




Manual

„Integration of the Zelfi Library“

for business partners



Version: 4.0, 22.08.2007

Zelfi AG
Erthalstraße 1
55118 Mainz
Telephone: +49 6131 / 9064850
Fax: +49 6131 / 9064853
eMail: business-partner@zelfi.com

Table of contents

Table of contents	2
1. Introduction	3
2. Short manual	3
3. Embedding the library	4
Eclipse	4
J2ME Polish	5
Antenna	6
Wireless Toolkit	6
4. Source code adjustments	7
MIDlet class	7
Canvas Class	7
5. The configuration file key.txt	9
6. Project transfer to Zelfi	9

1. Introduction

This manual is intended for business partners of the Zelfi AG. It shows how the Zelfi interface is embedded in a J2ME project.

Starting from the version 3.0 there are the following four variants of the library:

zelfi-lib-generic-midp1.jar	for MIDP 1.0 devices only.
zelfi-lib-generic-midp2.jar	for MIDP 2.0 devices only.
zelfi-lib-nokia-midp1.jar	for MIDP 1.0 with Nokia FullCanvas
zelfi-lib-nokia-midp2.jar	for MIDP 2.0 with Nokia FullCanvas

Qualitatively better advertisements are indicated by the use of MIDP 2.0. It is also possible to include links for WAP sites directly. This can increase the available income for the partner.

2. Short manual

The following steps are necessary:

- Download newest version of the library under <http://www.zelfi.com/en/partners/howto> (zelfi-lib-v3.2-en.zip).
- Insert the libraries of Zelfi into your project.
- In the MIDlet class:
 - o Import the class `com.zelfi.client.Advertiser`
 - o Consider the change of the constructor of the Canvas class (see below).
 - o Call `Advertiser.init(this);` in `startApp()`.
 - o Call `Advertiser.display()` if you like ad at the beginning.
 - o If MIDP 2.0 call `Advertiser.showWML(this);` in `destroyApp(boolean arg0)`.
- In the Canvas class:
 - o Import `com.zelfi.client.Advertiser`.
 - o The constructor is look like: `CanvasClass(MIDletClass midlet)`
 - o In suitable place (e.g. after completion of a level) call `Advertiser.display(this, midlet)`.
- Compile project and send it to Zelfi.

Note: A simple example project is available under http://www.zelfi.com/fileadmin/downloads/partner/example_workspace.zip.

3. Embedding the library

Eclipse

In order to include the library into the development environment Eclipse, the **Java Build Path** must be adapted.

