

**UNIVERSITE IBN ZOHR**

**FACULTE DES SCIENCES**

**Atelier JavaCard :**

**Calcul les Heures Travaillées d’un Employé**

Java Card

Par : EN-NAHEL Aissam

**Systèmes Informatiques Distribués & Big Data**

Année universitaire 2022-2023

Table des matières

[Table des matières 1](#_Toc141120698)

[Chapitre 1 : Installation de l’environnement de travail 2](#_Toc141120699)

[Chapitre 2 : Création du Project Eclipse 12](#_Toc141120700)

[1. Le Code : 12](#_Toc141120701)

[Employer\_javacard.java 12](#_Toc141120702)

[2. Explication des instructions : 14](#_Toc141120703)

[CALCUL\_HEURES 14](#_Toc141120704)

[AFFICHE\_HEURES 14](#_Toc141120705)

[A\_ZERO\_NBR\_H 14](#_Toc141120706)

[START\_TIME\_H 14](#_Toc141120707)

[START\_TIME\_MIN 14](#_Toc141120708)

[STOP\_TIME\_H 14](#_Toc141120709)

[STOP\_TIME\_MIN 14](#_Toc141120710)

[A\_ZERO 14](#_Toc141120711)

[Chapitre 3 : Démo 15](#_Toc141120712)

[Set Applet AID 15](#_Toc141120713)

[Set Package AID 15](#_Toc141120714)

[Generate Script 16](#_Toc141120715)

[create-Employer\_javacard.script 17](#_Toc141120716)

[select-Employer\_javacard.script 17](#_Toc141120717)

[Enregistrement des variables en utilisation EEPROM 18](#_Toc141120718)

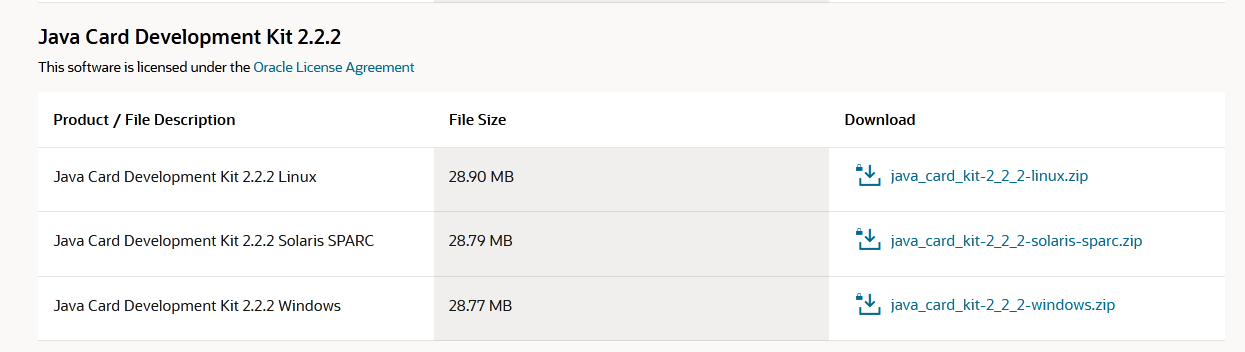
[Déployer l’Applet 18](#_Toc141120719)

[Utilisation de la carte 19](#_Toc141120720)

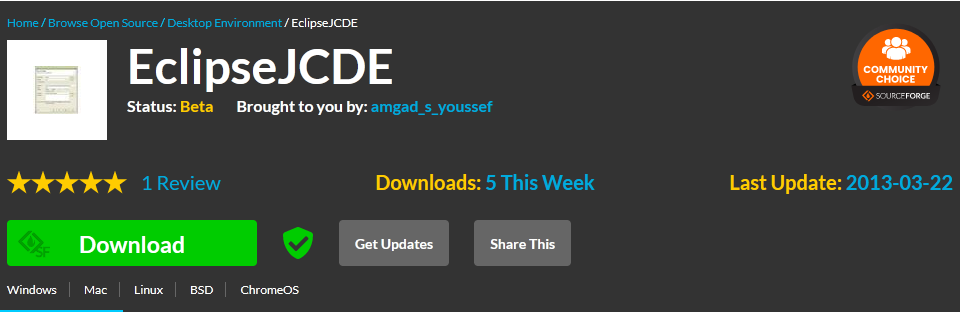
Chapitre 1 : Installation de l’environnement de travail

Pour configurer l'environnement de travail de mon projet JavaCard, je vais suivre ces étapes :

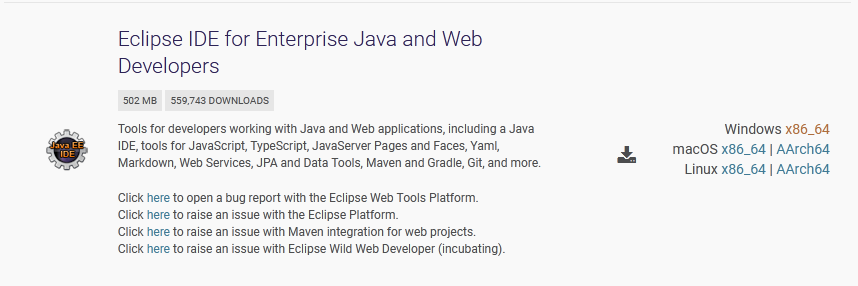
1. Tout d'abord, je vais télécharger le kit Java Card 2.2.2 pour Windows : [https://www.oracle.com/java/technologies/java-archive-downloads-javame-downloads.html](https://www.oracle.com/java/technologies/java-archive-downloads-javame-downloads.html%23license-lightbox)



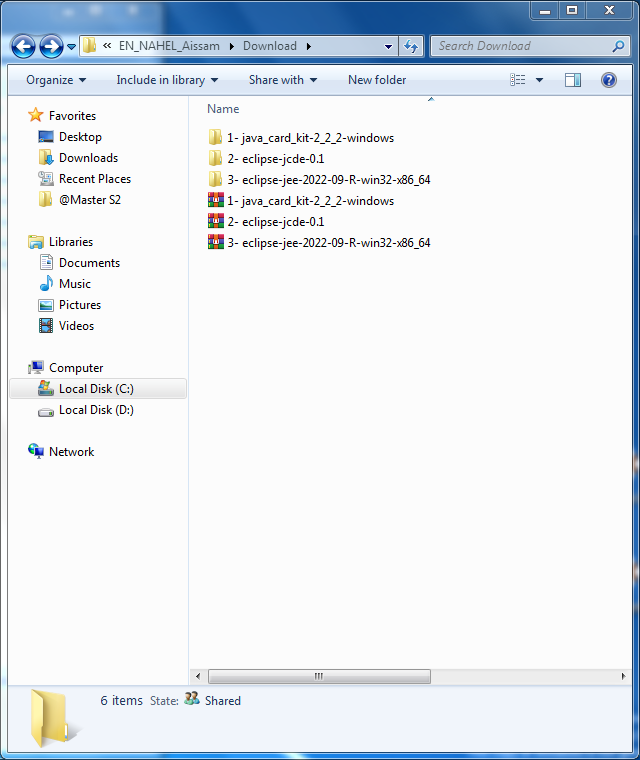
1. Ensuite, je vais télécharger Eclipse JCDE (Java Card Development Environment) : <http://sourceforge.net/projects/eclipse-jcde/>



1. Je vais utiliser Eclipse JEE version 2022-09-R pour Windows que j'ai déjà téléchargée à partir du lien suivant :: <https://www.eclipse.org/downloads/packages/release/2022-09/r>



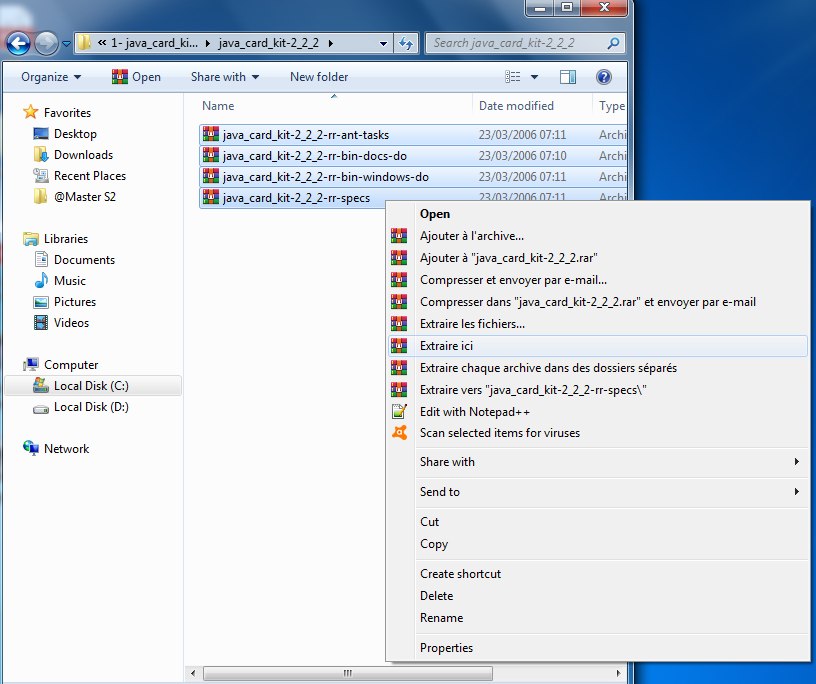
Voici tous les fichiers nécessaires :



