

Exemple d'utilisation de IzyNFC

Guillaume Bouffard guillaume.bouffard@xlim.fr

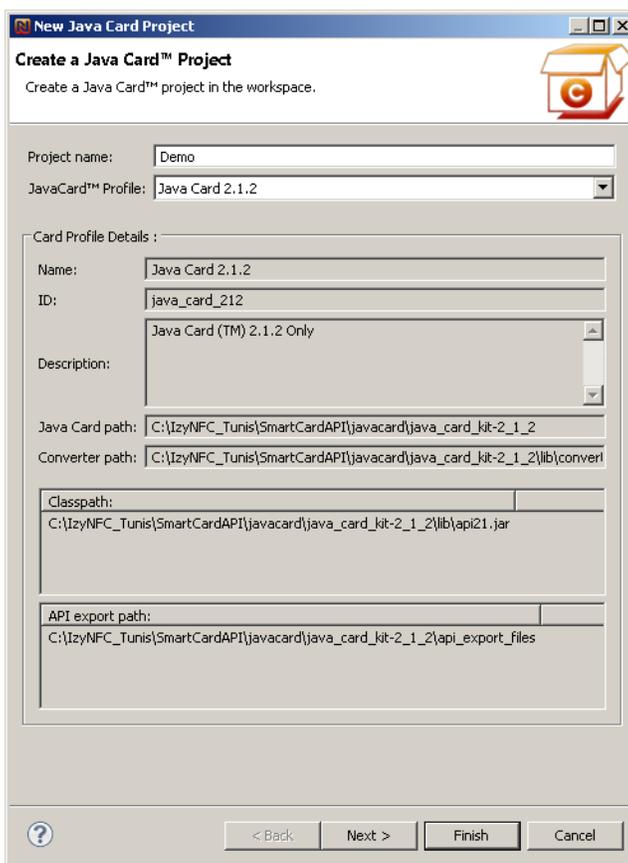
Avant-propos

Ce document explique les étapes nécessaires à l'utilisation d'IzyNFC, en partant de la création d'un projet à l'utilisation de l'application développée au travers de simulateur Java Card CREF fournie par Oracle. Les captures d'écran et les indications sont basées sur la version 20130725-1.1.0 en anglais de IzyNFC.

Notre première application Java Card

Création d'un Projet Java Card

Pour créer un projet, il suffit de cliquer sur `File > New > Java Card™ Project`. Dans cet



exemple, nommé `Demo`, nous allons utiliser le profile Java Card 2.1.2¹.

Une fois les informations renseignées, il suffit de cliquer sur `Finish`.*

¹ Ce profile correspond aux cartes en notre possession.

Création de notre première application

Une fois notre projet créé, il ne reste plus qu'à créer notre application. Pour cela, il suffit d'aller dans le menu `File > New > Java Card™ Applet`. Pour notre première application, nous allons utiliser les paramètres suivants :

The screenshot shows the 'New Java Card Applet' dialog box with the following configuration:

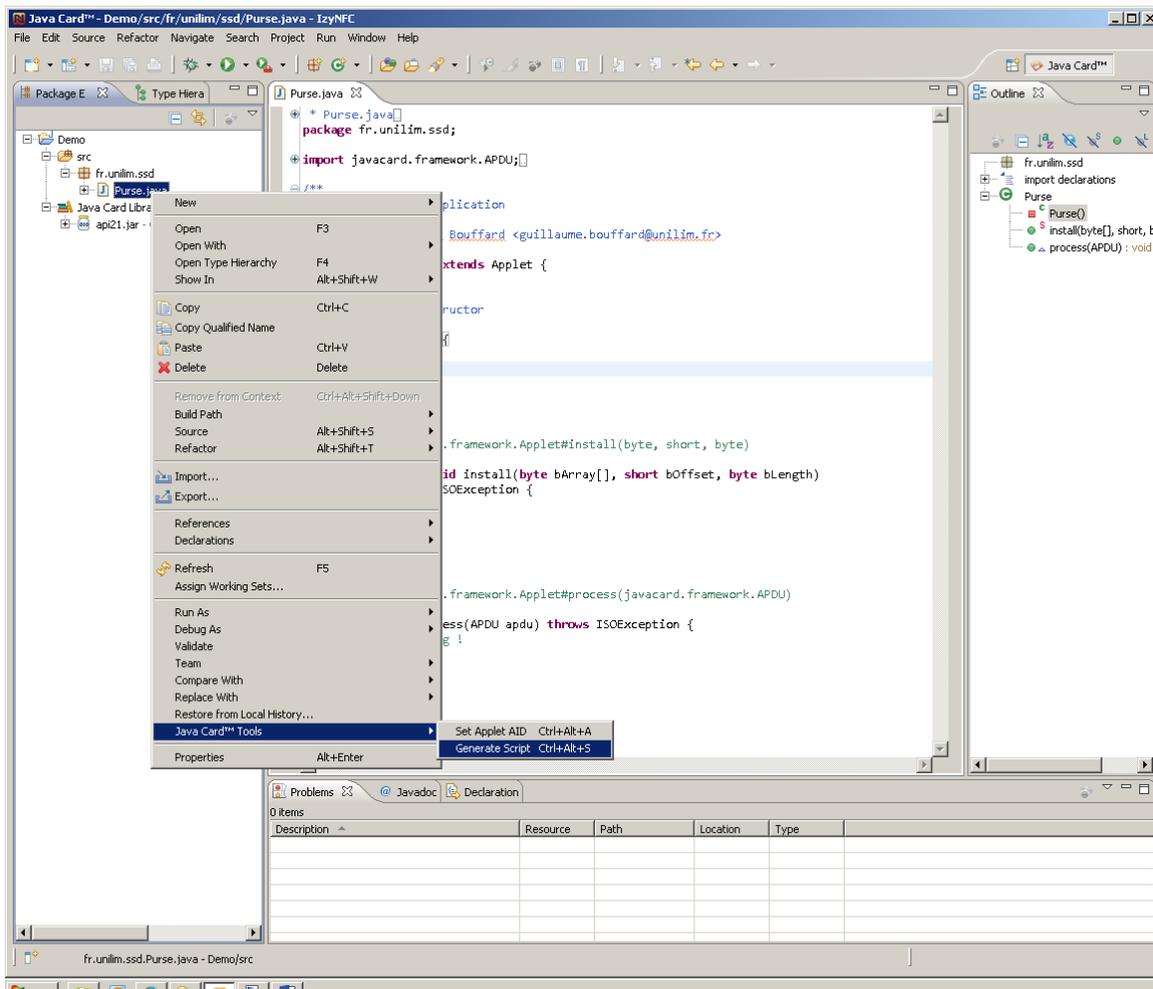
- Source folder: Demo/src
- Package AID: 66 72 2e 75 6e 69 6c 69 6d 2e 73 73 64
- Package: fr.unilim.ssd
- Enclosing type: (empty)
- Applet AID: 66 72 2e 75 6e 69 6c 69 6d 2e 73 73 64 41 70 70
- Name: Purse
- Modifiers: public (selected), default, private, protected, abstract, final, static
- Superclass: javacard.framework.Applet
- Interfaces: (empty)
- Do you want to add comments? (Generate comments checked)

Dans notre exemple, le package AID correspond à `fr.unilim.ssd` et l'applet AID à `fr.unilim.ssdApp`. Attention, il faut absolument que le package AID soit inclus dans l'applet AID. Il est aussi à noter que les AID précisés ici seront utilisé pour l'installation de l'applet. Il faut bien évidemment que les **AID soit scrupuleusement les mêmes**. En cliquant sur `Next`, il est proposé d'ajouter à notre applet le support RMI et des interfaces de partages (`Shareable Interface`). Dans cette explication, nous n'en aurons pas besoin. Il suffit donc de cliquer sur `Finish`.

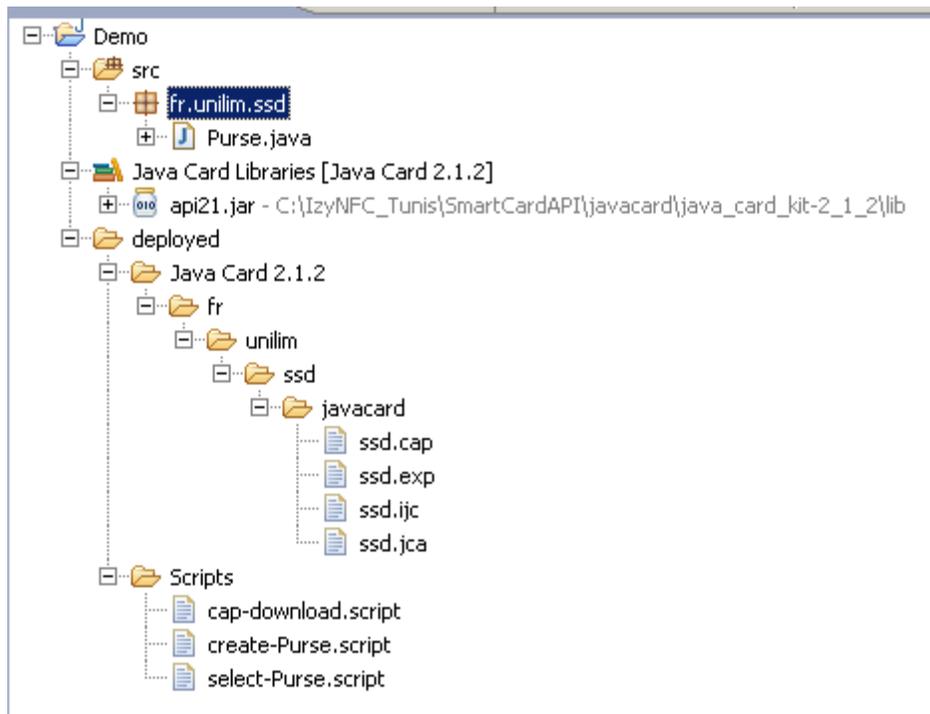
Compilation de notre application

Une fois que nous avons développé notre application, il est temps de la compiler pour l'installer.

