

le **cnam**

Développement d'une application Java Card 2.x sous Windows

khaled.garri@gmail.com
samia.bouzefrane@cnam.fr

Laboratoire CEDRIC
Chaire Systèmes Embarqués et Enfouis
CNAM

I. Outils de développement

I.1. Outils matériels

1. Une carte à puce Java Card.
2. Un lecteur de carte à puce.

I.1. Outils logiciels nécessaires

- a) Télécharger et installer le JDK le plus récent s'il n'est pas déjà installé sur votre machine (utiliser le lien suivant pour le télécharger : (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)).
- b) Télécharger et installer Eclipse pour Java sous Windows (<https://www.eclipse.org/downloads/>).
- c) Télécharger le projet Eclipse JSR268TK.zip dans un répertoire local par exemple C:\JavaCard, à partir du lien suivant : <http://cedric.cnam.fr/~bouzefra/cours/JSR268TK.zip>
Ce projet intègre des outils indispensables pour développer et installer un programme sur une carte.
- d) Depuis Windows 7, il n'est plus nécessaire d'installer le driver du lecteur de cartes à puce. Le lecteur est détecté automatiquement dès sa connexion sur un port USB. Ce qui n'est pas le cas sous Linux où l'installation du driver doit être faite explicitement.

II. Etapes de développement

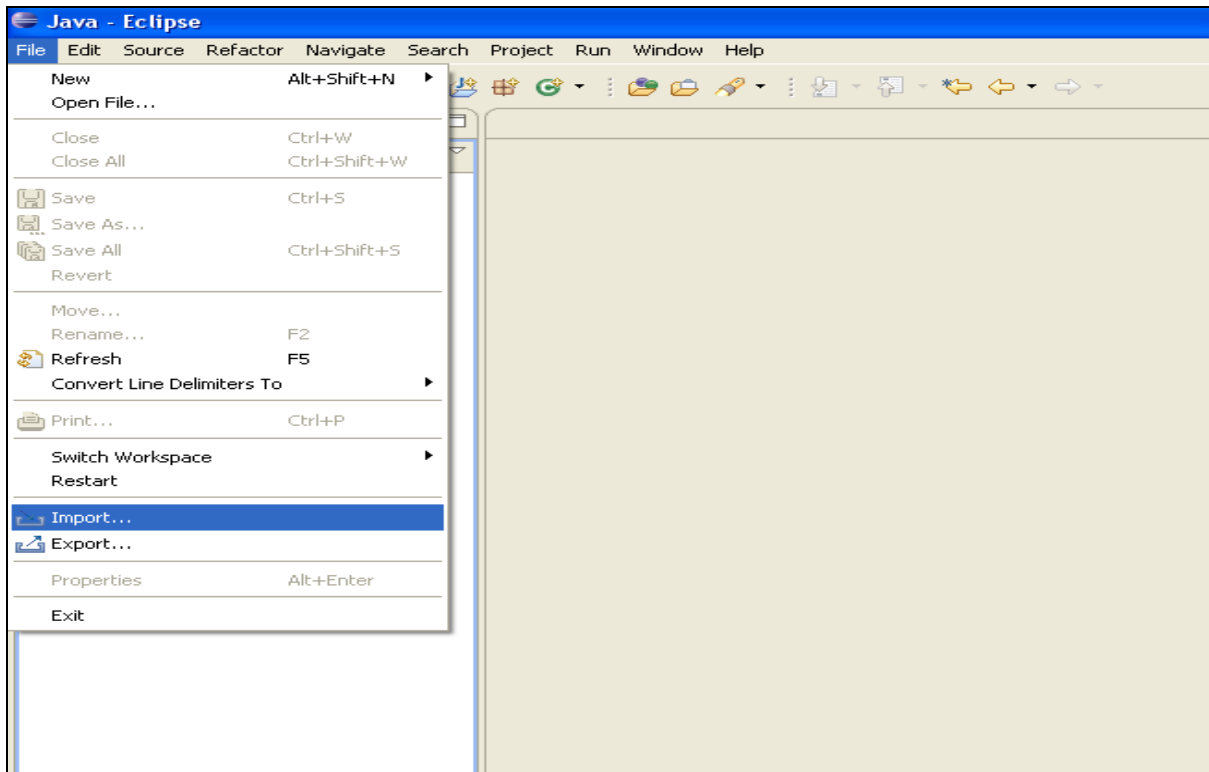
II.1 importer l'outil de développement sous eclipse :

- a) Décompresser le projet c:\JavaCard\JSR268TK.zip dans le même répertoire.
- b) Lancer Eclipse.
- c) Importer le projet java C:\JavaCard\JSR268TK\JSR268TK\JSR268TK-2 en suivant le cheminement suivant en partant du menu *File* :