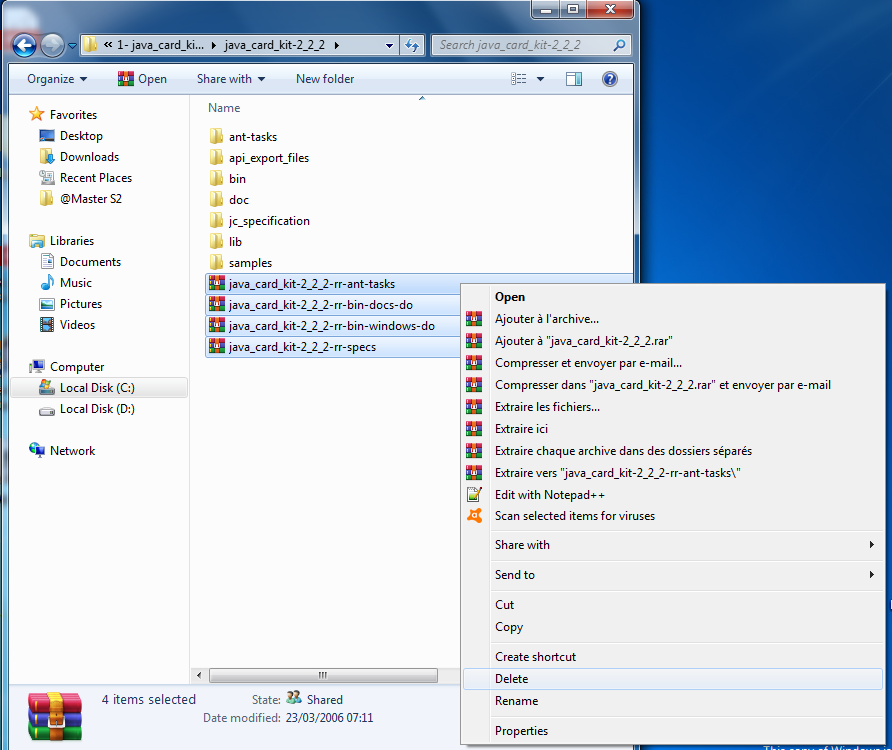
Une fois que j'aurai téléchargé ces fichiers, je commencerai par installer le Java Card kit 2.2.2

Je décompressai le fichier 1-java\_card\_kit-2\_2\_2-windows.zip

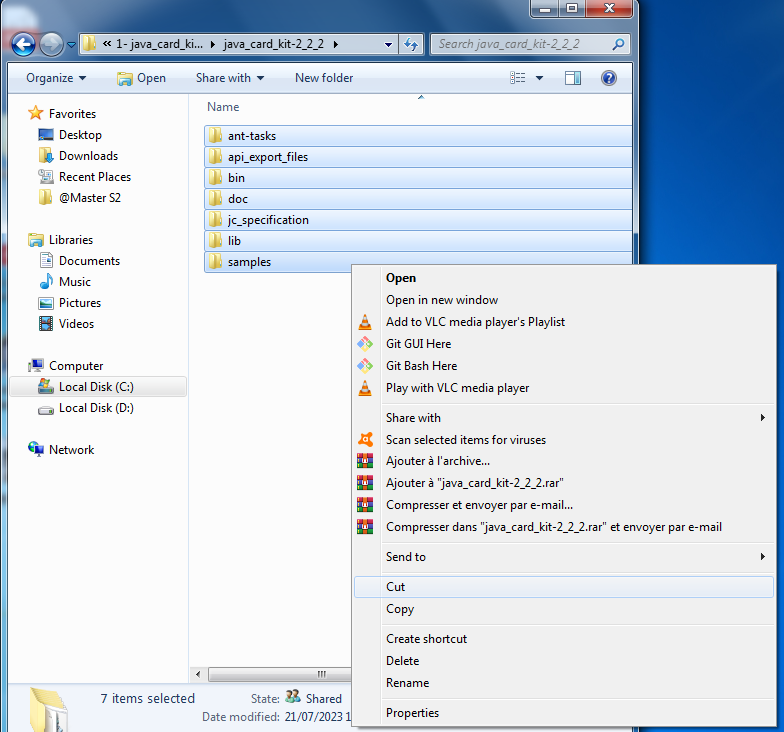
Ensuite, je décompressai les fichiers qui se trouvaient à l'intérieur du dossier 1-java\_card\_kit-2\_2\_2-windows\java\_card\_kit-2\_2\_2\ :

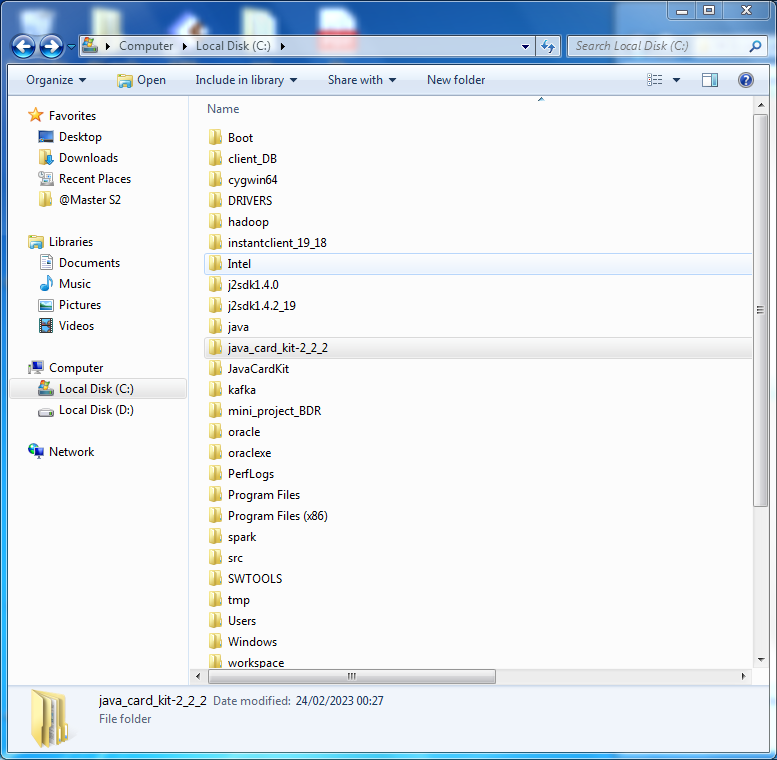


Maintenant, ces fichiers n'ont pas de rôles, il faut les supprimer :

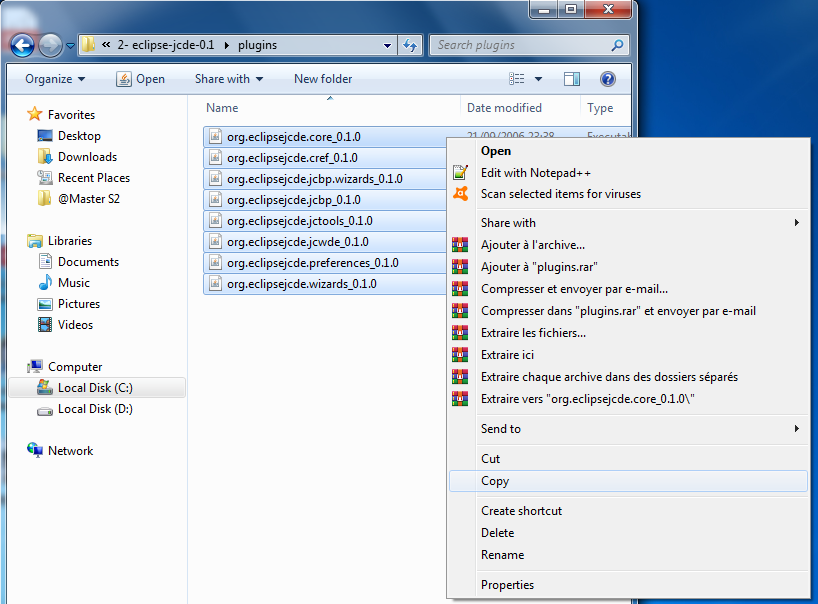


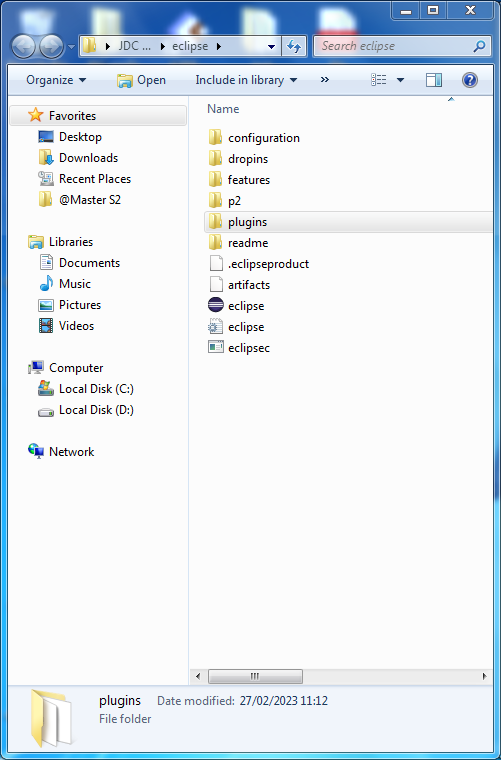
Je déplaçai les dossiers extraits vers le répertoire C:\java\_card\_kit-2\_2\_2-windows\



