# Section 10: CLIPS Java Native Interface

This section describes the CLIPS Java Native Inteface (CLIPSJNI) and the examples demonstrating the integration of CLIPS with a Swing interface. The examples have been tested using Java version 1.8.0\_31 running on Mac OS X 10.10.2 and Windows 7.

## **10.1 CLIPSJNI Directory Structure**

In order to use CLIPSJNI, you must obtain the source code by downloading the CLIPSJNI zip file from the Files page on the CLIPS SourceForge web page (see appendix A for the SourceForge URL). Once downloaded, you must then extract the contents of the file by right clicking on it and selecting the "Extract All..." menu item.

When unzipped the CLIPSJNI project file contains the following directory structure:

CLIPSJNI examples AnimalDemo resources AutoDemo resources SudokuDemo resources WineDemo resources java-src CLIPSJNI library-src

If you are using the CLIPSJNI on Mac OS X, then the native CLIPS library is already contained in the top level CLIPSJNI directory.

On Windows, it is necessary to verify that the correct DLL is installed. By default, the DLL for 64-bit Windows is used as the CLIPSJNI.dll file in the top level of the CLIPSJNI directory. If running CLIPSJNI with 32-bit Windows, delete the existing CLIPSJNI.dll file, then make a copy of the CLIPSJNI32.dll file and rename it to CLIPSJNI.dll.

On other systems, you must created a native library using the source files contained in the library-src directory before you can utilize the CLIPSJNI.

The CLIPSJNI jar file is also contained in the top level CLIPSJNI directory. The source files used to create the jar file are contained in the java-src directory.

## **10.2 Running CLIPSJNI in Command Line Mode**

You can invoke the command line mode of CLIPS through CLIPSJNI to interactively enter commands while running within a Java environment.

On Windows 7, launch the Command Prompt application (select Start > All Programs > Accessories > Command Prompt). Set the directory to the CLIPSJNI top level directory (using the cd command).

On Mac OS X, launch the Terminal application (located in the Applications/Utilities directory). Set the directory to the CLIPSJNI top level directory (using the cd command).

From the CLIPSJNI directory, enter the following command:

java -cp CLIPSJNI.jar CLIPSJNI.Environment

The CLIPS banner and command prompt should appear:

CLIPS (6.30 3/17/15)

CLIPS>

# 10.3 Running the Swing Demo Programs

The Swing CLIPSJNI demonstration programs can be run on Windows 7 or Mac OS X using the precompiled native libraries in the CLIPSJNI top level directory. On other systems a native library must first be created before the programs can be run.

#### 10.3.1 Running the Demo Programs on Mac OS X

Launch the Terminal application (located in the Applications/Utilities directory). Set the directory to the CLIPSJNI/examples/SudokuDemo directory (using the cd command). To run the Sudoku demo, enter the following command:

java -cp .:../../CLIPSJNI.jar -Djava.library.path=../.. SudokuDemo

The Sudoku Demo window should appear:



To run the Wine demo, set the directory to the CLIPSJNI/examples/WineDemo directory and enter the following command:

java -cp .:../../CLIPSJNI.jar -Djava.library.path=../.. WineDemo

The Wine Demo window should appear:

Prefe	rences	Meal			
Color: Don't Care		÷	Main Course:	Don't Know 💲	
Body:	Don't Care	+	Sauce:	Don't Know 💲	
Sweetness:	Don't Care	\$	Flavor:	Don't Know 📫	
Wine			Recommendation We	ight	
Chardonnay					
Riesling					
Soave					
Chenin Blanc					
Gamay					
Cabernet Sauvignon					
3					
Zinfandel					
Zinfandel Chablis					
Zinfandel Chablis Sauvignon Blanc					
Zinfandel Chablis Gauvignon Blanc Geverztraminer					
Zinfandel Chablis Gauvignon Blanc Geverztraminer /alpolicella					
Zinfandel Chablis Sauvignon Blanc Geverztraminer Valpolicella Pinot Noir					

To run the Auto demo, set the directory to the CLIPSJNI/examples/AutoDemo directory and enter the following command:

java -cp .:../../CLIPSJNI.jar -Djava.library.path=../.. AutoDemo

The Auto Demo window should appear:



To run the Animal demo, set the directory to the CLIPSJNI/examples/AnimalDemo directory and enter the following command:

java -cp .:../../CLIPSJNI.jar -Djava.library.path=../.. AnimalDemo

The Animal Demo window should appear:

	Animal Demo
Welcome to	the Animal Identification Expert System.
	Next
	Next

#### 10.3.2 Running the Demo Programs on Windows 7

Launch the Command Prompt application (select Start > All Programs > Accessories > Command Prompt). Set the directory to the CLIPSJNI/examples/SudokuDemo directory (using the cd command). To run the Sudoku demo, enter the following command:

java -cp .;../../CLIPSJNI.jar -Djava.library.path=../.. SudokuDemo

The Sudoku Demo window should appear:

🛃 Sudoku Demo 🗖 🗖 💌 🗸								
┝								
⊢								Clear
⊢								Reset
⊢								Solve
⊢								Techniques
⊢								
$\vdash$								
Select cell and enter digit 1-9 or press backspace/delete.								

To run the Wine demo, set the directory to the CLIPSJNI/examples/WineDemo directory and enter the following command:

java -cp .;../../CLIPSJNI.jar -Djava.library.path=../.. WineDemo

The Wine Demo window should appear:

🛃 Wine Demo						
Prefe	rences		Meal			
Color:	Color: Don't Care 💌		Main Course:	Don't Know 💌		
Body:	Don't Care	-	Sauce:	Don't Know 💌		
Sweetness:	Don't Care	-	Flavor:	Don't Know 🔻		
Wine			Recomm	Recommendation Weight		
Chardonnay						
Riesling						
Soave						
Chenin Blanc						
Gamay						
Cabernet Sauvignon						
Zinfandel						
Chablis						
Sauvignon Blanc						
Geverztraminer						
Valpolicella						
Pinot Noir						
Burgundy						

To run the Auto demo, set the directory to the CLIPSJNI/examples/WineDemo directory and enter the following command:

java -cp .;../../CLIPSJNI.jar -Djava.library.path=../.. AutoDemo

The Auto Demo window should appear:

🛃 Auto Demo	
Welcome to the Engine Diag	nosis Expert System.
Next	

To run the Animal demo, set the directory to the CLIPSJNI/examples/AnimalDemo directory and enter the following command:

java -cp .;../../CLIPSJNI.jar -Djava.library.path=../.. AnimalDemo

The Animal Demo window should appear:



## **10.4 Creating the CLIPSJNI JAR File**

If you wish to add new functionality to the CLIPSJNI package, such as new Java methods which may call existing or new native functions, it is necessary to recreate the CLIPSJNI jar file. The CLIPSJNI distribution already contains the precompiled CLIPSJNI jar file in the top level CLIPSJNI directory, so if you are not adding new functionality to the CLIPSJNI package, you do not need to recreate the jar file (unless you want to create a jar file using a Java version prior to version 1.8.0).

To create the jar file, first open a terminal window where you can enter Java tool commands. On Mac OS X, launch the Terminal application (located in the Applications/Utilities directory). On Windows 7, launch the Command Prompt application (select Start > All Programs > Accessories > Command Prompt).

Using the appropriate commands (cd on Mac OS X and Windows 7), set the current directory to CLIPSJNI/java-src , then enter the following command to compile the CLIPSJNI java source:

javac CLIPSJNI/\*.java

Once compiled, enter the following command to place the class files in a jar file:

jar -cf CLIPSJNI.jar CLIPSJNI/\*.class

Once the CLIPSJNI.jar file is created, move it from the CLIPSJNI/java-src directory to the top level CLIPSJNI directory.

If you are adding new native functions to the CLIPSJNI package, it is also necessary to create the JNI header file which will be used to compile the native library. While you are still in the CLIPSJNI/java-src directory, enter the following command:

javah -jni CLIPSJNI.Environment

This command creates a file named CLIPSJNI\_Environment.h which must be moved from the CLIPSJNI/java-src directory to the CLIPSJNI/library-src directory.

## **10.5 Creating the CLIPSJNI Native Library**

The CLIPSJNI distribution already contains a precompiled universal library for Mac OS X, libCLIPSJNI.jnilib, and for Windows, CLIPSJNI.dll, in the top level CLIPSJNI directory. It is necessary to create a native library only if you are using the CLIPSJNI with an operating system other than Mac OS X or Windows. You must also create the native library if you want to add new functionality to the CLIPSJNI package by adding additional native functions. The steps for creating a native library varies between operating systems, so some research may be necessary to determine how to create one for your operating system.

