

Outils pour l'intégration et/ou le développement

Cette page présente l'usage des outils de développement dans le cadre des projets ESUP-SGC / ESUP-NFC-TAG. Ces outils peuvent être utilisés au moment de l'intégration de la solution ainsi que pour le développement de nouvelles interfaces.

Lors de l'intégration du projet il est intéressant d'importer les sources dans un outils de développement pour bénéficier de ses fonctionnalités (auto-complétion, coloration syntaxique, indentation...). Par ailleurs cela peut permettre d'implémenter de nouveaux services ex :

- Esup-SGC : CardIdsServices (gestion d'identifiants), ExtUserInfoService (Interfaces d'import des données utilisateurs), Export2AccessControlService (Exports pour les contrôles d'accès)
- Esup-Nfc-Tag : TagIdCheckApi (contrôle des identifiants de cartes), AppliExtApi (Interfaces)
- SpringToolSuite
 - Pré-requis
 - Importer le projet
 - Editer la configuration
 - Versionning de la configuration
 - Compilation / lancement
- Android Studio
 - Pré-requis
 - Importer le projet
 - Editer la configuration
 - Compilation
 - Pré-requis
 - Importer le projet
 - Compilation

SpringToolSuite

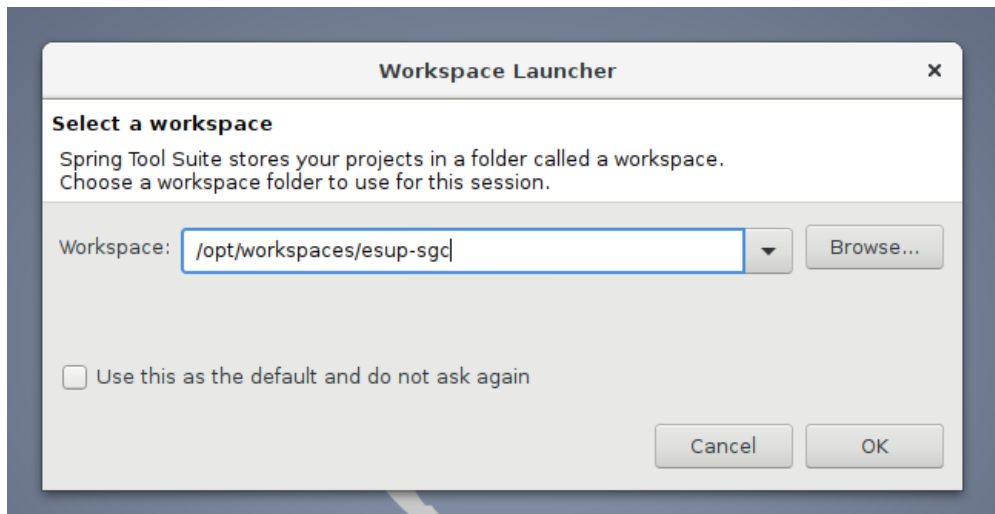
Esup-SGC et Esup-Nfc-Tag-Server sont développés à l'aide de SpringToolSuite STS (basé sur Eclipse) : <https://spring.io/tools>