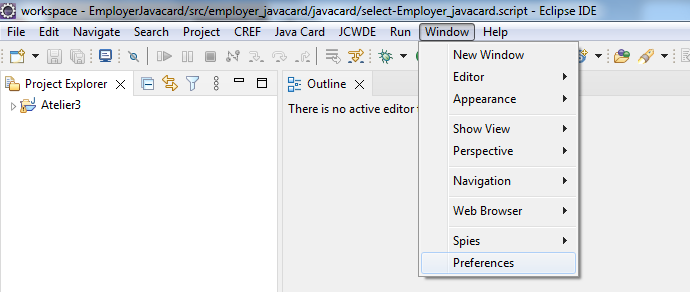


Puis j'ai configuré Eclipse : Aprés je décompressai le fichier 2-eclipse-jcde-0.1.zip et le fichier 3-eclipse-jee-2022-09-R-win32-x86\_64.zip, j’ai copie les fichiers jars qui se trouve dans 2-eclipse-jcde-0.1\**plugins** vers 3-eclipse-jee-2022-09-R-win32-x86\_64\eclipse\**plugins**:



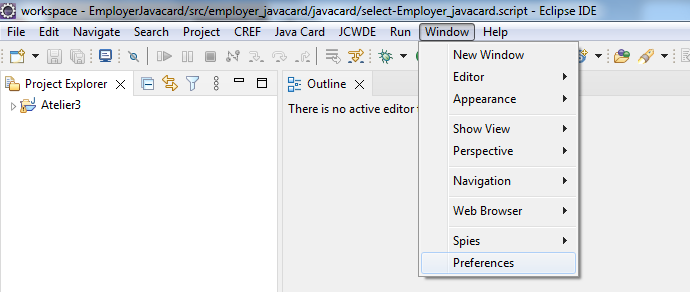


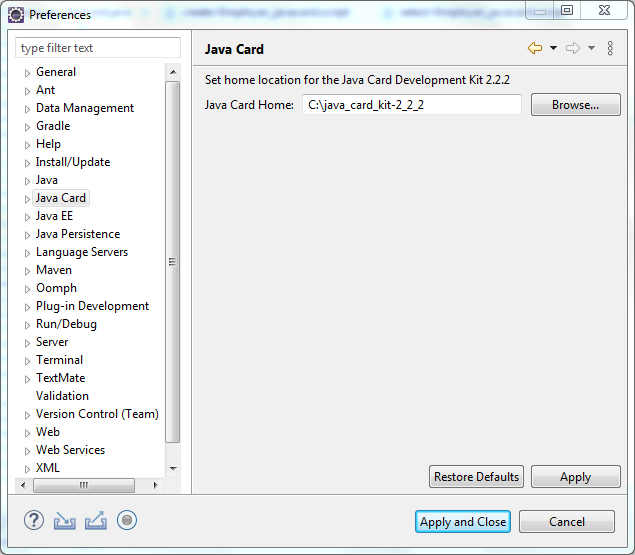
Maintenant, les options liées à Java Card sont activées dans Eclipse :



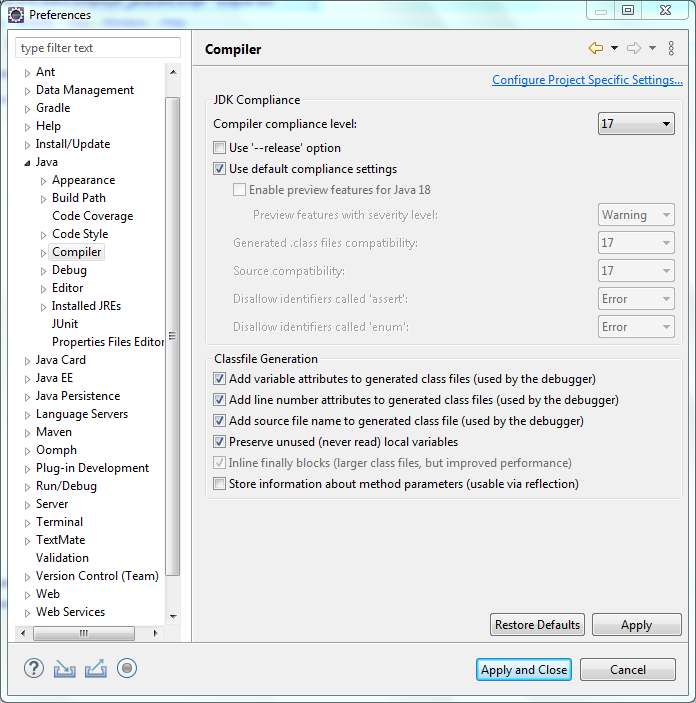
Il faut déterminer le chemin du Java Card kit 2.2.2 dans Eclipse :

Window 🡪 Preferences 🡪 Java Card

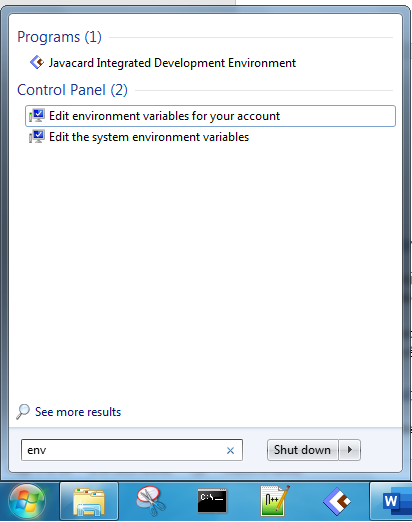




Et pour le Compiler JDK, J'utilisai **17** :

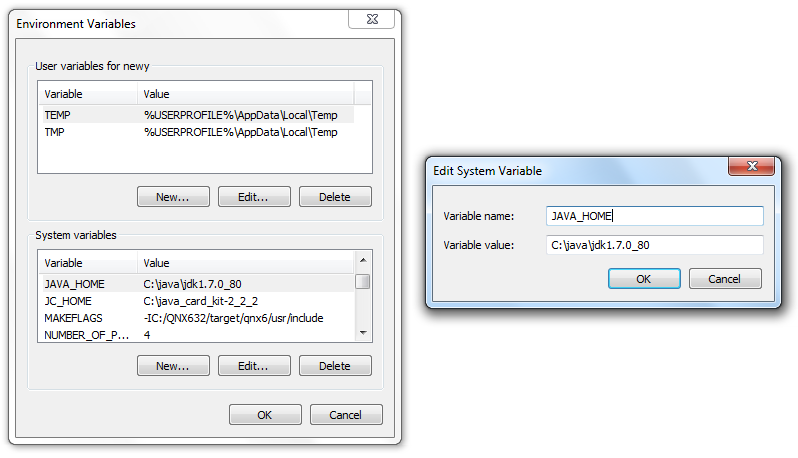


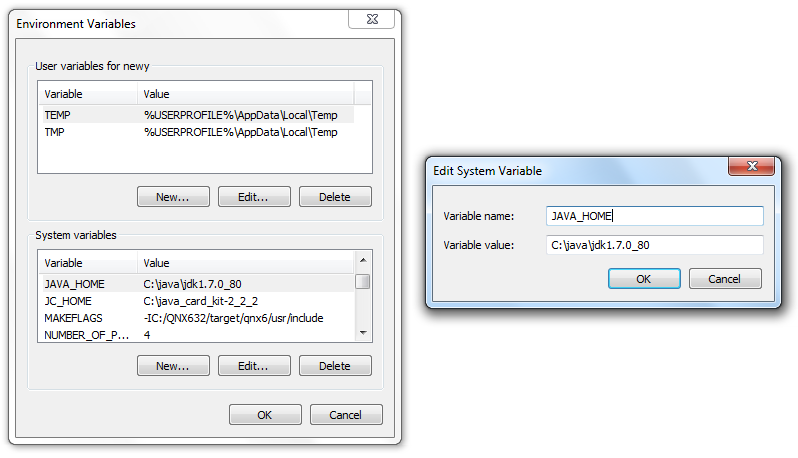
La dernière étape consiste à configurer les variables d'environnement :



J'utilisai le JDK 1.7 :