- clic droit sur le package et dans la rubrique `Java Card Tools` prendre `Convert`,
- puis recommencer en faisant `Generate Script`.



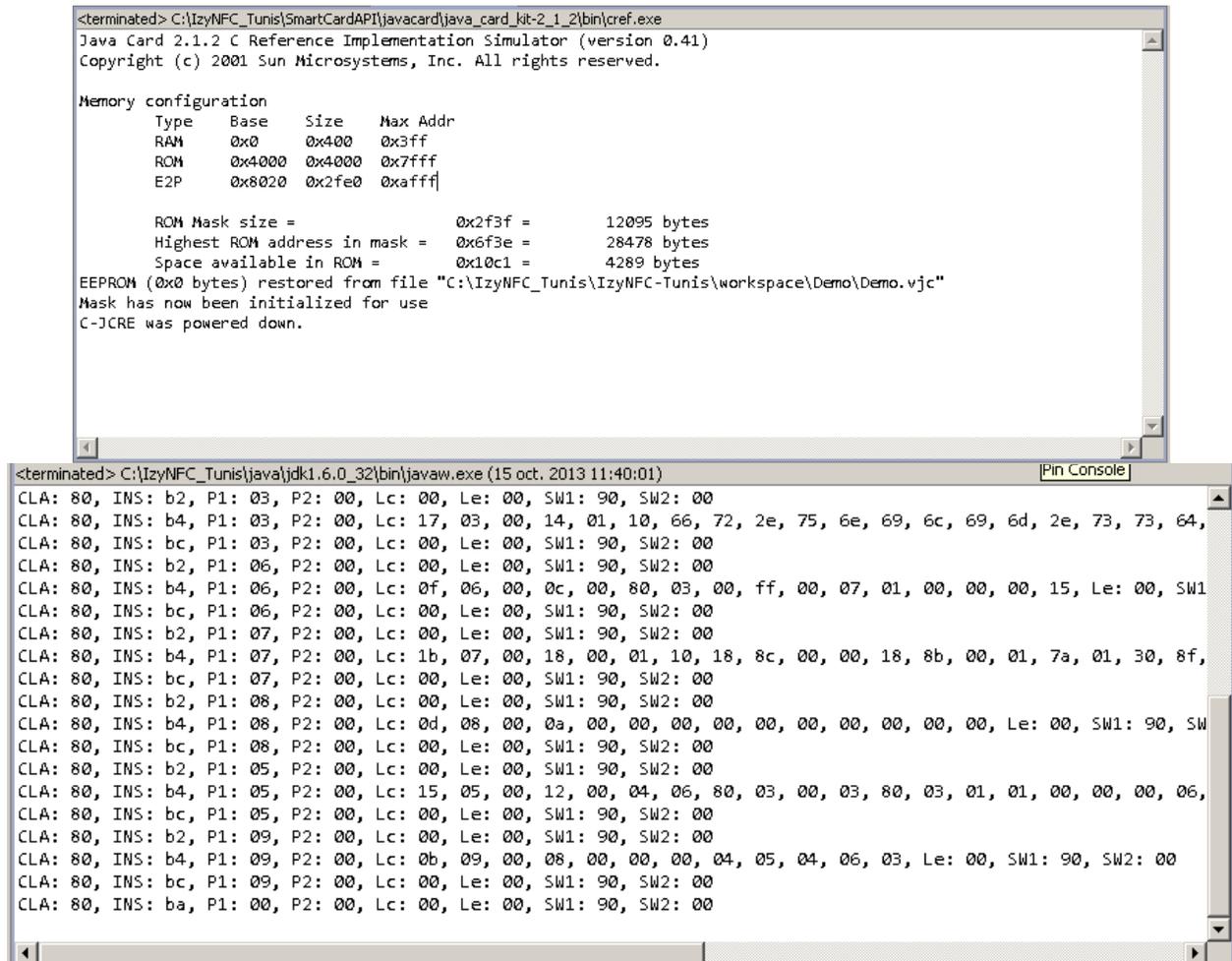
Une fois généré, nous obtenons les informations suivantes :



Il ne reste plus qu'à lancer le simulateur. Pour utiliser votre applet :

- Lancer le simulateur (à chaque fois). Pour cela, clic droit sur le Projet (ici Demo) > Run As > Emulated CREF.
- Clic droit sur le script Scripts/cap-download.script et dans la rubrique Run As choisir Launch JavaCard Script
- faire la même manip avec le script Scripts/create-Purse.script
- et refaire encore la même chose avec le script Scripts/select-Purse.script.