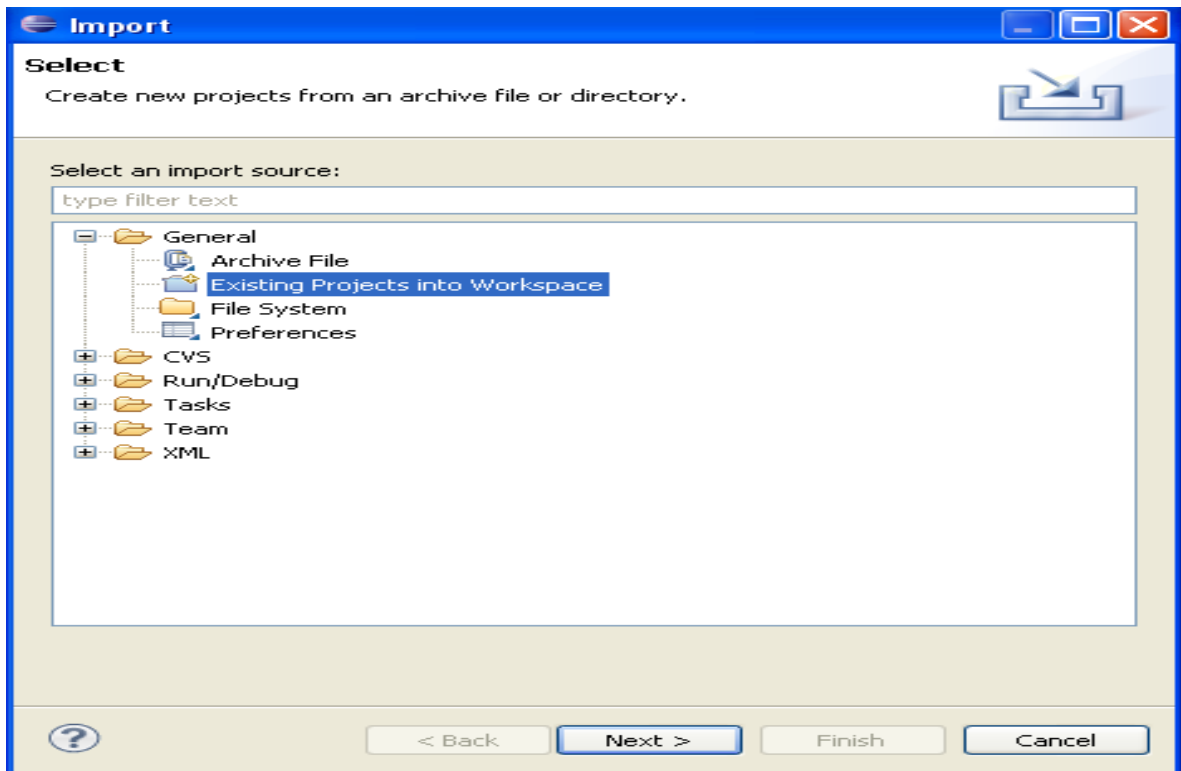
File → Import → General → Existing Project into Workspace → Browse

Chercher l'emplacement où vous avez décompressé le fichier JSR268TK.zip. Dans notre cas c'est : C:\JavaCard\JSR268TK\JSR268TK-2. Sélectionner le répertoire JSR268TK-2 et puis faire OK.

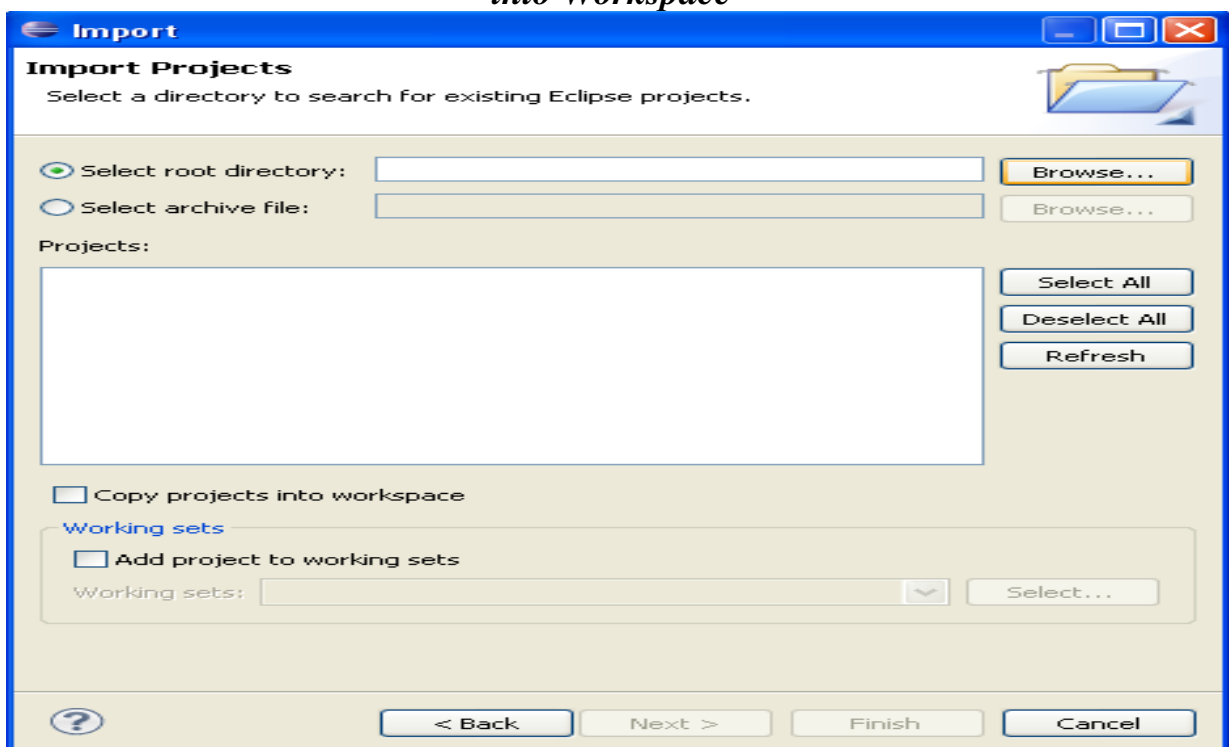
Les captures d'écran suivantes explicitent cette manipulation.