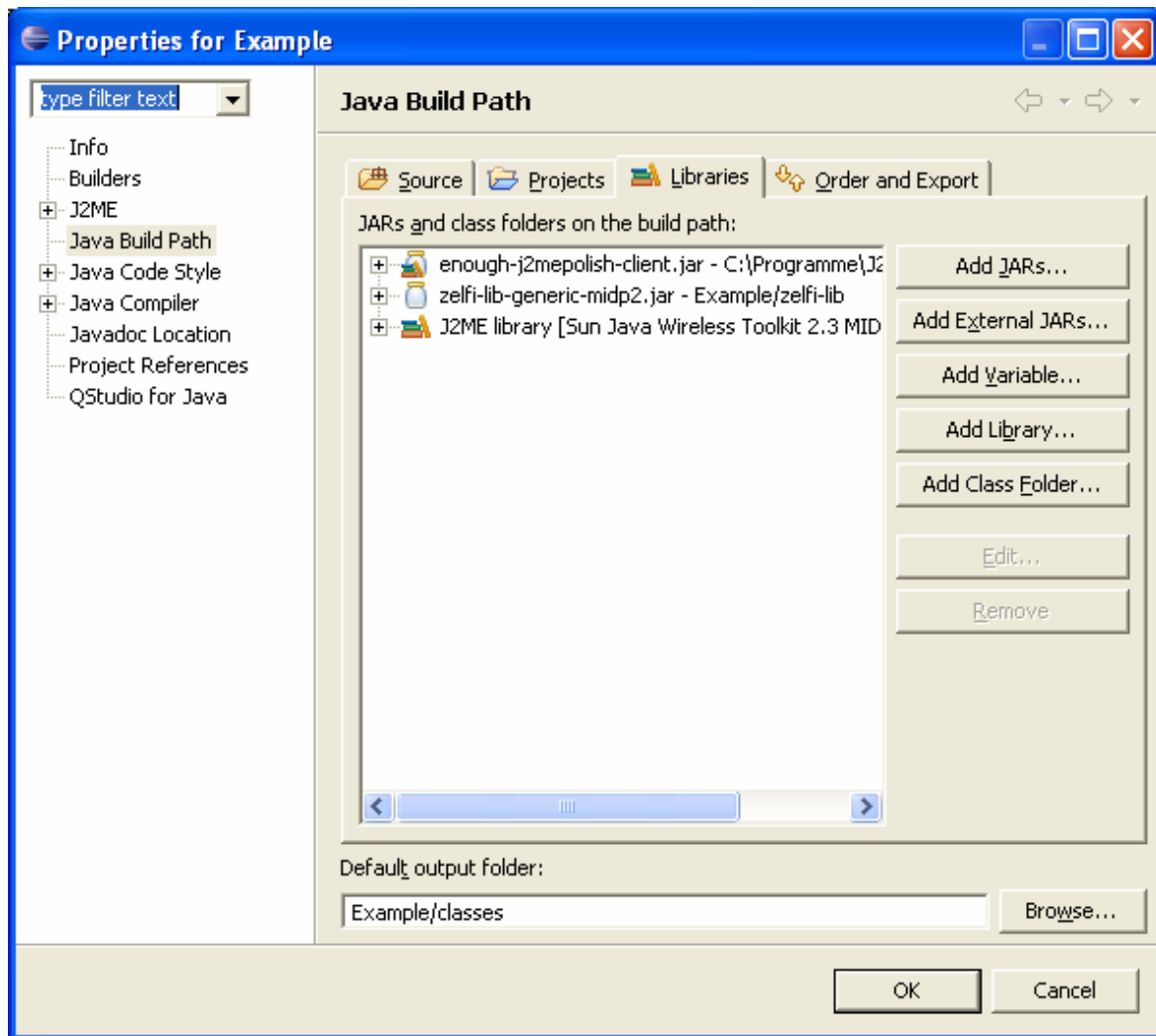


Figure 1: Set Java Build Path, step 1.