Pour chaque script, pour voir le retour des APDU envoyé, il suffit de cliquer sur . On obtient alors les vues suivantes.



```
<terminated> C:\IzyNFC_Tunis\SmartCardAPI\javacard\java_card_kit-2_1_2\bin\cref.exe
Java Card 2.1.2 C Reference Implementation Simulator (version 0.41)
Copyright (c) 2001 Sun Microsystems, Inc. All rights reserved.

Memory configuration
Type      Base      Size      Max Addr
RAM       0x0       0x400    0x3ff
ROM       0x4000    0x4000    0x7fff
E2P       0x8020    0x2fe0    0xffff

ROM Mask size =          0x2f3f =          12095 bytes
Highest ROM address in mask = 0x6f3e =          28478 bytes
Space available in ROM =  0x10c1 =           4289 bytes
EEPROM (0x0 bytes) restored from file "C:\IzyNFC_Tunis\IzyNFC-Tunis\workspace\Demo\Demo.vjc"
Mask has now been initialized for use
C-JCRE was powered down.

<terminated> C:\IzyNFC_Tunis\java\jdk1.6.0_32\bin\javaw.exe (15 oct. 2013 11:40:01)
CLA: 80, INS: b2, P1: 03, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b4, P1: 03, P2: 00, Lc: 17, 03, 00, 14, 01, 10, 66, 72, 2e, 75, 6e, 69, 6c, 69, 6d, 2e, 73, 73, 64,
CLA: 80, INS: bc, P1: 03, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b2, P1: 06, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b4, P1: 06, P2: 00, Lc: 0f, 06, 00, 0c, 00, 80, 03, 00, ff, 00, 07, 01, 00, 00, 00, 15, Le: 00, SW1
CLA: 80, INS: bc, P1: 06, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b2, P1: 07, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b4, P1: 07, P2: 00, Lc: 1b, 07, 00, 18, 00, 01, 10, 18, 8c, 00, 00, 18, 8b, 00, 01, 7a, 01, 30, 8f,
CLA: 80, INS: bc, P1: 07, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b2, P1: 08, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b4, P1: 08, P2: 00, Lc: 0d, 08, 00, 0a, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, Le: 00, SW1: 90, SW
CLA: 80, INS: bc, P1: 08, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b2, P1: 05, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b4, P1: 05, P2: 00, Lc: 15, 05, 00, 12, 00, 04, 06, 80, 03, 00, 03, 80, 03, 01, 01, 00, 00, 00, 06,
CLA: 80, INS: bc, P1: 05, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b2, P1: 09, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b4, P1: 09, P2: 00, Lc: 0b, 09, 00, 08, 00, 00, 00, 04, 05, 04, 06, 03, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: bc, P1: 09, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: ba, P1: 00, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
```

Problèmes Rencontrés

En plus des problèmes classiques des entiers non supportés par le simulateur, il peut advenir d'autres problèmes.

Constructeur non fonctionnel

Suivant la configuration de l'environnement de développement, il arrive que le constructeur simple :

```
private Purse() {
    register();
}
```

Doit être remplacé par un constructeur utilisant des commandes GlobalPlatform :

```
private Purse (byte buffer[], short offset, byte length) {
    // data offset is used for application specific parameter.
    // initialization with default offset (AID offset).
    short dataOffset = offset;
    boolean isOP2 = false;

    if (length >= 9) {
        // shift to privilege offset
        dataOffset += (short) (1 + buffer[offset]);
        // finally shift to Application specific offset
        dataOffset += (short) (1 + buffer[dataOffset]);
        // go to proprietary data
        dataOffset++;

        // update flag
        isOP2 = true;
    } else {
        // Install parameter compliant with OP 2.0.
    }

    // register this instance
    if (isOP2) {
        register(buffer, (short) (offset + 1), (byte) buffer[offset]);
    } else {
        register();
    }
}
```

}

Ce problème apparait si le chargement d'un applet est refusé au travers du script `Scripts/cap-download.script`

Utilisation de l'applet impossible

Il est advenu, en TP, que des étudiants ne pouvaient pas utiliser leur applet, pourtant correctement installé. Cela est dû à une réaction étrange du CREF. Pour pallier à ce problème, avant le `powerdown`, dans le script `Scripts/create-Purse.script`, il faut rajouter, au moins, la commande de sélection de l'applet créé. Cette commande est disponible dans le fichier `Scripts/select-Purse.script`.

Il peut être aussi rajouté, à la suite de la commande `select`, les diverses commandes applicatives attendus par l'applet.