

RADII settings, paths, ... (preliminary version)

JBuilder7 paths

The paths to required libraries can be set in menu Project->Project Properties->Required Libraries. Two general groups of packages must be set, JavaCard package (for compilation of JavaCard applet) and Gemplus card simulator (for real-time debugging). To change packages in Required Libraries tab highlight packages group (e.g. GemXpresso – Crypto) and click on Edit. Then select proper package group in left side of window and add required packages in the right side as shown on figure 1.

1. JavaCard package (usually named as GemXpresso – Crypto):

- **cryptix-jce-api.jar** - usually positioned in \$RADIII_installDir\$\lib\ cryptix-jce-api.jar
- **cryptix-gemxpresso.jar** - usually positioned in \$RADIII_installDir\$\lib\cryptix-gemxpresso

2. Gemplus simulator (GSE). Depends on type of card, which you like to simulate. All simulators packages are usually positioned in \$RADIII_installDir\$\lib\gse\. Link only such single package, that corresponds to card you like to simulate. .

- e.g. if you like to simulate GXPPro-R3 card, then link package **gse_gxppro-r3.jar**
- e.g. if you like to simulate GXPLite-Generic card, then link package **gse_gxplite-generic.jar**

Note, that **GXPLite-Generic simulator do NOT support SHA-1 and MD5** hash functions. If you like to debug applet with these functions, use simulator of GXPPro-R3 card.

JBuilder7 debugging

See figure 5 for overview.

1. Place breakpoint withing JBuilder IDE (red bullet on side of source code line).
2. Run debugg session using Run->Debug project – simulator is automatically started for you. Type of simulated card is determined by gse_XXX.jar package linked in step 2. In JBuilder logging window, you should see something like (GXPPRO-R3 card is simulated here):

```
Processing keys diversification for default key set...
GSE is starting the simulation of the "GXPPRO-R3" card
Simulator is listening on port 5000...
Server is running...
Socket (3653) locked by host: 127.0.0.1
< Card Reset >
```

3. Switch to JCardManager, upload and install applet to Terminal:"Simulator", Card:"simulated_card_type" (GXPPro-R3 in our case). When line of code with breakpoint is executed, debugger is triggered.
4. **Do not forget to close simulated card before running another debugging session** using big red square button in simulator window (in JBuilder IDE).

RADIII deployment

For setup working applet on real or simulated smart card, you have to accomplish following steps. Steps 3-7 can be automatized using RADIII Deployment.

1. Create, compile and convert JavaCard applet (described in PV181: Training 5 – Java Card technology)
2. Run JCardManager, insert card or run simulator, select proper reader Terminal and card type.
3. Authenticate against card (AUTHENTICATE command). Target file points to file that contains diversified keys that are used during authentication. These files are usually positioned in \$RADIII_installDir\$\resources\targets\
 - a. If you are using simulator, then select proper file (e.g. GXPPro-R3.properties for GXPPro-R3) or just check “Set default card property file”.
 - b. If you are using real card, than proper file should be GXPPro-R3 E32PK.properties for GXP E32PK Pro-R3 r.3.2 (ATR= 3B 7D 94 00 00 80 31 80 65 B0 83 01 02 90 83 00 90 00) or **GXP_E64PK_MUNI.properties** for GXP E64 PK (ATR= 3B 7E 94 00 00 80 25 A0 00 00 00 28 56 80 10 21 00 01 08). “Set default card property file” must be unchecked!
4. Delete older version of applet, if installed (applet with same AID). At first, delete applet instance, then applet package.
5. Upload new applet to card. Check “Automatic selection for the extension...” that automatically select *.sap version of applet for simulator or *.jar version of applet for real card. Select or write manually path to applet files. These files are typically located in
\$Applet_directory\$\oncard\card_name\...\javacard\applet_name\applet_name.sap.
E.g. E:\Develop\LabakApplet\oncard\GXPLite-Generic\Labak\javacard\Labak
6. Install applet. Set Package, Applet and Instance AID as set in JBuilder7 javacard project. You can access and modify these values using JBuilder7->GexXpresso RAD->Edit GemXpresso Project. Package AID and Applet AID must be different (e.g. applet AID = package AID concatenated with ‘01’), applet AID and instance AID can be same.
7. Select applet by its applet AID.