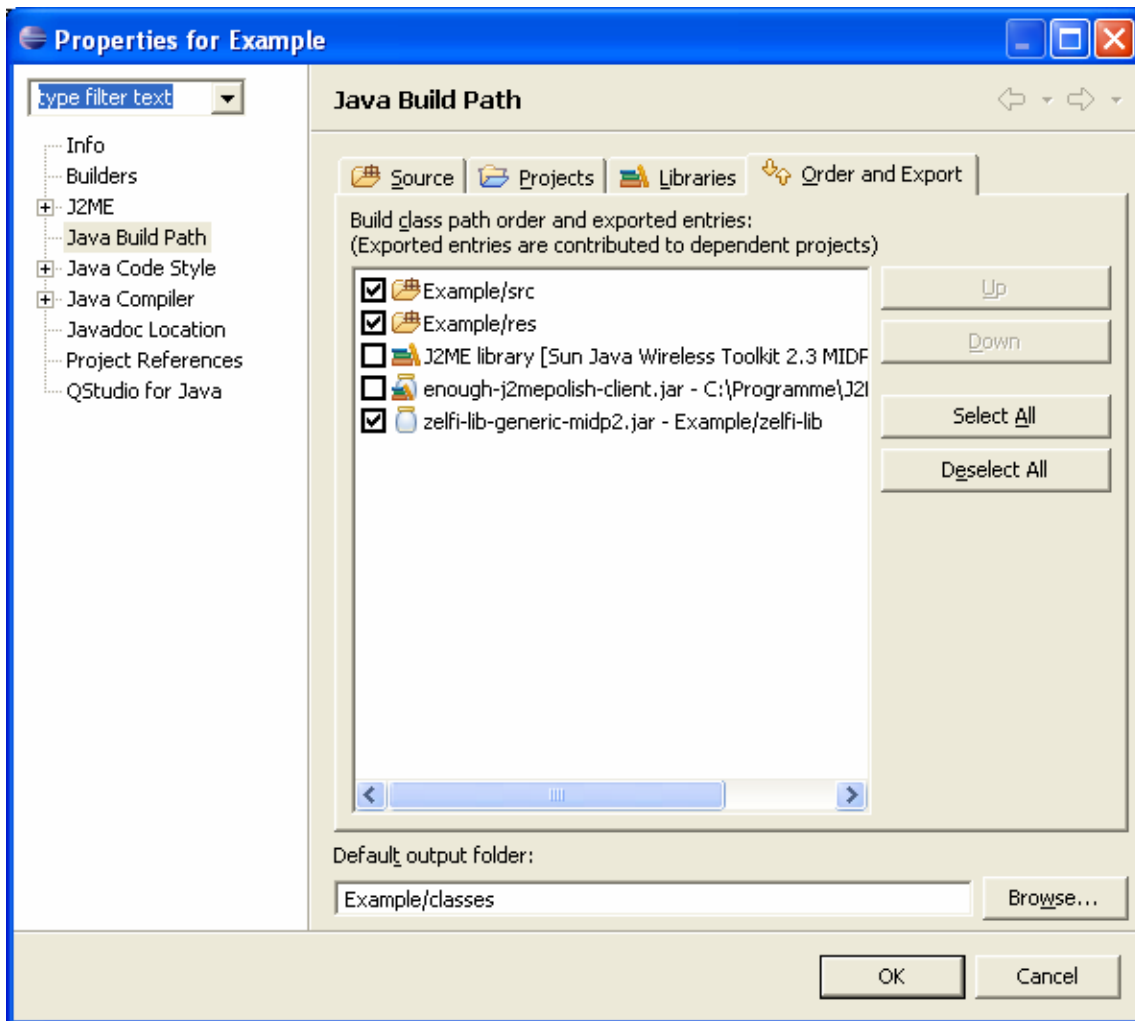


Figure 2: Set Java Build Path, Step 2.

More information about Eclipse: <http://www.eclipse.org/>

J2ME Polish

If J2ME polish is used as build tool, proceed with the following settings.

```
<property name="ZELFI-GENERIC-MIDP2.lib" value="/zelfi-lib/zelfi-lib-generic-midp2.jar"/>
```

```
<build
  usePolishGui="false"
  workDir="${dir.work}"
  binaryLibrary ="${ ZELFI-GENERIC-MIDP2.lib}"
  Symbols="polish"
>
```

More information about J2ME Polish: <http://www.j2mepolish.org/>

Antenna

If you use the build tool Antenna proceed with settings below.

```
<property name="ZELFI-GENERIC-MIDP2.lib" value=""/zelfi-
lib/zelfi-lib-generic-midp2.jar"/>

<path id="bootclasspath">
  <pathelement path="{MIDP.lib}"/>
  <pathelement path="{CLDC.lib}"/>
  <pathelement path="{ZELFI-GENERIC-MIDP2.lib}"/>
</path>

<!-- Package everything. -->
  <wtkpackage jarfile="{midlet.name}.jar" profile="MIDP-
2.0" config="CLDC-1.0" jadfile= "{midlet.name}.jad"
obfuscate="true" >
  <fileset dir="classes"/>
  <fileset dir="res"/>
  <bootclasspath path="{MIDP.lib}"/>
  <bootclasspath path="{CLDC.lib}"/>
  <libclasspath path="{ZELFI-GENERIC-MIDP2.lib}"/>
</wtkpackage>
```

More information about Antenna: <http://antenna.sourceforge.net/>

Wireless Toolkit

The J2ME Wireless Toolkit is a platform with emulator environment for developers.