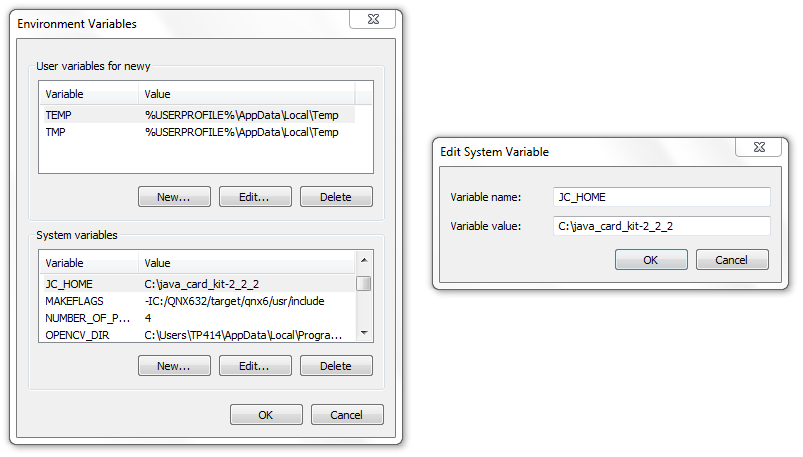
**JAVA\_HOME** : C:\java\jdk1.7.0\_80

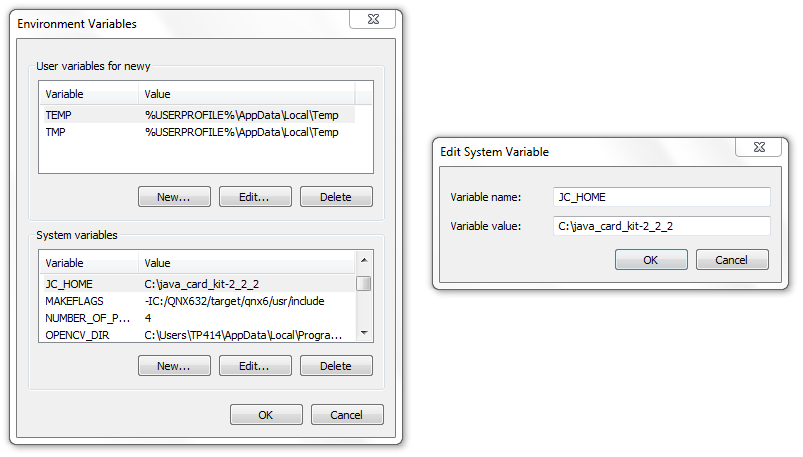




Il faut ajouter nouvelle variable :

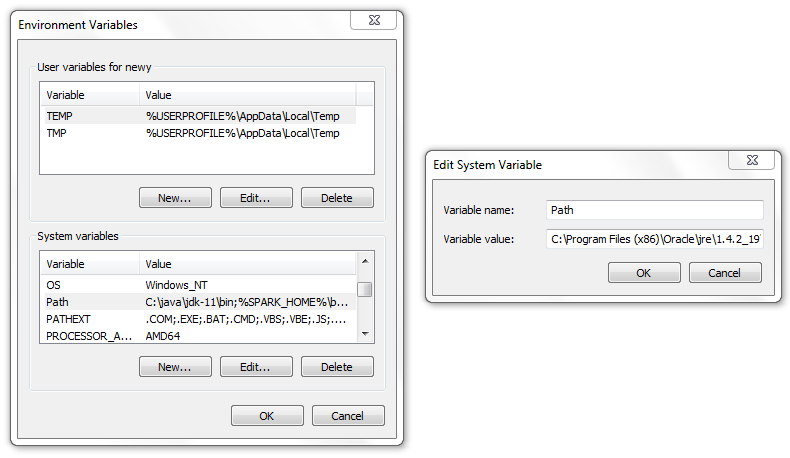
**JC\_HOME** : C:\java\_card\_kit-2\_2\_2

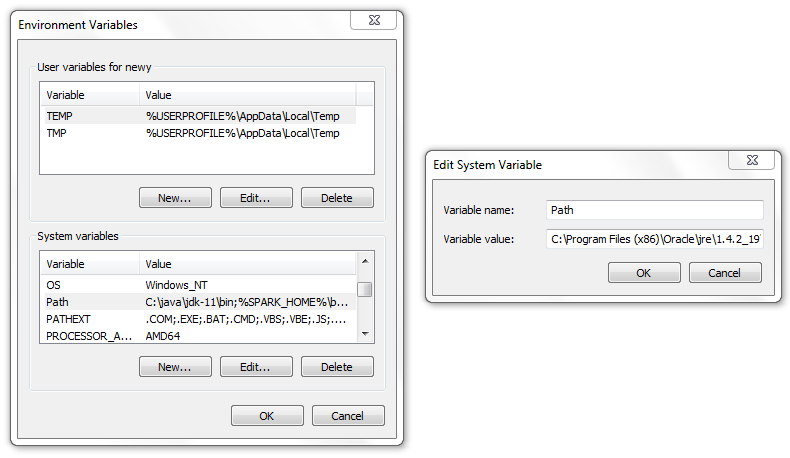


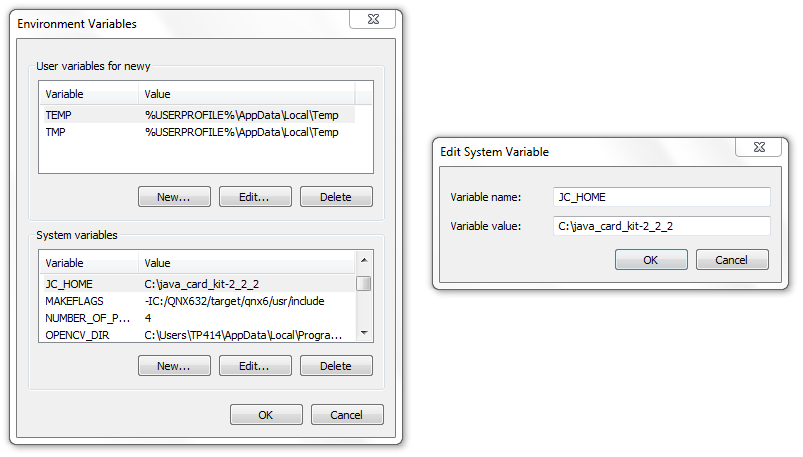


Et enfin, il faut ajouter le chemin de Java compatible avec Java Card à la variable d'environnement "**Path**". J'ai utilisé Java 1.4 :

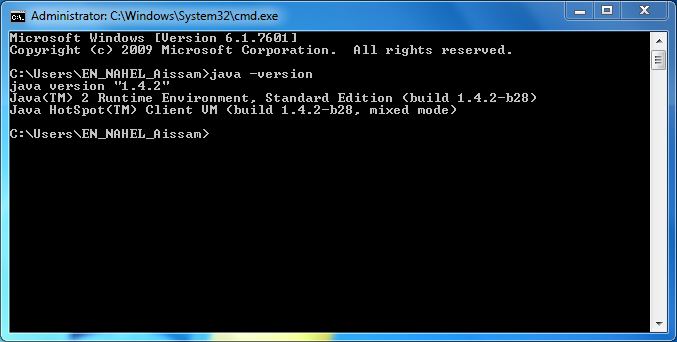
**Path** : C:\Program Files (x86)\Oracle\jre\1.4.2\_19\bin;





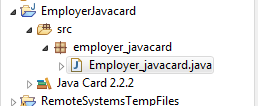


Pour vérifier la version de Java:



Chapitre 2 : Création du Project Eclipse

Le projet Java Card "**EmployerJavacard**" est composé d'un seul fichier Java Card, à savoir "**Employer\_javacard.java**" qui contient l'implémentation de l'applet Java Card qui gère le calcul des heures travaillées d'un employé.



1. Le Code :

Employer\_javacard.java

**package** employer\_javacard;

**import** javacard.framework.APDU;

**import** javacard.framework.Applet;

**import** javacard.framework.ISO7816;

**import** javacard.framework.ISOException;

**public** **class** Employer\_javacard **extends** Applet {

**private** **byte** nbr\_hours = 0;

**private** **byte** start\_hour = 0;

**private** **byte** start\_min = 0;

**private** **byte** stop\_hour = 0;

**private** **byte** stop\_min = 0;

**private** **static** **final** **byte** ***CLA*** = (**byte**) 0xB0;

**private** **static** **final** **byte** ***CALCUL\_HEURES*** = (**byte**)0x01;

**private** **static** **final** **byte** ***AFFICHE\_HEURES*** = (**byte**)0x02;

**private** **static** **final** **byte** ***A\_ZERO\_NBR\_H*** = (**byte**)0x03;

**private** **static** **final** **byte** ***A\_ZERO*** = (**byte**)0x04;

**private** **static** **final** **byte** ***START\_TIME\_H*** = (**byte**)0x05;

**private** **static** **final** **byte** ***START\_TIME\_MIN*** = (**byte**)0x06;

**private** **static** **final** **byte** ***STOP\_TIME\_H*** = (**byte**)0x07;

**private** **static** **final** **byte** ***STOP\_TIME\_MIN*** = (**byte**)0x08;

**private** Employer\_javacard() { }

**public** **static** **void** install(**byte** bArray[], **short** bOffset, **byte** bLength) **throws** ISOException {

**new** Employer\_javacard().register();

}

**public** **void** process(APDU apdu) **throws** ISOException {

**if** (**this**.selectingApplet()) { **return**; }

**byte**[] buffer = apdu.getBuffer();

**if**(buffer[ISO7816.***OFFSET\_CLA***] != ***CLA***) { ISOException.*throwIt*(ISO7816.***SW\_CLA\_NOT\_SUPPORTED***);

}

**switch**(buffer[ISO7816.***OFFSET\_INS***]) {

**case** ***CALCUL\_HEURES***:

**short** diff\_h = **(short**)(stop\_hour - start\_hour);

diff\_h = **(short**)(diff\_h\*60); //h est en minutes

**short** diff\_min = **(short**)(diff\_h + (stop\_min - start\_min));

nbr\_hours = (**byte**) (diff\_min / 60);

**if** (diff\_min % 60 >= 30) {

nbr\_hours += 1;