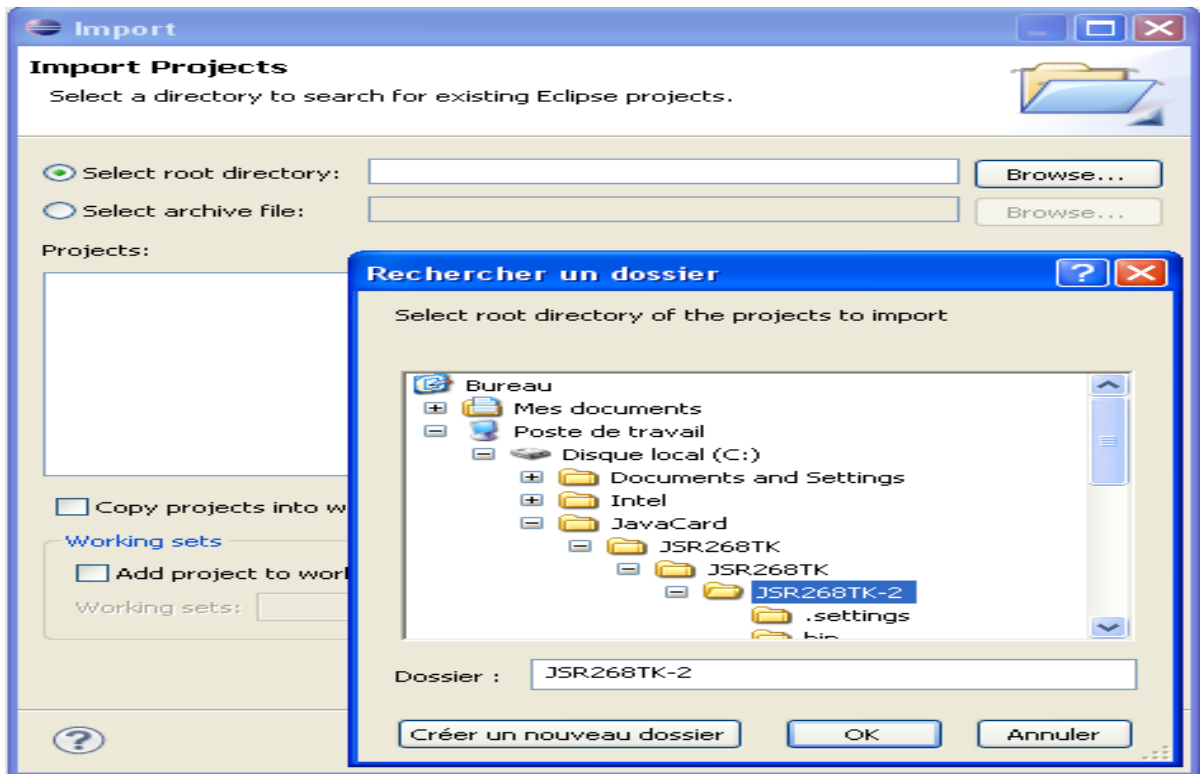
Cliquer alors sur le bouton *File* par la suite sur *Import*



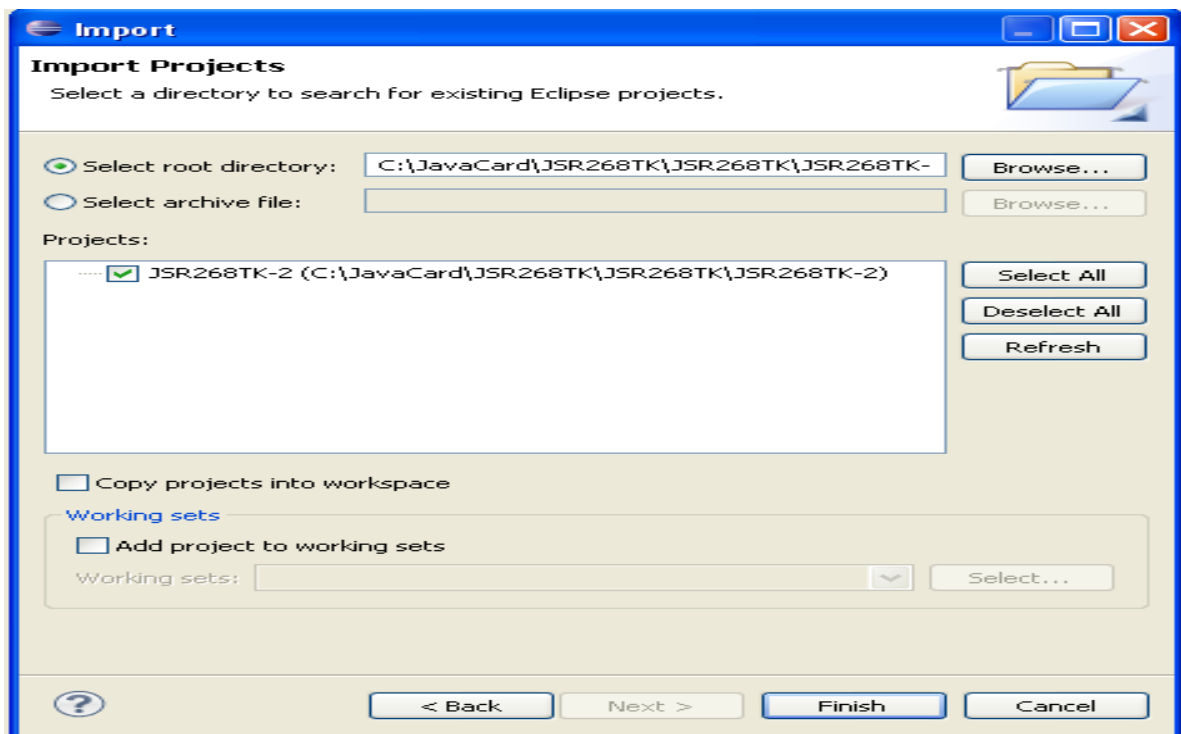
Cliquer alors sur le bouton + devant *General* → Cliquer sur *Existing Projects into Workspace*



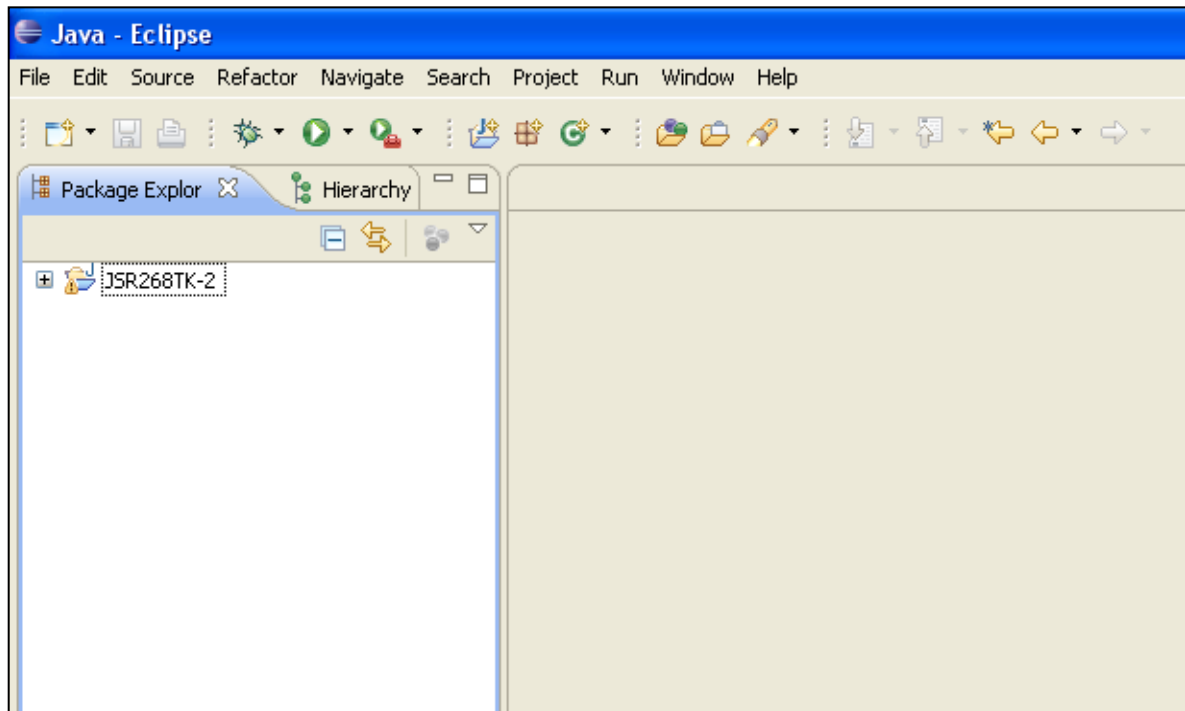
Cliquer alors sur le bouton *Browse* pour chercher le projet à importer.



