aicas GmbH

JamaicaCAR SDK

Version 2.6.9dev
17 March 2016

Released under NDA

aicas

Contents

Executive Summary

This document explains the installation process and usage of the JamaicaCAR SDK and the JamaicaCAR Eclipse Plugin.

1 Using the SDK with Eclipse

1.1 Requirements

- Eclipse (version 4.4 or higher) installed.
- JamaicaCAR SDK.
- JamaicaCAR Eclipse Plugin.

1.2 Installation

1.2.1 Eclipse Plugin

The plugin can be downloaded from `https://www.aicas.com/cms/en/JamaicaCAR`. After unzipping the downloaded file, the plugin can be installed and updated through the Eclipse plugin manager.

The plugin requires Eclipse 4.4 or later and a Java 7 compatible virtual machine. However, using the latest available Eclipse version and an up-to-date virtual machine is recommended. The following instructions refer to Eclipse 4.4. Newer versions of Eclipse may differ slightly in the menu item labels.

To install the plugin, select the menu item

`Help > Install New Software...,`

add the update site from the unzipped downloaded file, and install `JamaicaCAR Tools`. The plugin is available after a restart of Eclipse.

1.2.2 JamaicaCAR SDK

Install the JamaicaCAR SDK by unziping the file available at `https://www.aicas.com/cms/en/JamaicaCAR` into the desired location.

1.3 Configuring Eclipse

To add the JamaicaCAR SDK as a JRE, start by opening

`Windows > Preferences > Java > Installed JREs.`

Add a new JRE. Choose `JamaicaCAR Emulator` as JRE Type. Set the JRE Home Directory to the root directory of your JamaicaCAR SDK installation.

1.4 Usage

1.4.1 Creating JamaicaCAR Xlets

Create a new JamaicaCAR Xlet project with

`File > New > Other... > JamaicaCAR > JamaicaCAR Xlet Project.`

Make sure the JamaicaCAR SDK is chosen as project JRE.

Create a new JamaicaCAR Xlet class. This can be done via

`File > New > Other... > JamaicaCAR > JamaicaCAR Xlet Class`

or via the shortcut in the `New Java Class` dropdown list on the toolbar. Implement the Xlet.

1.4.2 Creating JamaicaCAR Xlets with the CodenameOne Designer

`File > New > Other... > JamaicaCAR > JamaicaCAR Xlet Project with GUI Builder.`

The created Xlet project contains:

- HelloWorld.java the Xlet main class.
- StateMachineBase.java the StateMachineBase which gets generated by the designer.
- StateMachine.java the StateMachine which can be used to hook the GUI elements with user code.
- GUI.res the resource file used by the CodenameOne Designer.
- xlet.properties the Xlet properties.
- codenameone_settings.properties a properties file that is used by the CodenameOne Designer.

1.4.3 Running JamaicaCAR Xlets

To run the Xlet, go to

`Run > Run configurations`

and create a new JamaicaCAR Xlet run configuration. Set the Xlet Project and Xlet Main Class. Select the checkbox `Start as daemon xlet` if the Xlet should run in daemon mode. Then the run configuration can be started.

To set start arguments for the Xlet use the `Xlet Start Arguments` box. The arguments have to be written in well-typed JSON, e.g.: {"arg1":"value1","arg2":"value2"}. Environment variables can be set in the `Enviroment` tab.

1.4.4 Debugging JamaicaCAR Xlets

To debug the Xlet, go to

`Run > Debug configurations.`

If a JamaicaCAR Xlet run configuration for the Xlet Class was previously created, this configuration will also be listed in the debug configurations. It can be used to debug the application. Otherwise, a new JamaicaCAR Xlet debug configuration must be created and the Xlet Project and Xlet Main Class must be set. Then the debug configuration can be started. To link other projects or jars to the debug session use the `Classpath` tab.

1.5 Switching between Xlet States

Once a Xlet was started (either in default or debug mode), the emulator enables switching between the states *running*, *paused*, and *stopped*. This can be done by selecting the appropriate state in the `State` menu item.

2 Using the Emulator from the Command Line

The emulator can be started with the following command line arguments:

```
> emulator_bin [-h|-help] \
               [-version] \
               [{-cp | -classpath} XletClassPath] \
               [-fontProperties <fonts.properties>] \
               [-securityConfiguration <security.jar>] \
               [-installationDirectory <xletsDir>] \
               [-extensionDirectory <extensionDir>] \
               [-xletExtensionDirectory <xletExtensionDir>] \
               [-startArgs <startArgs>] \
               [-verbosityLevel <level>] \
               [-autoStartXlets] \
               [-disableDrag] \
               [-theme <theme.jar>] \
               [XletJarName]
```

The given command line arguments have the following effect:

- **-h** or **-help** prints usage information of the emulator and exits.
- **-version** prints the version of the emulator and exits.
- **-cp/-classpath** specifies the classpath needed by the Xlet. This is a list of JAR files that are separated by the path separator (';' on Windows, otherwise ':').
- **-fontProperties** tells the emulator which TTF fonts are available on the target platform.
  The file contains entries in the following form:
  `<font-identifier>(\ |:|=)<font-file>`,
  where `<font-identifier>` is one of
  - *fontname-style*
  - *fontname*

  and *style* is one of the four case-insensitive strings:
  "PLAIN", "BOLD","BOLDITALIC", or "ITALIC".
  Note: This feature still is under development. Changes are to be expected for future versions.
- **-securityConfiguration** enables authentication, permissions and resource budget checks w.r.t. the provided security configuration. The default is to not perform such checks.
- **-installationDirectory** selects the name of the directory in which Xlets will be installed. This directory must exist. It may contain applications installed during a previous run of the emulator.
- **-extensionDirectory** selects the name of the directory where additional JAR libraries needed by the Xlets can be stored. These JAR files are loaded once by a global ClassLoader and can be accessed by all Xlets.
- **-xletExtensionDirectory** selects the name of the directory where additional JAR libraries needed by the Xlets can be stored. These JAR files are loaded separately for each Xlet.
- **-startArgs** specifies the arguments that are passed to the Xlet when it gets started.
- **-verbosityLevel** sets the verbosity level for logging output.

- **-autoStartXlets** asks the Emulator to automatically start all installed daemon Xlets that are declared to be *autostart Xlets* in their *xlet.properties* file. The autostart Xlets are started in the background, so that commands for non-autostart Xlets can be processed concurrently. The emulator sends the usual IPC signals and alarms in order to notify the environment of the start-up progress of autostart Xlets.
- **-disableDrag** disables generation of drag events in LWUIT and CodenameOne.
- **-theme** selects the path to the JAR file containing the theme configuration. If not specified, `<Path to JamaicaCAR SDK>/lib/theme.jar` is used.
- **XletJarName** is the name of the Xlet JAR file.