#### 10.5.1 Creating the Native Library on Mac OS X

Launch the Terminal application (located in the Applications/Utilities directory). Set the directory to the CLIPSJNI/lbrary-src directory (using the cd command).

To create a universal native library that can run on both Intel 32 and 64 bit architectures, enter the following command:

make -f makefile.mac

Once you have create the native library, copy the libCLIPSJNI.jnilib file from the CLIPSJNI/library-src to the top level CLIPSJNI directory.

#### 10.5.2 Creating the Native Library on Windows 7

The following steps assume you have Microsoft Visual Studio 2013 installed. First, launch the Command Prompt application (select Start > All Programs > Accessories > Command Prompt).

Next, execute the script that sets up the environment variables for the appropriate target machine. For example, the vcvars64.bat batch file in the directory "Program Files (x86)/Microsoft Visual Studio 12.0/VC/bin/amd64".

Set the directory to the CLIPSJNI/lbrary-src directory (using the cd command).

To create the native library DLL, enter the following command:

nmake -f makefile.win

Once you have create the native library, copy the CLIPSJNI.dll file from the CLIPSJNI/librarysrc to the top level CLIPSJNI directory.

#### 10.5.3 Creating the Native Library On Other Systems

The file makefile.linux is intended to generate a native library for Linux systems using Java version 1.8.0. It can be invoked using the following command:

make -f makefile.linux

The shared library generated by this makefile is libCLIPSJNI.so. You will likely need to change the directory paths in the makefile to the appropriate location for your Java installation.

#### 10.6 Recompiling the Swing Demo Programs

If you want to make modification to the Swing Demo programs and recompile them, you can use one of the following commands to do so (assuming you are in the appropriate directory for the example and the CLIPSJNI.jar file is present in the top level CLIPSJNI directory):

javac -classpath ../../CLIPSJNI.jar SudokuDemo.java javac -classpath ../../CLIPSJNI.jar WineDemo.java javac -classpath ../../CLIPSJNI.jar AutoDemo.java javac -classpath ../../CLIPSJNI.jar AnimalDemo.java

### 10.7 Internationalizing the Swing Demo Programs

The Swing Demo Programs have been designed for internationalization. Several software generated example translations have been provided including Japanese (language code ja), Russian (language code ru), Spanish (language code es), and Arabic (language code ar). To make use of one of the translations, specify the language code when starting the demonstration program. For example, to run the Animal Demo in Japanese on Mac OS X, use the following command:

java -cp .:../../CLIPSJNI.jar -Djava.library.path=../.. -Duser.language=ja AnimalDemo

The welcome screen for the program should appear in Japanese rather than English:



It may be necessary to install additional fonts to view some languages. On Mac OS X, you can see which languages are supported by launching System Preferences and clicking the Language & Region icon. On Windows 7, you can see which languages are supported by launching Control Panel and selecting the Keyboards and Languages tab from Region and Language Options.

To create translations for other languages, first determine the two character language code for the target language. Make a copy in the resources directory of the ASCII English properties file for the demo program and save it as a UTF-8 encoded file including the language code in the name and using the .source extension. A list of language code is available at http://www.mathguide.de/info/tools/languagecode.html. For example, to create a Greek translation file for the Wine Demo, create the UTF-8 encoded WineResources\_el.source file from the ASCII WineResources.properties file. Note that this step requires that you to do more than just duplicate the property file and rename it. You need to use a text editor that allows you to change the encoding from ASCII to UTF-8.

Once you've created the translation source file, edit the values for the properties keys and replaced the English text following each = symbol with the appropriate translation. When you have completed the translation, use the Java native2ascii utility to create an ASCII text file from the source file. For example, to create a Greek translation for the Wine Demo program, you'd use the following command:

```
native2ascii -encoding UTF-8 WineResources_el.source WineResources_el.properties
```

Note that the properties file for languages containing non-ASCII characters will contain Unicode escape sequences and is therefore more difficult to read (assuming of course that you can read the language in the original source file). This is the reason that two files are used for creating the translation. The UTF-8 source file is encoded so that you can read and edit the translation and the ASCII properties file is encoded in the format expected for use with Java internationalization features.