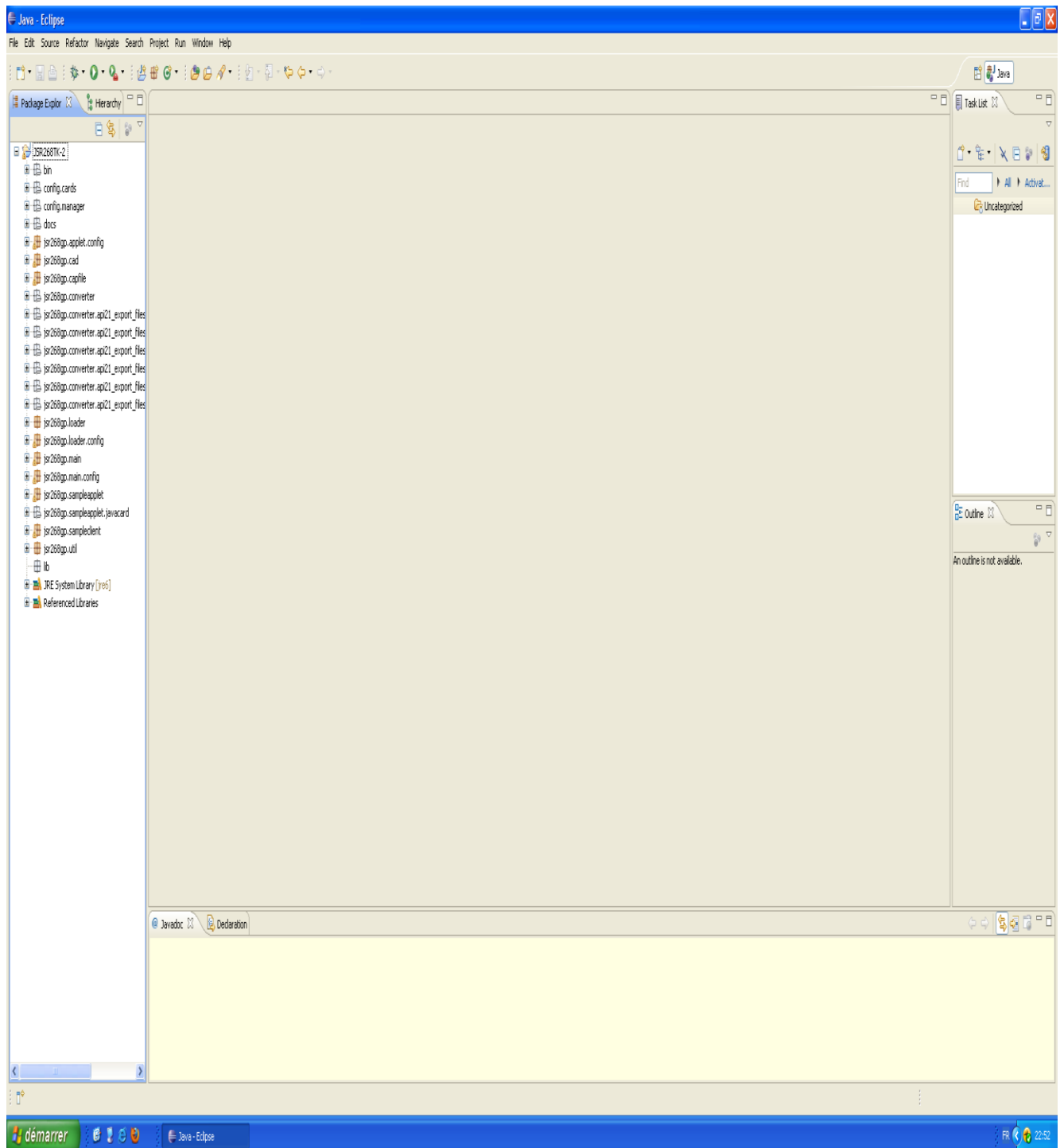
Cliquer alors sur le bouton + devant **C** → Cliquer sur + JavaCard → cliquer sur + JSR268TK → Cliquer sur + JSR268TK → Cliquer sur JSR268TK-2 → Cliquer sur le bouton **OK**



Cliquer sur le bouton *Finish*



Vous devriez voir cette fenêtre, mais afin de visualiser les packages de ce projet cliquer sur « + » devant JSR268TK-2



Vous devriez voir cette fenêtre

Ci-dessous un schéma qui décrit d'une manière générale les différentes parties de l'environnement de développement Java Card. Par la suite, nous détaillerons

chaque partie ainsi que sa configuration afin que nous puissions développer une application Java Card (côté client et côté carte).

build.xml : comporte l'exécution de trois programmes
deleter : *supprime* l'ancienne applet sur la carte.
loader : charge la nouvelle applet sur la carte.
run : exécute le programme du terminal.