}

**break**;

**case** ***AFFICHE\_HEURES***:

buffer[0] = nbr\_hours;

apdu.setOutgoingAndSend((**short**)0, (**short**)1);

**break**;

**case** ***A\_ZERO\_NBR\_H***:

nbr\_hours = 0;

**break**;

**case** ***A\_ZERO***:

start\_hour = 0;

start\_min = 0;

stop\_hour = 0;

stop\_min = 0;

**break**;

**case** ***START\_TIME\_H***:

apdu.setIncomingAndReceive();

start\_hour = buffer[ISO7816.***OFFSET\_CDATA***];

**break**;

**case** ***START\_TIME\_MIN***:

apdu.setIncomingAndReceive();

start\_min = buffer[ISO7816.***OFFSET\_CDATA***];

**break**;

**case** ***STOP\_TIME\_H***:

apdu.setIncomingAndReceive();

stop\_hour = buffer[ISO7816.***OFFSET\_CDATA***];

**break**;

**case** ***STOP\_TIME\_MIN***:

apdu.setIncomingAndReceive();

stop\_min = buffer[ISO7816.***OFFSET\_CDATA***];

**break**;

}

}

}

1. Explication des instructions :

Voici une description détaillée des différentes instructions utilisées dans le code :

CALCUL\_HEURES

L'instruction **CALCUL\_HEURES** (0x01) est utilisée pour calculer le nombre d'heures travaillées. Elle effectue la différence entre l'heure de début du travail et l'heure de fin du travail afin de déterminer la durée totale des heures travaillées.

AFFICHE\_HEURES

L'instruction **AFFICHE\_HEURES** (0x02) est utilisée pour afficher le nombre d'heures travaillées.

A\_ZERO\_NBR\_H

L'instruction **A\_ZERO\_NBR\_H** (0x03) est utilisée pour mettre le nombre d'heures travaillées à zéro. Cette instruction est exécutée une seule fois, lors de la première utilisation de l'applet, elle permet d'initialiser le compteur d'heures travaillées à zéro avant de commencer le suivi des heures.

START\_TIME\_H

L’instruction **START\_TIME\_H** a 0x05 est pour enter l’heur de début de travail et l’enregistrer à le variable "**start\_hour"** l’heur de début de travail.

START\_TIME\_MIN

L'instruction **START\_TIME\_MIN** (0x06) est utilisée pour entrer les minutes de l'heure de début du travail et les enregistrer dans la variable "**start\_min**", elle permet de spécifier les minutes de l'heure de début du travail.

STOP\_TIME\_H

L'instruction **STOP\_TIME\_H** (0x07) est utilisée pour entrer l'heure de fin du travail et l'enregistrer dans la variable "**stop\_hour**", elle permet de spécifier l'heure de fin du travail.

STOP\_TIME\_MIN

L'instruction **STOP\_TIME\_MIN** (0x08) est utilisée pour entrer les minutes de l'heure de fin du travail et les enregistrer dans la variable "**stop\_min**", elle permet de spécifier les minutes de l'heure de fin du travail.

A\_ZERO

L'instruction **A\_ZERO** (0x04) est utilisée pour mettre les variables "**start\_hour**", "**start\_min**", "**stop\_hour**" et "**stop\_min**" à zéro, elle permet de réinitialiser les valeurs des heures et des minutes, en préparation pour un nouveau calcul des heures travaillées.

Chapitre 3 : Démo

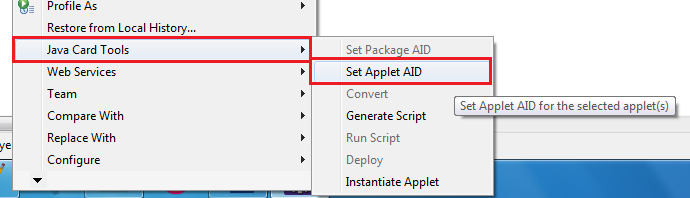
Dans ce chapitre, je vais vous présenter une démonstration de l’application de calcul des heures travaillées travers différentes étapes : la configuration de l'applet AID et du package AID, la génération du script, l'enregistrement des variables et l'utilisation de EEPROM, le déploiement de l'applet et l'utilisation l'applet.

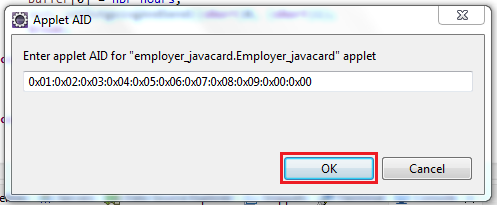
Set Applet AID

La première étape consiste à configurer l'identifiant d'applet (AID). L'AID est une valeur unique qui identifie de manière unique notre applet.

Dans Eclipse je vais suivre :

eclipse --> clic droit sur le package --> Java Card Tool --> Set Applet AID



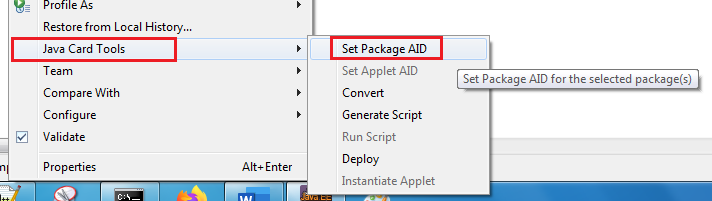


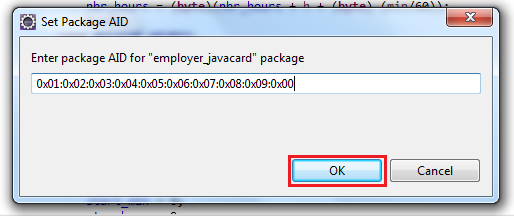
Set Package AID

En plus de l'AID, nous devons également configurer le package AID de l’applet. Le package AID est une autre valeur unique qui identifie le package auquel notre applet appartient.

Dans Eclipse je vais suivre :

eclipse --> clic droit sur le package --> Java Card Tool --> Set Package AID

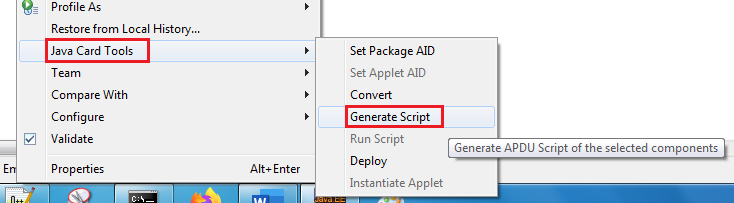




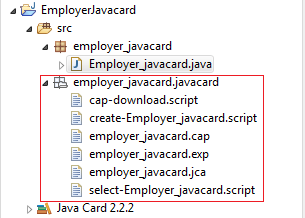
Generate Script

Je vais générer des scripts, dans Eclipse je vais suivre :

eclipse --> clic droit sur le package --> Java Card Tool --> Generate Script

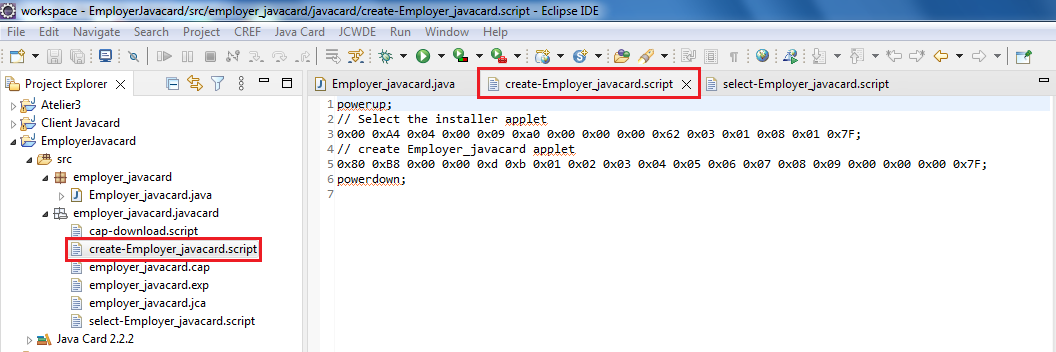


Les scripts générés sont :



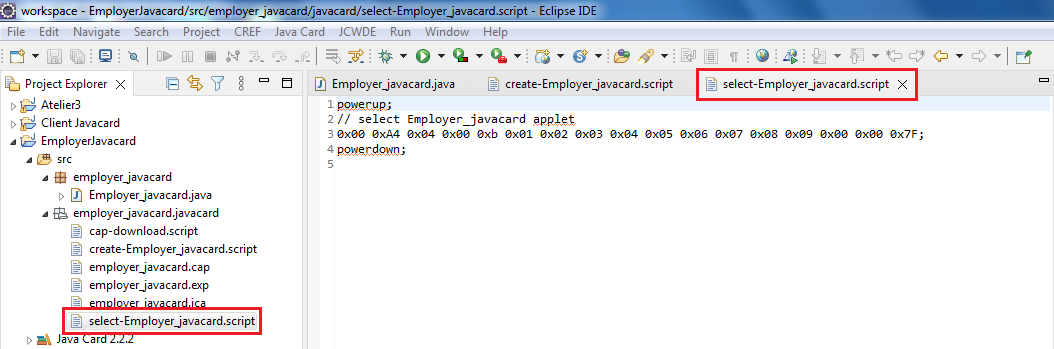
create-Employer\_javacard.script

Le fichier **create-Employer\_javacard.script** contient la commande de sélection de l'applet d'installation et la commande de création de l'applet Javacard Employeur.



select-Employer\_javacard.script

Le fichier **select-Employer\_javacard.script** contient la commande de sélection de l'applet.