Clever way how to automatize whole process is to use RADIII deployment that works as batch for commands described in steps 3-7 and can be repeatedly executed. **Each card type has its own tab in Deployment!** Based on card type selected in JCardManager main window commands set is executed. See figure 3, 4 for overview.

These sets can be different, however **most often they differ only in:**

- Authenticate command – proper *.properties file must be selected. Check “Set default card property file” if using simulated card.
- Upload file into a card command – proper directory for given card type must be selected (subdirectory of \$Applet_directory\$\oncard\ directory).

New deployment can be created by Deployment ->New from JCardManager. Add all available card types.

Running deployment

If you have selected proper terminal, card type and deployment is loaded using Deployment->Open or Reopen command, then press **Deployment->Deploy**. Steps 3 – 7 are then sequentially executed and results is displayed in logging window. **Every command should end with “<- 90 00” status** (except Delete command, if no applet was previously installed).

If Select fail, then find last command, that ends sucesfully (90 00) and check settings for subsequent command. Repeat this process, untill Select is OK. If you are using debbuger connected to simulated card, you can actually debug the Install command (constructor of applet is executed).

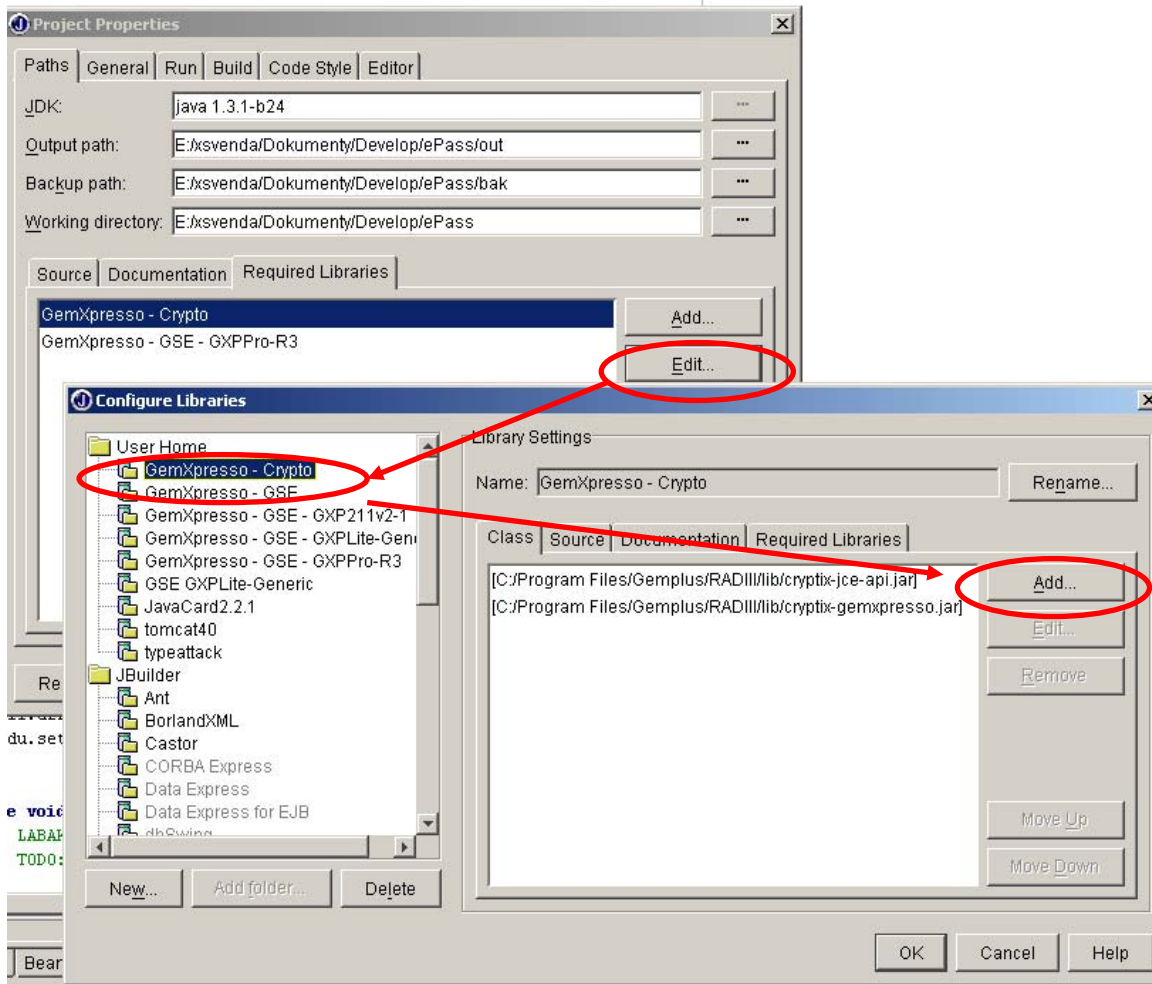


Figure 1. JBuilder libraries

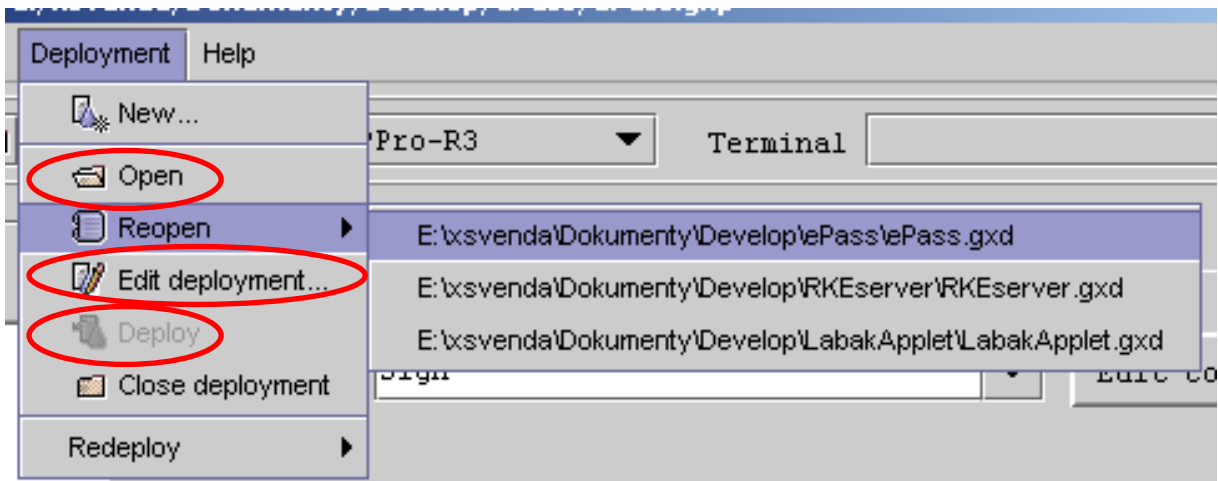


Figure 2 Open deployment

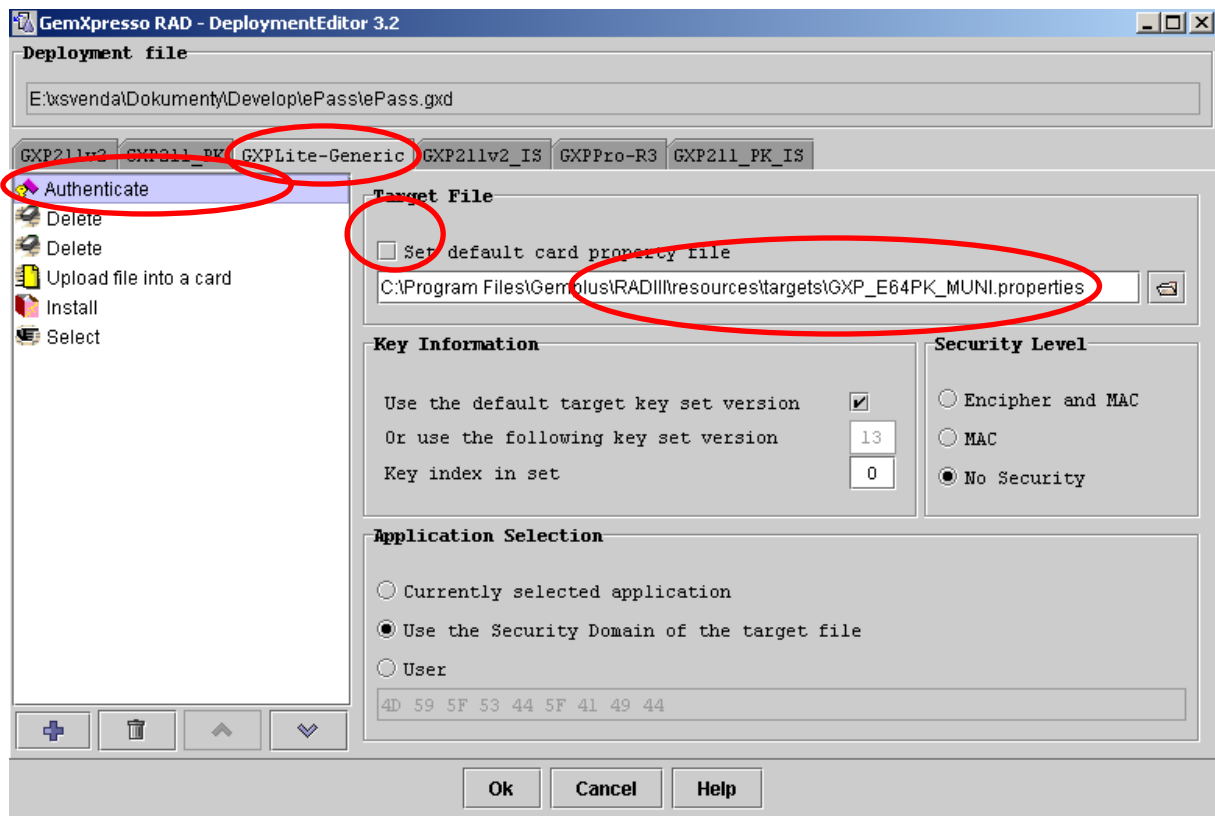


Figure 3 Deployment – Authenticate against real card GXP E64 PK

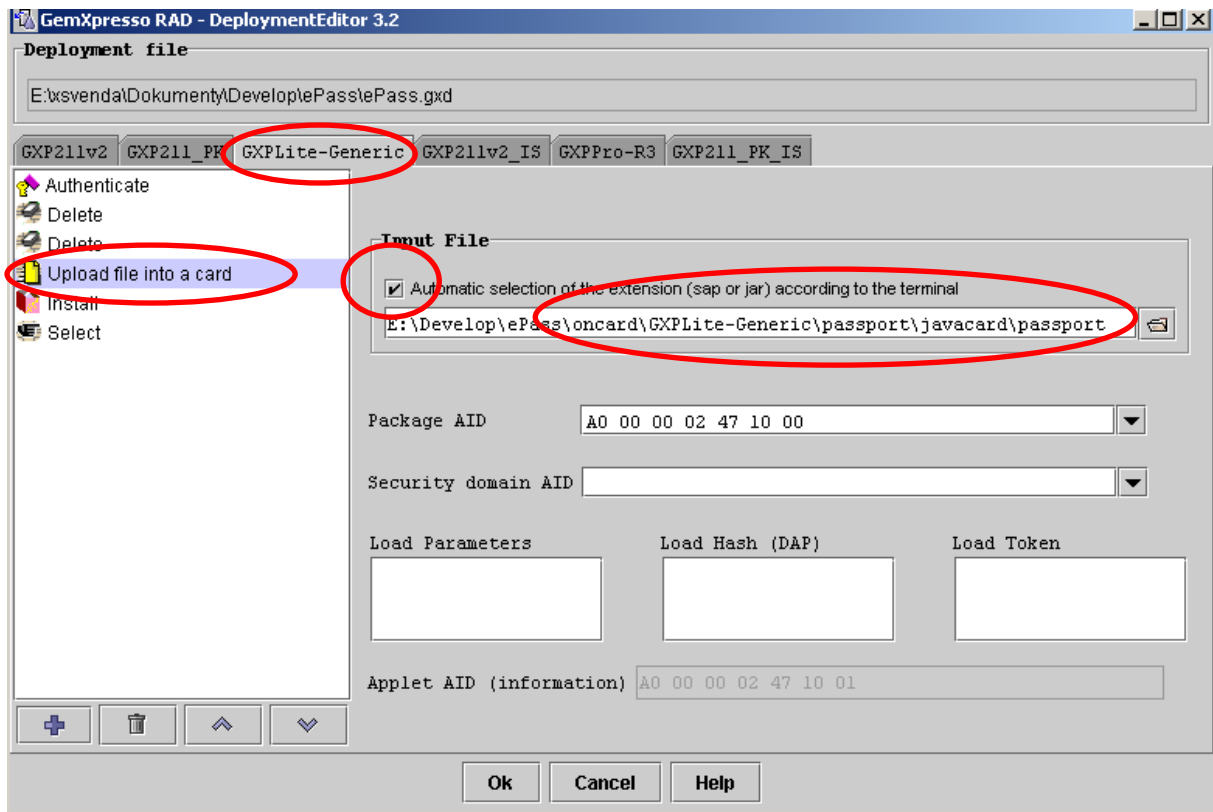


Figure 4 Deployment - Upload file into a card

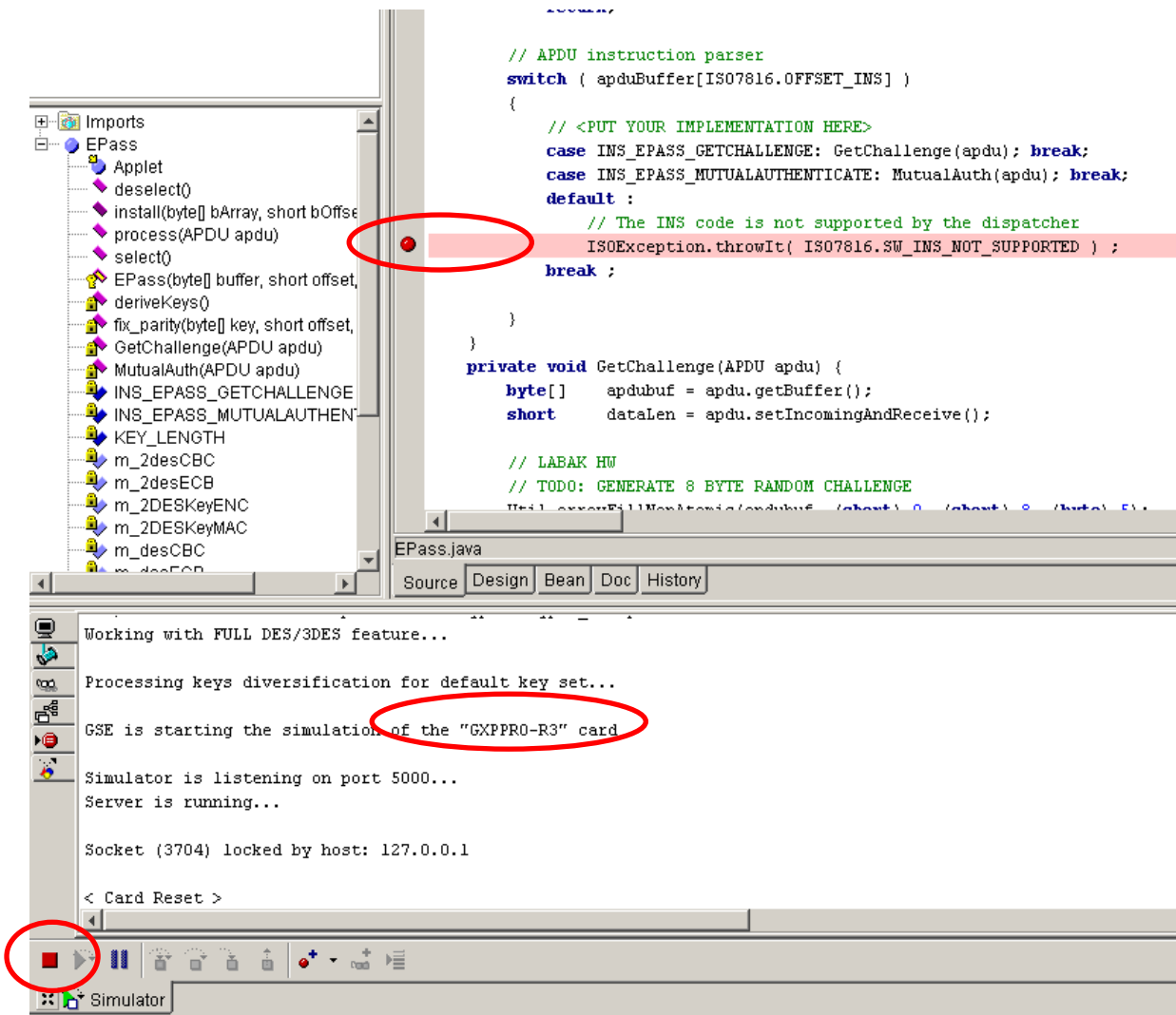


Figure 5 Debug session