ManagerConfig.xml : sert à configurer le lecteur de carte à puce en mettant son nom.

SampleTestApplet.java : le squelette de l'applet

Build.xml : sert à compiler SampleTestApplet.java et générer éventuellement le fichier SampleTestApplet.cap qui sera installé sur la carte. Ce fichier comprend bien entendu l'AID de l'applet ainsi que l'AID du package auquel l'applet appartient.

SampleClient.java : la partie du terminal qui va interroger la carte.

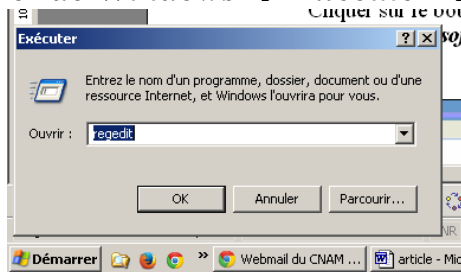
II.2 Configuration de l'outil :

- a) Modification du *ManagerConfig.xml* en mettant le nom du lecteur connecté à la machine (qui est le terminal).
- b) Développement de la partie terminal (*SampleTestApplet.java*).
- c) Développement de la partie carte (*SampleTestApplet.java*).
- d) Compilation de l'applet.
- e) Installation de l'applet sur la carte.
- f) Exécution de la partie terminal (c-à-d le client).

a) Modification du *ManagerConfig.xml*

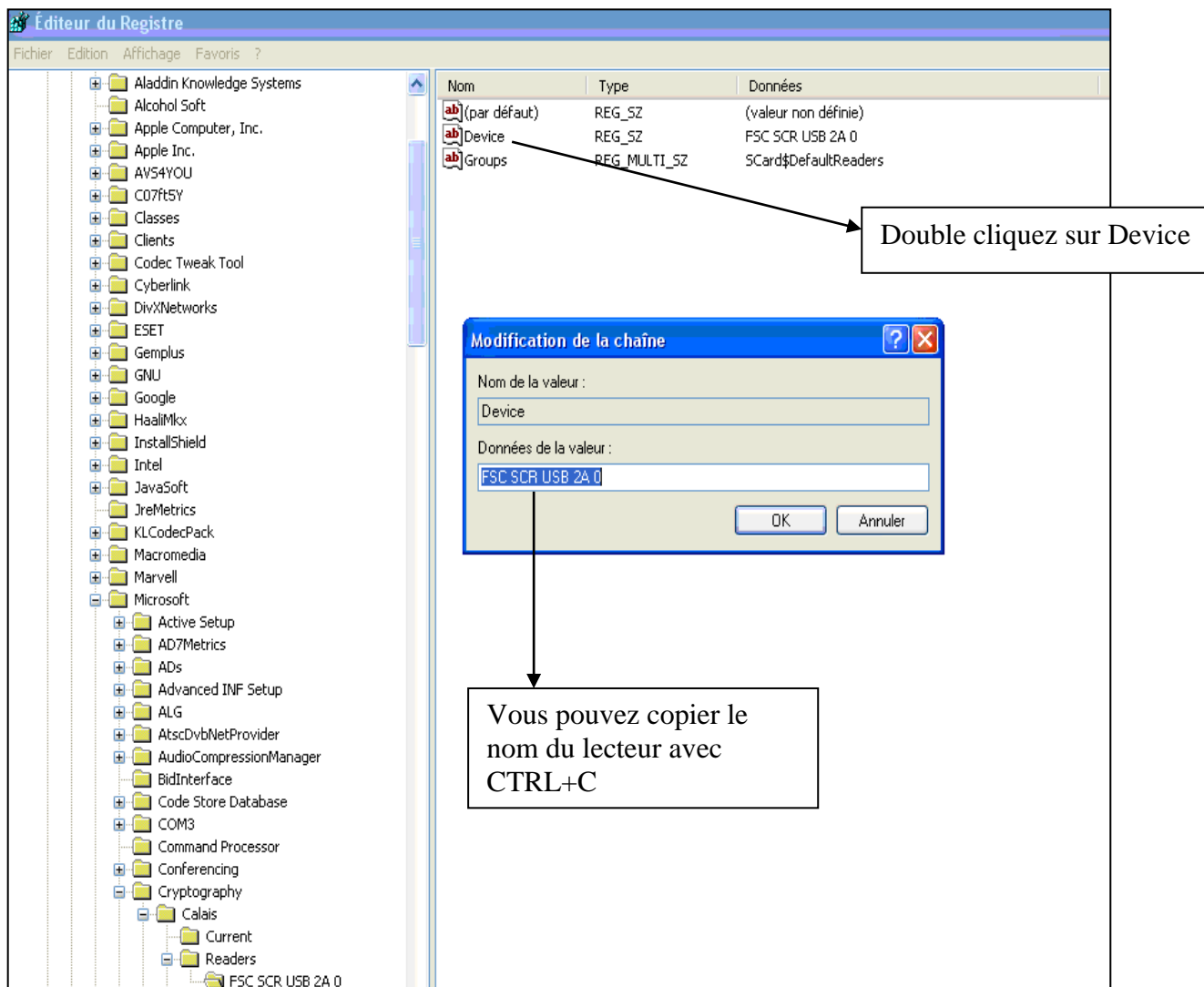
- La première étape consiste à modifier le fichier *ManagerConfig.xml* en insérant le nom du lecteur déjà connecté physiquement à la machine. Le nom exact du lecteur se trouve dans la base des registres. Nous l'obtenons de la manière suivante :