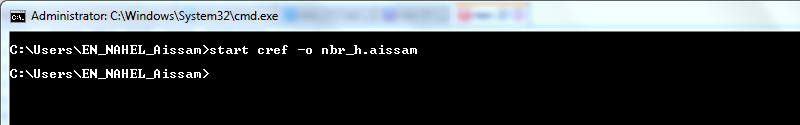


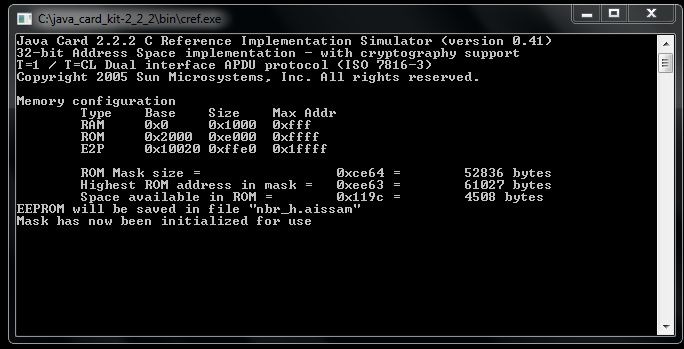
Enregistrement des variables en utilisation EEPROM

Avant de déployer l'applet, je dois enregistrer les variables nécessaires pour notre application. Cela peut inclure des variables telles que les heures de début et de fin du travail. Je vais utiliser le fichier **nbr\_h.aissam** pour stocker les données.

Dans cmd.exe :

start cref -o nbr\_h.aissam

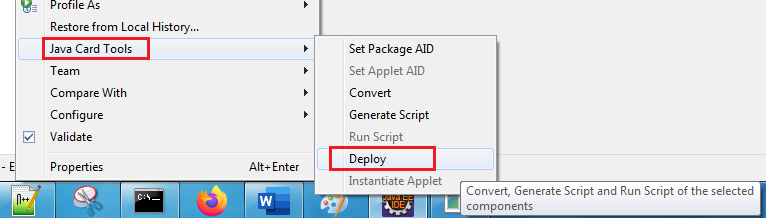


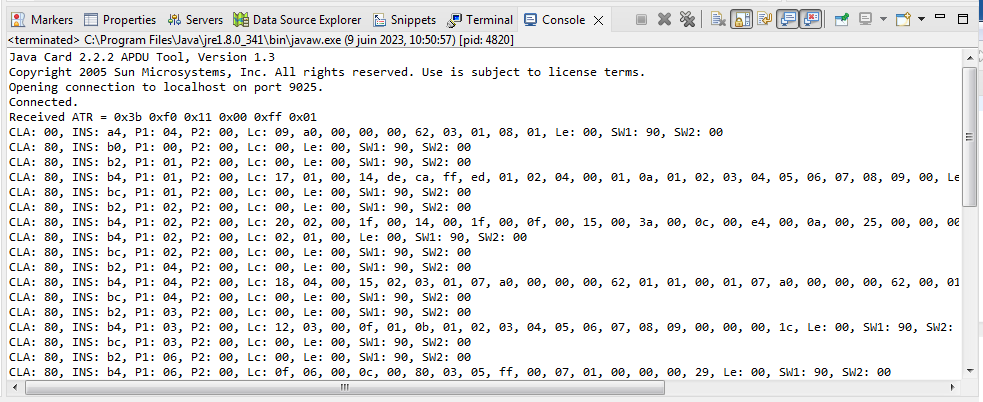


Déployer l’Applet

Je suis prêt à déployer l'applet. Dans Eclipse je vais suivre :

eclipse --> clic droit sur le package --> Java Card Tool --> Deploy



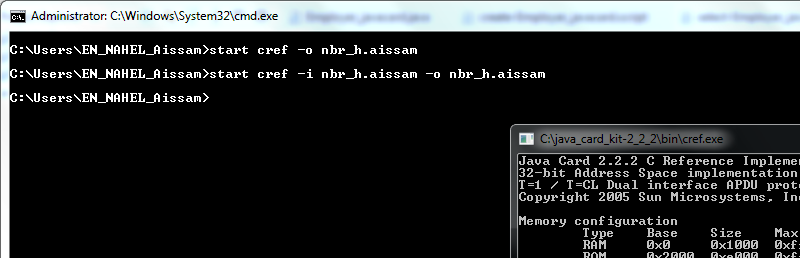


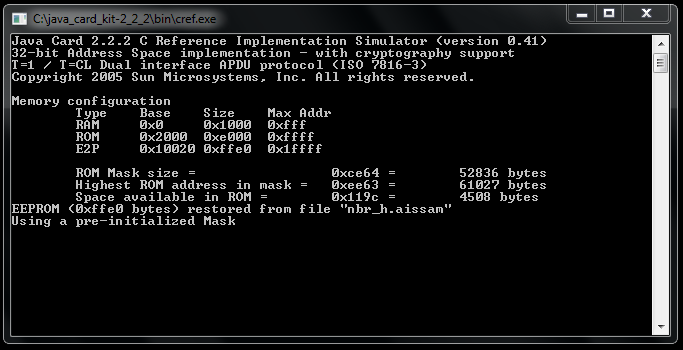
Utilisation de la carte

Une fois l'applet déployé, je vais l’utiliser pour simuler le comptage des heures travaillées. Je vais envoyer les commandes appropriées et observer les résultats affichés.

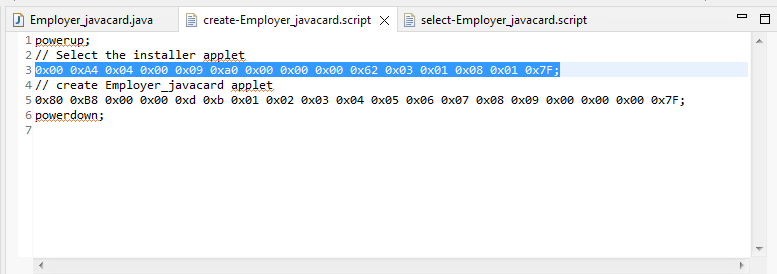
Tout d'abord, j'ai lancé la commande "start cref -i nbr\_h.aissam -o nbr\_h.aissam" dans le cmd.exe afin d'utiliser le fichier dans lequel j'ai enregistré les données des heures travaillées:

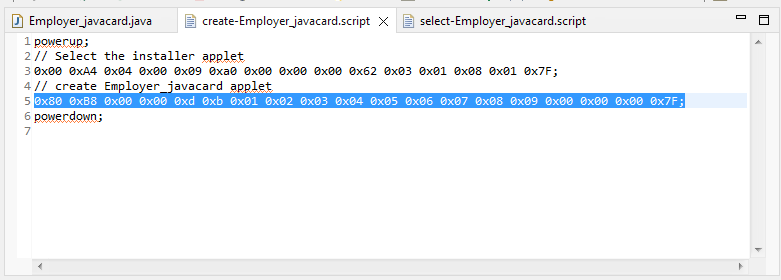
start cref -i nbr\_h.aissam -o nbr\_h.aissam



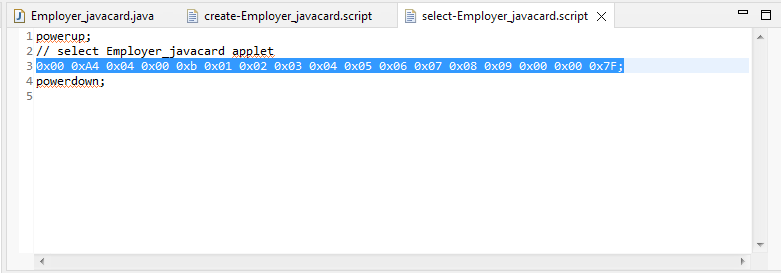


Ensuite, j'ai utilisé la commande "**apdutool**" pour établir une connexion avec l'applet. J'ai sélectionné l'installation de l'applet et la creation de l'applet JavaCard Employeur en utilisant les commandes du fichier "**create-Employer\_javacard.script**". Ces deux dernières commandes, ne sont effectuées qu'une seule fois.





Ensuite, j'ai exécuté la commande de sélection de l'applet en utilisant la commande du fichier "**select-Employer\_javacard.script**". Cette commande permet de sélectionner l'applet Employeur spécifique pour l'utilisation.



