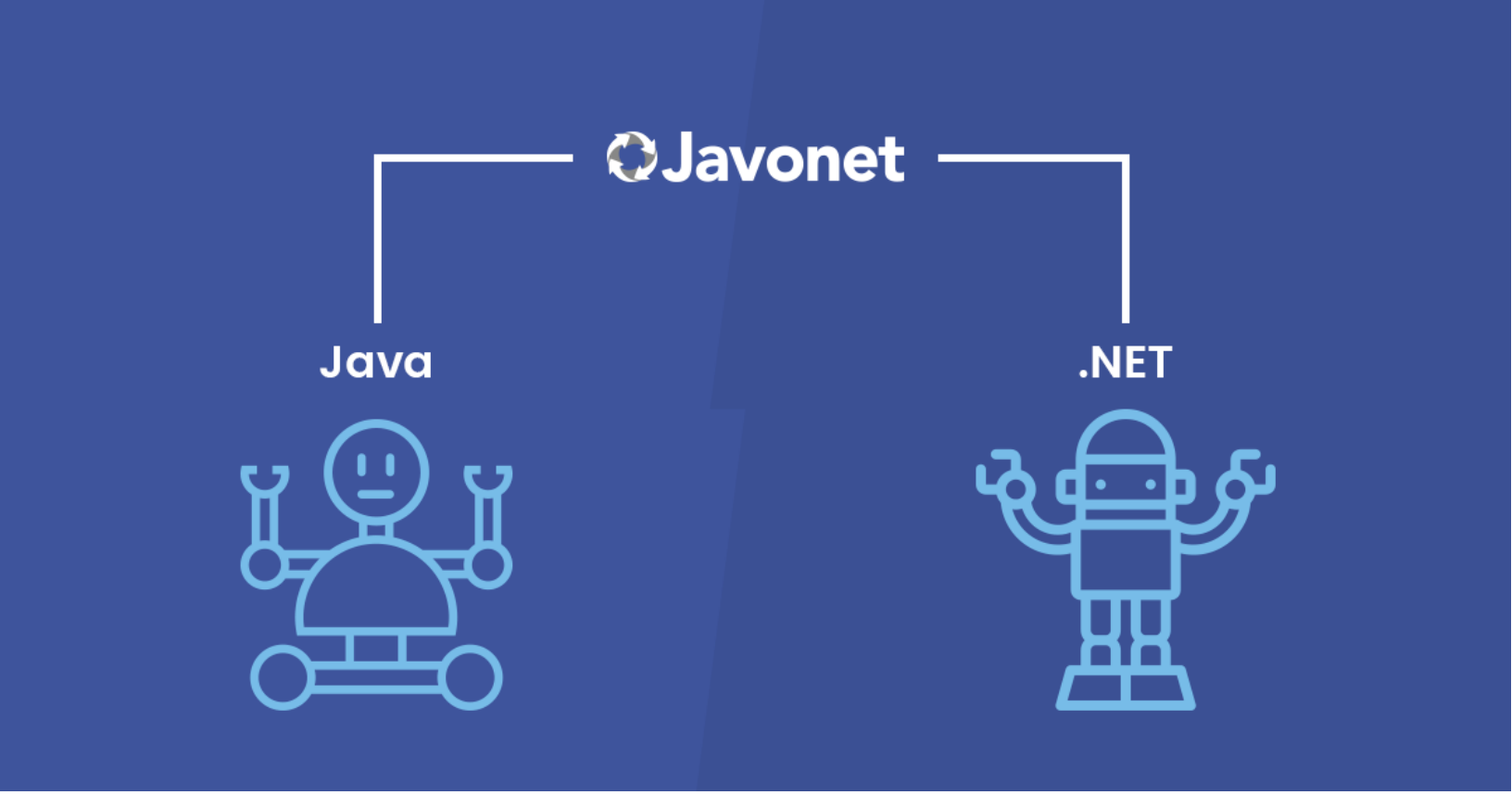

combit List & Label 24 .NET component for Java Developers

combit List & Label 24 .NET component is now accessible also for Java developers. combit .NET UI components can be used from Java thanks to third-party Javonet component which allows to use any .NET UI component in Java AWT, Swing and JavaFX applications.

List & Label 24 components can be used directly through Javonet API or using the strongly typed wrapper. The wrapper is a ready to use compiled Java Jar package which replicates the combit List & Label 24 .NET API in Java. You will find the ready to use wrapper in combit install directory **combit\LL24\Samples\Java\DataBindingSample\combit.listlabel24.jar**

Authored by: Javonet Team
www.javonet.com

Technical support:
Email: support@javonet.com
Phone USA: +1 650 924 9915
Phone Europe & Asia: +48 22 379 24 74

SdNcenter sp. z o.o.
Franciszka Klimczaka 1
02-797 Warsaw, Poland

Contents

1 Using combit List & Label in Java Project

To use the combit List & Label in Java application copy the combit.listlabel24.jar, combit.listlabel24.dll and Javonet-1.5.jar (or newer) to root directory of your Java project and add “combit.listlabel24.jar” to project build path.

You can find the sample Java application in **combit\LL24\Samples\Java\DataBindingSample**

To get your own copy of Javonet Jar package you should register for Javonet free trial or commercial license. Go to http://javonet.link/get-trial to register for free trial and download Jar.

2 combit List & Label Sample Project

2.1 How to Run the Java Sample Project using Eclipse

Below you will find step by step guide how to run the Java sample project.

1. Open **Eclipse IDE**.

Because by default the source of the sample does not include Eclipse project files you need to create new workspace and import the project.

2. **Create or open Eclipse Workspace**.
3. Right click on Project Explorer and choose **Import** next **General** > **Projects from Folder or Archive** and press **Next**.
4. Press **Directory...** and point to the **DataBindingSample** within the **combit\LL24\Samples\Java** folder. Project should be automatically detected. Press **Finish** to complete.

Now your project is ready, you only need to add missing dependencies and set Javonet activation key.

6. Register for free trial Javonet key Signup for Javonet.
7. After registration is completed download **Javonet Jar** and copy to the Eclipse project.
8. Right click **Javonet Jar** in Eclipse Project Explorer and choose **Build Path** > **Add to Build Path**
9. Right click **combit.ListLabel24.jar** in Eclipse Project Explorer and choose **Build Path** > **Add to Build Path**
10. **Set your email and Javonet license key** in **DataBindingSample/src/Form1.java** file:

*ListLabelActivation.setLicense(*****your-email-here*****, *****your-license-key-here*****);*

Now you can run the sample.

2.2 Strongly Typed Wrapper

Strongly-typed wrapper replicates List & Label 24 .NET API for Java, currently it exposes only the key operations required for this sample:

- using a List & Label .NET Dataprovider to connect to an Access database with OleDB and respond to an event of the Dataprovider
- calling/opening the List & Label Designer; including real data preview
- printing into the List & Label preview control on the form and respond to a preview control button click event
- exporting into any of the supported List & Label export formats

You can further extend this wrapper to expose the strongly typed methods with signatures matching the combit List & Label .NET component (see also the .NET help at https://docu.combit.net/net/en) and forwarding the calls via Javonet.

2.3 Extending the wrapper

Wrapper project **combit.listlabel24** source code with ready to use ANT build script to compile the Jar, is available on GitHub repository: https://github.com/Javonet/combit-listlabel24-java. You can clone this repository to cover additional features of List & Label component.

Read the **combit-listlabe24-java** repository readme file for more details. To learn more how to perform different types of calls from Java to .NET API using Javonet, go to Javonet Guides for Java developers (https://www.javonet.com/java-devs/guides/).