Bouton Démarrer de Windows → Exécuter → Taper: regedit



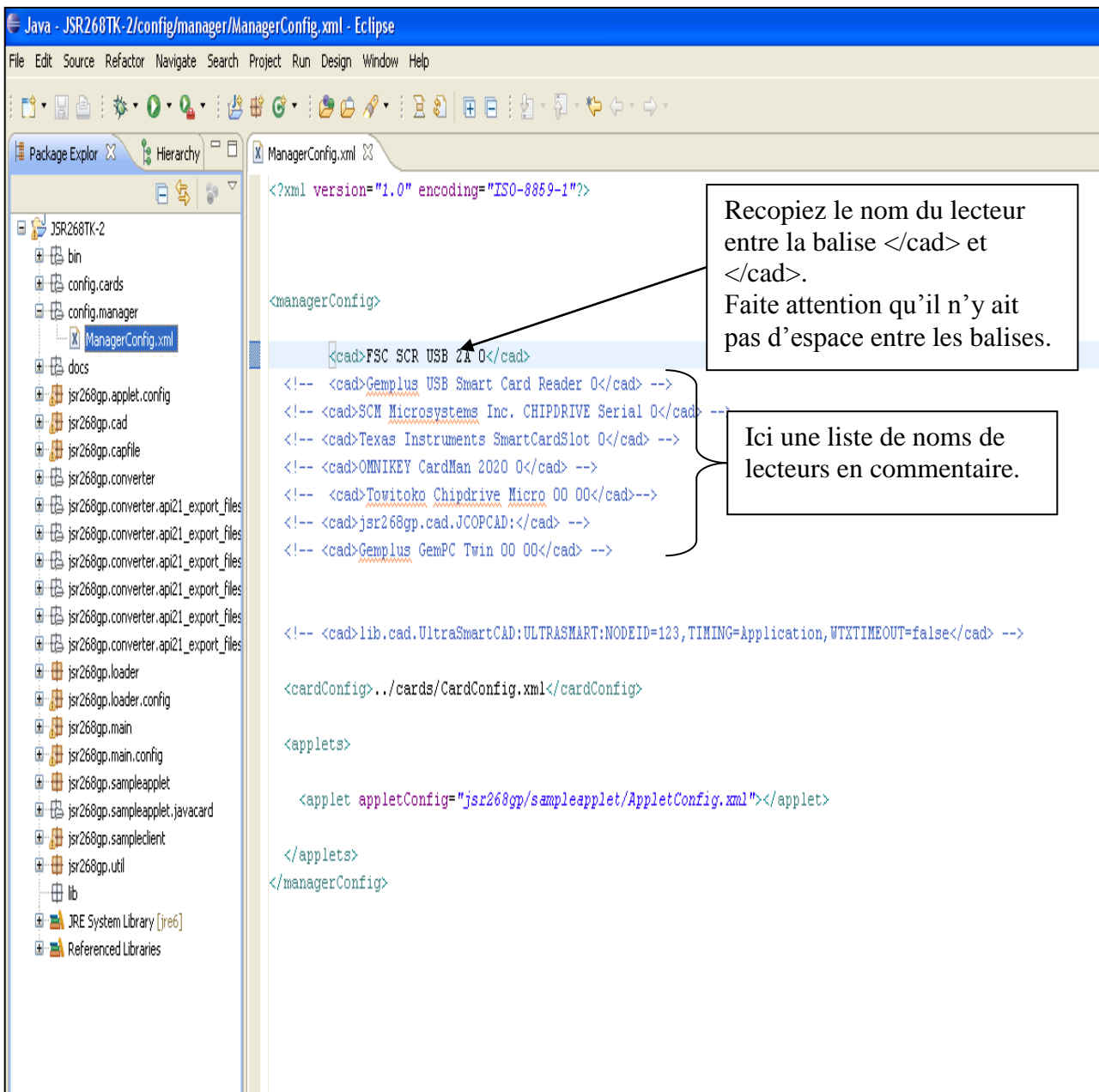
Le nom du lecteur se trouve dans l'emplacement suivant:

Cliquer sur le bouton + devant Hkey local machine → + software → + Microsoft → + cryptography → + calais → + readers



Ouvrir le fichier *ManagerConfig.xml* (un double clic de souris sur le fichier).

- Enregistrer les modifications en tapant Ctrl+S.



b) Développement de la partie terminal :

- La partie Terminal correspond au fichier Java standard *SampleClient.java*. Rappelons que le client est un programme Java standard. Seule l'applet qui sera installée sur la carte est écrite en Java Card.

- Ouvrir le fichier *SampleClient.java* (un double clic de la souris sur le fichier). Effacez complètement le contenu de ce fichier avec CTRL+A et la touche suppr. Puis téléchargez le programme en ligne suivant qui est un Client Echo: <http://cedric.cnam.fr/~bouzefra/cours/SampleClient.java> et recopier son contenu dans le fichier *SampleClient.java*.

Même si les captures d'écran suivantes ne correspondent pas vraiment au programme du Client Echo, vous trouverez tout de même les mêmes composantes que dans le programme Echo.

```

package jsr268gp.sampleclient;
import javax.smartcardio.ATR;

public class SampleClient {
    public static final byte CLA = (byte) 0x80;
    public static final byte INS_SET = (byte) 0x10;
    public static final byte INS_GET = (byte) 0x20;

    static byte AID [] = {
        (byte) 0x&0D, (byte) 0x00, (byte) 0x00, (byte) 0x18,
        (byte) 0x50, (byte) 0x00, (byte) 0x00, (byte) 0x00,
        (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x52,
        (byte) 0x41, (byte) 0x44, (byte) 0x41
    };

    public SampleClient() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @param args
     */
    public static void main(String[] args) {
        //Sélectionner votre lecteur de carte
        TerminalFactory tf = TerminalFactory.getDefault();
        CardTerminals list = tf.terminals();
        CardTerminal cad = list.getTerminal("USB CCID Smart Card Reader 0");

        if (cad == null) {
            System.out.println("debug1");
        }
        try {
            // Remarque: pour le transtypage de tableau byte vers une variable string en hexa
            // utiliser la méthode byteArrayToHexString de la classe Util

            //Etablir la connexion avec la carte à puce
            Card c = cad.connect("T=0");
            System.out.println("Card : "+c);
            System.out.println("\n");
        }
    }
}