Simply copy the Zelfi library into the "lib" folder of the project and the WTK includes the lib automatically.

More information about WTK: http://java.sun.com/products/sjwtoolkit/download-2_3.html

4. Source code adjustments

MIDlet class

The following class must be imported into the MIDlet class.

Import in MIDlet class:

```
import com.zelfi.client.Advertiser;
```

The static function `Advertiser.init()` which initializes the Zelfi library is called in `startApp()` of the MIDlet class.

Call of `init()`:

```
Advertiser.init(this);
```

If you wish advertisement at the start of the application directly, you can call following function in `startApp()`:

Indicates the advertisement at `startApp()`:

```
Advertiser.display(this, midlet, true, true);
```

The **MIDP 2.0** Versions of the Zelfi library allow a direct link to WAP pages. In order not to interrupt the game the consumer has the possibility to visit the WAP page later. With terminating application he will be asked once again. In the `destroyApp(boolean arg0)` the following function call must be included.

Indicate to the RMS stored WAP page with terminating application:

```
Advertiser.showWML(this);
```

Canvas Class

To have access to the functions from Advertiser class you import the Advertiser class.

Import in Canvas class:

```
import com.zelfi.client.Advertiser;
```

Constructor:

```
MIDletClass midlet;  
  
public CanvasClass(MIDletClass midlet) {  
    this.midlet=midlet;  
    ...  
}
```

The object of the MIDlet class must be stored as instance variable, to pass it later in the Zelfi interface.

Finally the class needs the function display() of the Advertiser. This function runs a Thread which waits for the signal to indicate the advertisement. If the given time is up advertisement is presented after a function call.

Define a possible position of the advert:

```
Advertiser.display(this,midlet);
```

Important: An advertising message is not indicated after every call of Advertiser.display () only if the given time interval reached. The programmer can suggest Zelfi a suitable interval of the play interruptions by the advertising messages.

In order not to impede the flow of the game, programs can give the accurate position of the call.

Games threads should be stopped, before Advertiser.display () is called.

With following function you can define in destroyApp() so that wap actions or ad will be shown if the user quit the MIDlet.

Define a wap actions or ad at the end:

```
Advertiser.shutdown(this);
```

NEW: The optional function display () has two further parameters. (*wait* and *internet*).

```
Advertiser.display(this,midlet,false,true);
```

wait

- Typ boolean (true or false)
- true = the function is terminated only if the advertisement is ready.
- false = the function is terminated directly.
- **Attention! Never use this with parameter „true“ in functions paint (), keyPressed(), keyReleased and commandAction (). Application blocks!**

Internet

- Typ boolean (true or false)
- true = advertisement from the internet.
- false = static advertisement.

- „false“ for test or debug only!
- Static advertisement can't be accounted!

5. The configuration file key.txt

The key.txt file from the JAR of the Zelfi library can be changed to specify the waiting periods between the advert breaks.

- The first line is the server address for the advertisement download. This line should not be changed.
- The second line rules the waiting period up to the first advert break. A waiting period between 0 ms and 60000 ms can be selected.
- The third line describes the waiting period for further advert breaks and can be set between 0 ms and 300000 ms.

```
http://gtdcjzt.zelfi.com/04/pa?id=v2hy07n5&game=lib_4 //Server address (do not change)
60000 //waiting period, up to the first advert break. (60000 = 1 min)
300000 //waiting period, for further advert breaks. (300000 = 5 min)
```

Contents of the file key.txt: Only fat marked values may be changed.

6. Project transfer to Zelfi

Transfer to Zelfi:

- JAR-File(s)
- JAD-File(s)
- Short description of the game, screenshots, list of the tested mobile phones.
- Description of the position, in which `Advertiser.display` is called.
- Recommendation for waiting periods.

Contact: business-partner@zelfi.com