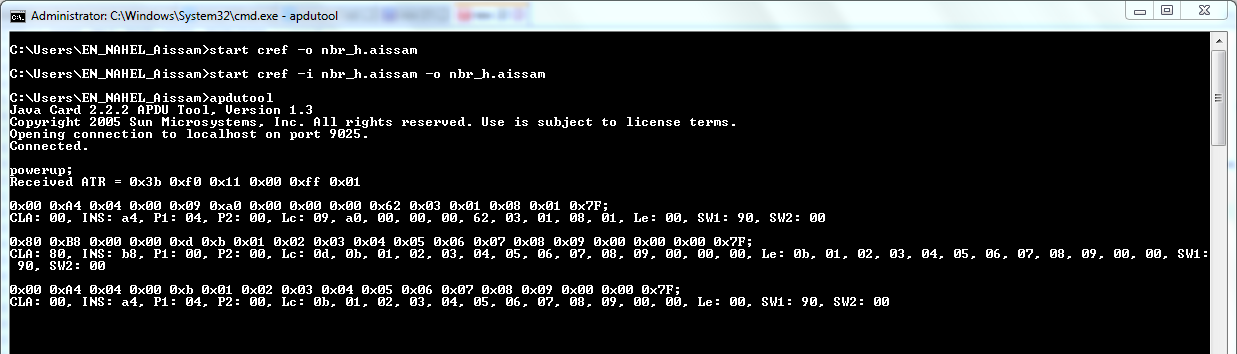
apdutool

powerup;

0x00 0xA4 0x04 0x00 0x09 0xa0 0x00 0x00 0x00 0x62 0x03 0x01 0x08 0x01 0x7F;

0x80 0xB8 0x00 0x00 0xd 0xb 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x00 0x00 0x00 0x7F;

0x00 0xA4 0x04 0x00 0xb 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x00 0x00 0x7F;



Les commandes suivantes sont utilisées pour interagir avec l'applet, elles permettent de configurer :  
- l'heure de début du travail : 11h:00min par exemple, **11** en hexadécimal est **0x0B**, et **00** en hexadécimal est **0x00**.

- l'heure de fin du travail : 12h:45min par exemple, **12** en hexadécimal est **0x0C**, et **45** en hexadécimal est **0x2D**.

Et elles permettent de calculer les heures travaillées et d'afficher les résultats.

0xB0 **0x02** 0x00 0x00 0x00 0x00; // (**AFFICHE\_HEURES**)

0xB0 **0x03** 0x00 0x00 0x00 0x00; // (**A\_ZERO\_NBR\_H**)

0xB0 **0x04** 0x00 0x00 0x00 0x00; // (**A\_ZERO**)

0xB0 **0x02** 0x00 0x00 0x00 0x00; // (**AFFICHE\_HEURES**)

//exemple de l’heure de debut de travail : 11h:00min

0xB0 **0x05** 0x00 0x00 0x01 **0x0B** 0x01; // (**START\_TIME\_H**) (**11h**:)

0xB0 **0x06** 0x00 0x00 0x01 **0x00** 0x01; // (**START\_TIME\_MIN**) (:**00min**)

//exemple de l’heure de debut de travail : 12h:45min

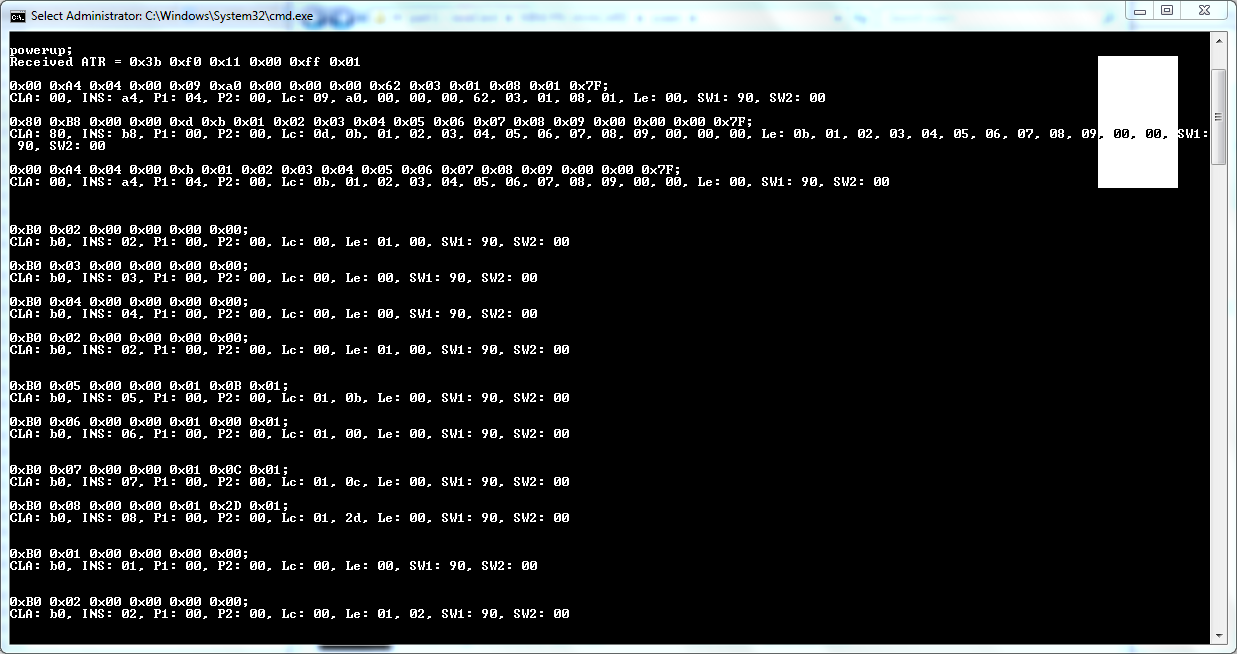
0xB0 **0x07** 0x00 0x00 0x01 **0x0C** 0x01; // (**STOP\_TIME\_H**) (**12h**:)

0xB0 **0x08** 0x00 0x00 0x01 **0x2D** 0x01; // (**STOP\_TIME\_MIN**) (:**45min**)

0xB0 **0x01** 0x00 0x00 0x00 0x00; // (**CALCUL\_HEURES**)

0xB0 **0x02** 0x00 0x00 0x00 0x00; // (**AFFICHE\_HEURES**)

0 heures travaillées

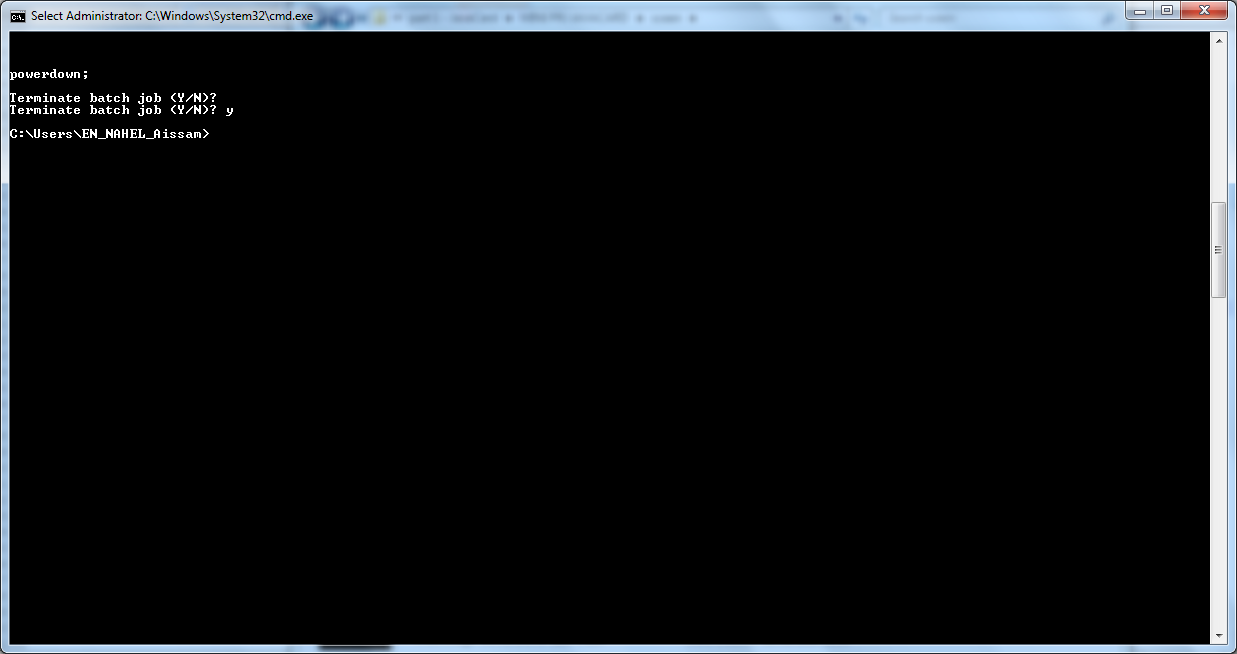


2 heures travaillées

0 heures travaillées

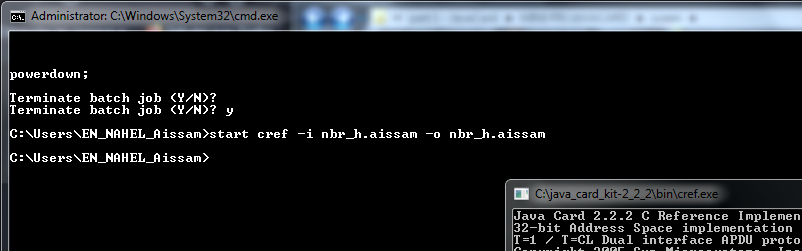
On termine la connection avec l'applet en utilisant la commande **powerdown;**

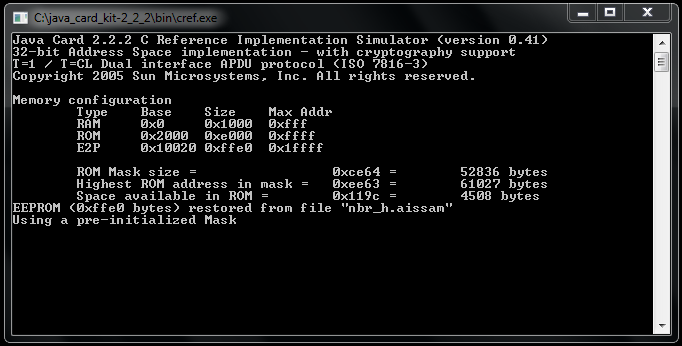
powerdown;



Pour tester si les données sont correctement enregistrées et effectuer une nouvelle simulation, j'ai à nouveau exécuté la commande "start cref -i nbr\_h.aissam -o nbr\_h.aissam" dans le **cmd.exe**, cette commande permet d'utiliser le fichier "**nbr\_h.aissam**" dans lequel j'ai précédemment enregistré les données des heures travaillées.