```

Déclaration de Constantes

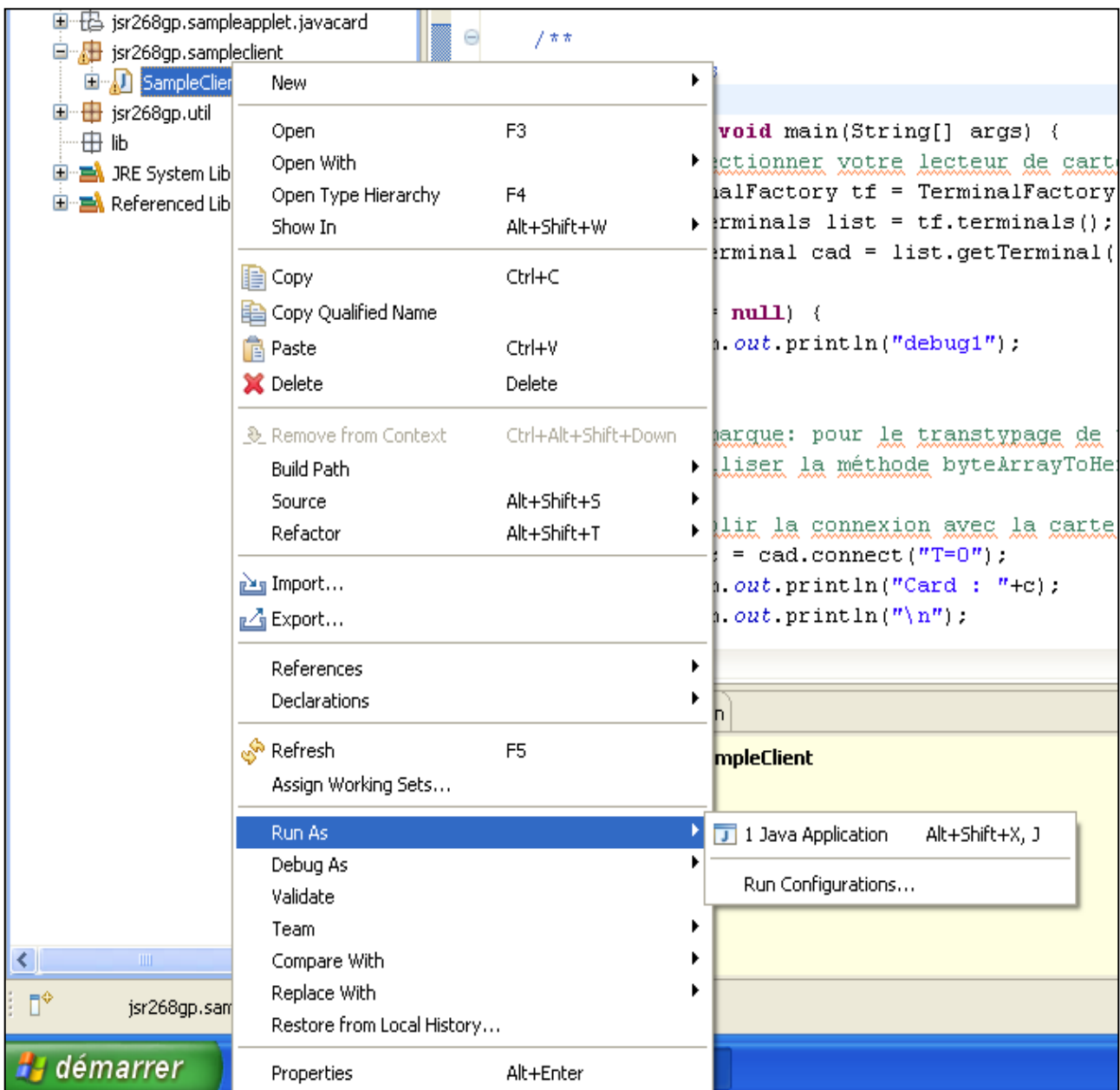
Déclaration de l'AID de l'applet qui va interagir avec cette partie terminal

- Pour exécuter le programme Client, il faut suivre les étapes suivantes. Toutefois, le lancement d'un Client n'aura un sens que lorsque l'applet sera

installée sur la carte. Car c'est le Client qui envoie des commandes APDU à l'applet de la carte.

- Cliquer sur + devant *jsr268.sampleClient*.
- Cliquer sur le bouton droit de la souris sur le fichier *SampleClient.java*.
- Sélection **Run As**.
- Cliquer **Java Application**.

Une console affiche le résultat ou d'éventuelles erreurs.



c) Développement de la partie carte (*SampleTestApplet.java*).

-La partie carte correspond au programme *SampleTestApplet.java*.

(Remarque : Après chaque modification de cette partie n'oubliez pas d'enregistrer en cliquant sur Ctrl+S).

-Ouvrir le fichier *SampleTestApplet.java* (un double clic de la souris sur le fichier).

- Télécharger le fichier en ligne suivant qui correspond au programme de l'applet Echo :

(<http://cedric.cnam.fr/~bouzefra/cours/SampleTestApplet.java>)

- Remplacer le contenu de *SampleTestApplet.java* par l'applet Echo ainsi téléchargée.

Ces captures d'écran ne contiennent pas l'applet Echo mais montrent bien les composants de toute applet Java Card.

The screenshot shows the Eclipse IDE with the file *SampleTestApplet.java* open. The code is as follows:

```
package jsr268gp.sampleapplet;
import javacard.framework.*;
public class SampleTestApplet extends Applet {

    public static final byte CLA = (byte) 0x80;
    public static final byte INS_SET = (byte) 0x10;
    public static final byte INS_GET = (byte) 0x20;
    private static byte echo = (byte) 0x01;

    public boolean select () {
        return (true);
    }

    public void deselect () {}

    public SampleTestApplet () { super(); }

    public static void install (byte [] bArray, short bOffset, byte bLength) throws ISOException {
        SampleTestApplet s = new SampleTestApplet ();
        s.register ();
    }

    public void process (APDU apdu) throws ISOException {
        byte [] buffer = apdu.getBuffer ();
        if (selectingApplet ()) return;

        if (buffer[ISO7816.OFFSET_CLA] != CLA)
            ISOException.throwIt (ISO7816.SW_CLA_NOT_SUPPORTED);

        switch (buffer[ISO7816.OFFSET_INS]) {
            case INS_SET : setByte (apdu); break;
            case INS_GET : getByte (apdu); break;
            default :
                ISOException.throwIt (ISO7816.SW_INS_NOT_SUPPORTED);
        }
    }

    private void getByte (APDU apdu) {
        short le=apdu.setOutgoing ();
        apdu.setOutgoingLength (le);
        apdu.sendBytes ((short) 0, le);
    }

    private void setByte (APDU apdu) {
        byte [] buffersandu getBuffer ();
    }
}
```

Annotations in the image:

- SampleApplet.java :** Concevoir la partie applet. Double clique sur *SampleApplet.java* (points to the Package Explorer).
- Déclaration des Constantes** (points to the constant declarations).
- Méthodes : *install()*, *select()*, *deselect()*.** (points to the *select*, *deselect*, and *install* methods).
- Méthode *process ()* : intercepte toutes les commande APDUs.** (points to the *process* method).
- Choisir quelle méthode à exécuter, code d'instruction dans la Commande APDU** (points to the switch statement in the *process* method).
- Implémentation des méthodes.** (points to the *getByte* and *setByte* private methods).

Cliquer sur + devant *jsr268.sampleApplet* afin de visualiser le contenu du package. Nous avons trois fichiers :

SampleTestApplet.java : est l'applet que nous allons installer sur la carte

Appletconfig.xml : sert à la configuration de la carte (à ne pas modifier).

Build.xml : contient l'AID de l'applet ainsi que l'AID du package. Il sert à la compilation de l'applet autrement dit la génération du fichier .cap qui sera installé par la suite sur la carte.

e) Compilation de l'applet.

Afin de compiler l'applet, nous procédons comme suit :

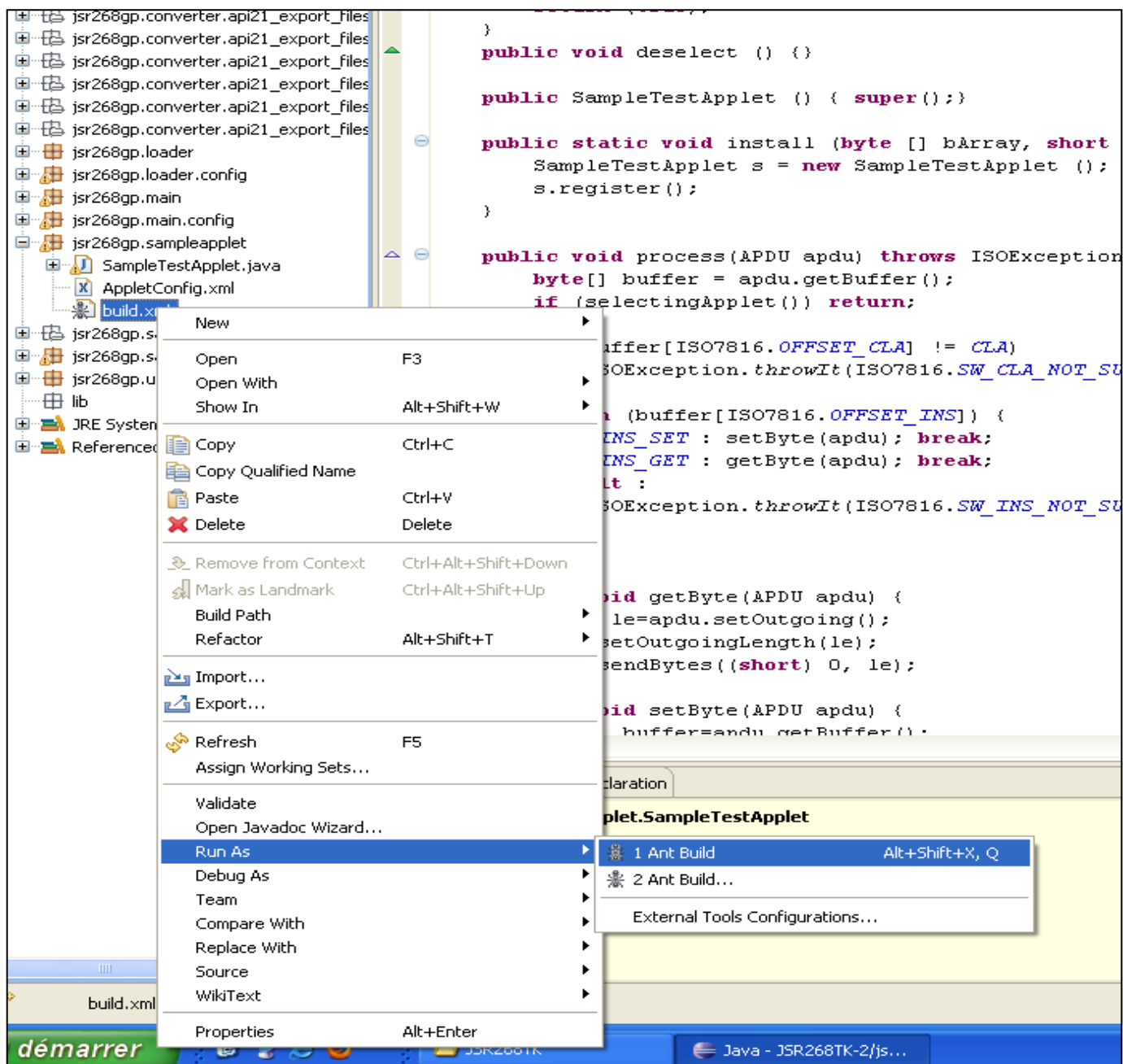
Dans le paquetage *jsr268.sampleApplet*, faire un clic-droit sur le fichier

build.xml

Sélection ***Run As***

Cliquer sur ***Ant Build***

S'il n'y a aucune erreur, un fichier cap est généré, il correspond à l'applet *SampleTestApplet.java*. Il ne reste qu'à charger ce fichier sur la carte et exécuter le client afin d'interroger l'applet.



g) Installation de l'applet sur la carte.

Charger le fichier cap sur la carte :

Cliquer sur le bouton + devant le package *bin*.

Faire un clic-droit sur le fichier *build.xml*

Sélectionner *Run As*

Cliquer sur *Ant Build*

Ce fichier va exécuter trois programmes :

delete : supprime l'ancien fichier (applet) sur la carte.

load : installe le nouveau fichier .cap qui correspond au programme *SampleTestApplet.java* nouvellement généré.

run : exécute le programme *SampleClient.java* pour envoyer des commandes APDU à l'applet en vue de la tester.

Remarque :

Pour chaque modification de l'applet, il faut la recompiler et la charger sur la carte.