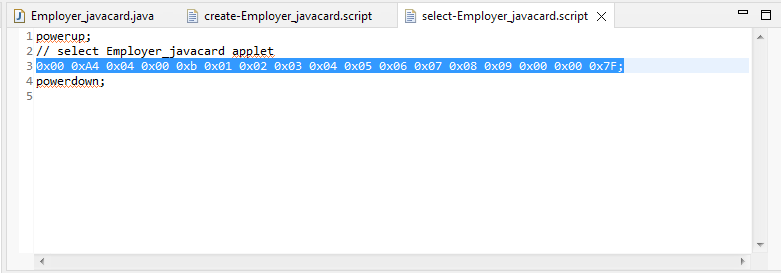
start cref -i nbr\_h.aissam -o nbr\_h.aissam





J’établir la connexion avec l'applet par "**apdutool**", puis j'ai sélectionné l'applet.

apdutool



Lors de cette deuxième interaction avec l'applet, j'ai utilisé les commandes suivantes, qui permettent de configurer :  
- l'heure de début du travail : 09h:45min par exemple, **09** en hexadécimal est **0x09**, et **45** en hexadécimal est **0x2D**.

- l'heure de fin du travail : 12h:45min par exemple, **14** en hexadécimal est **0x0E**, et **15** en hexadécimal est **0x0F**.

Et elles permettent de calculer les heures travaillées et d'afficher les résultats.

powerup;

0x00 0xA4 0x04 0x00 0xb 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x00 0x00 0x7F;

0xB0 **0x02** 0x00 0x00 0x00 0x00; // (**AFFICHE\_HEURES**)

//exemple de l’heure de début de travail: 09h:45min

0xB0 **0x05** 0x00 0x00 0x01 **0x09** 0x01; // (**START\_TIME\_H**) (**09h**:)

0xB0 **0x06** 0x00 0x00 0x01 **0x2D** 0x01; // (**START\_TIME\_MIN**) (:**45min**)

//exemple de l’heure de fin de travail : 14h:15min

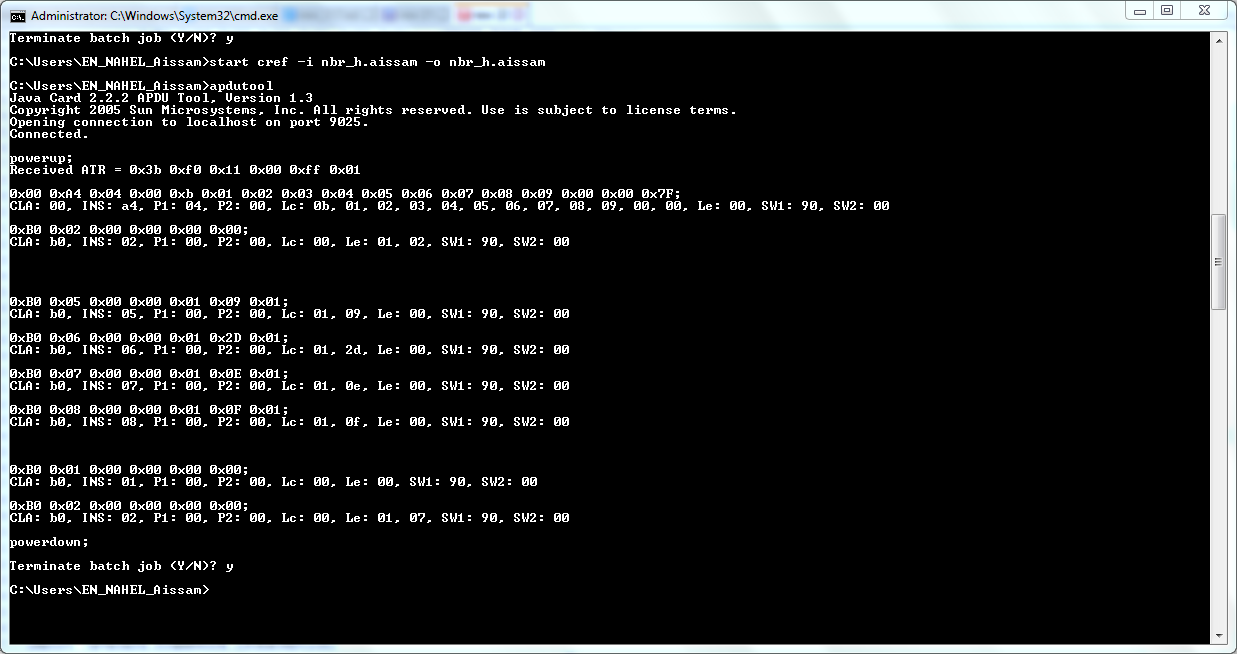
0xB0 **0x07** 0x00 0x00 0x01 **0x0E** 0x01; // (**STOP\_TIME\_H**) (**14h**:)

0xB0 **0x08** 0x00 0x00 0x01 **0x0F** 0x01; // (**STOP\_TIME\_MIN**) (:**15min**)

0xB0 **0x01** 0x00 0x00 0x00 0x00; // (**CALCUL\_HEURES**)

0xB0 **0x02** 0x00 0x00 0x00 0x00; // (**AFFICHE\_HEURES**)

powerdown;



7 heures travaillées

2 heures travaillées

Pour vérifier le nombre d'heures travaillées

start cref -i nbr\_h.aissam -o nbr\_h.aissam

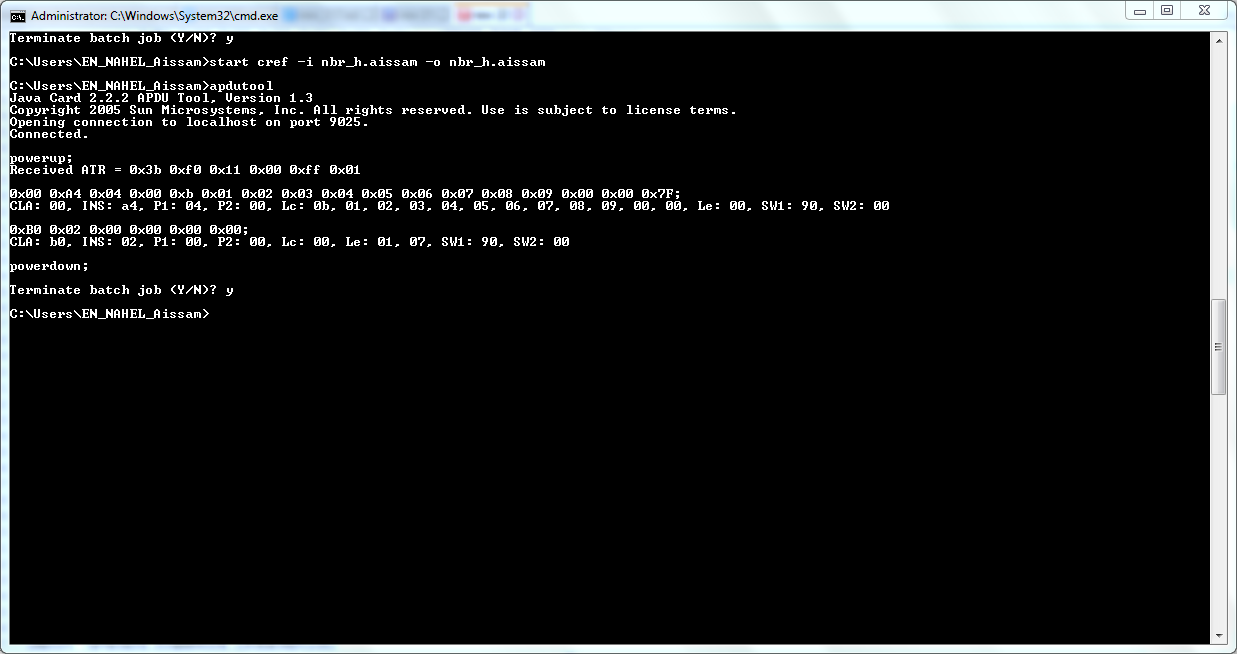
apdutool

powerup;

0x00 0xA4 0x04 0x00 0xb 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x00 0x00 0x7F; // (Sélectionner l’Applet)

0xB0 **0x02** 0x00 0x00 0x00 0x00; // (**AFFICHE\_HEURES**)

powerdown;



7 heures travaillées