Pré-requis

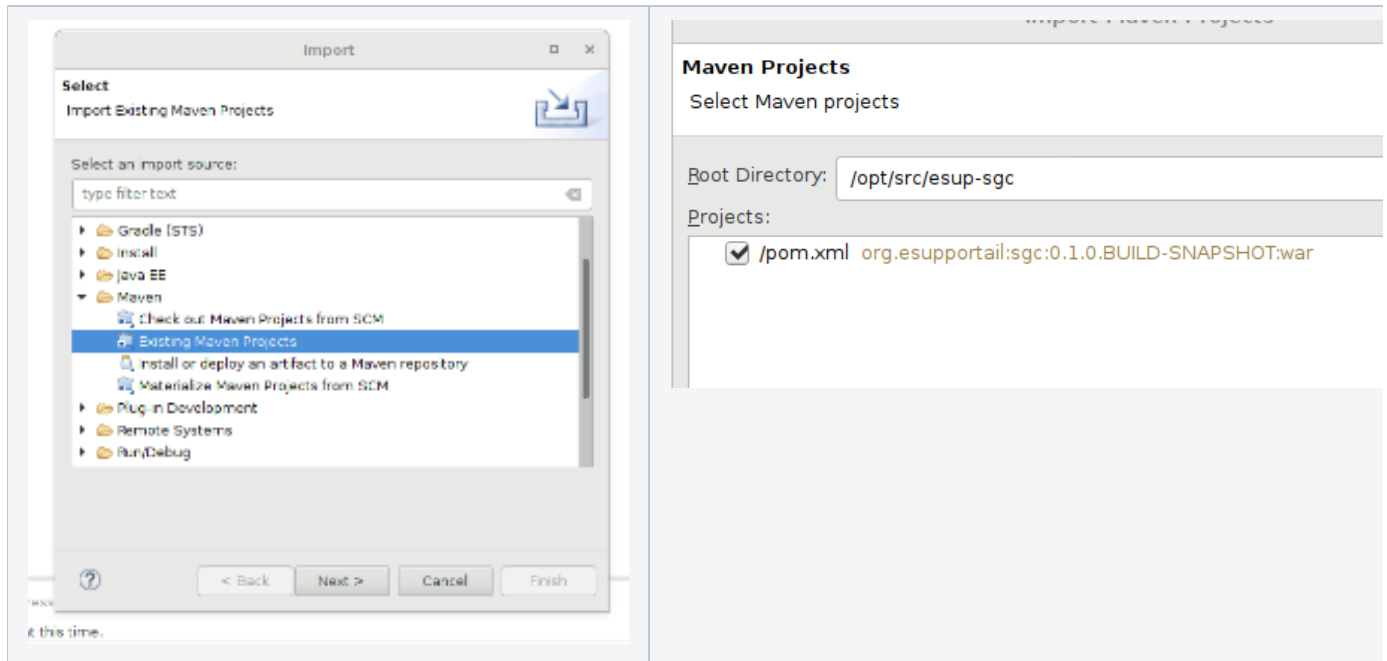
- Java JDK 8
- Maven

Importer le projet

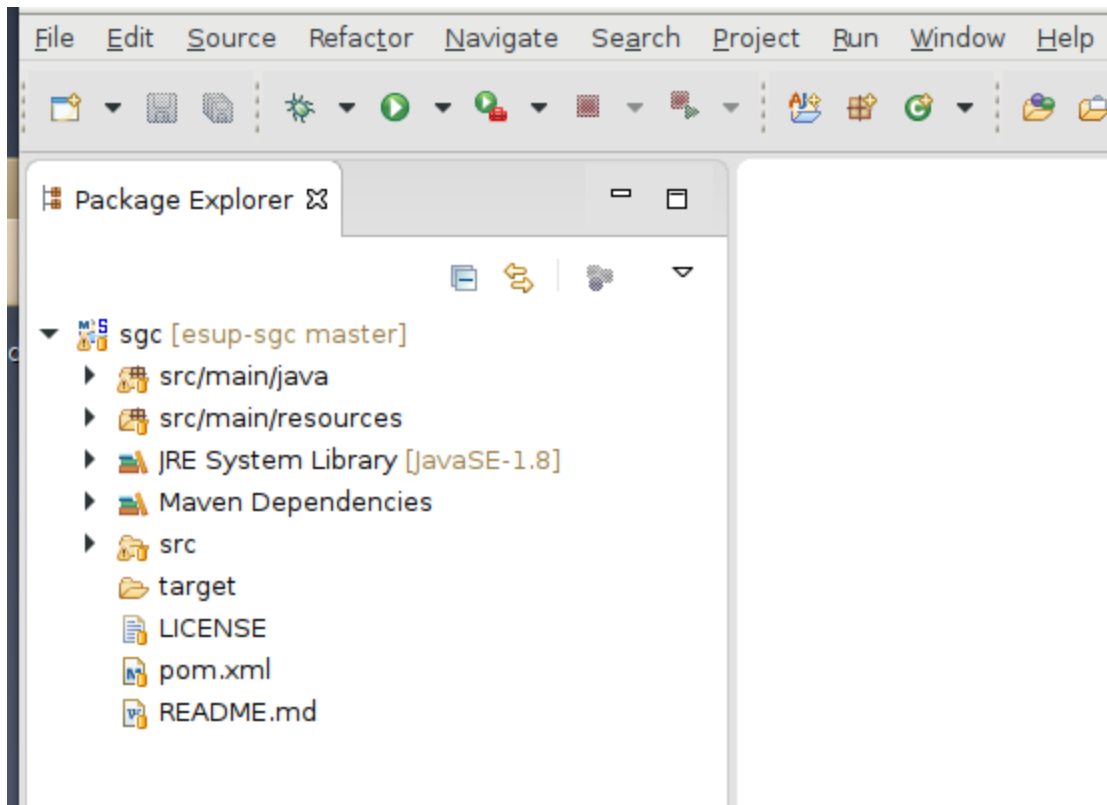
Après avoir cloné le projet (dans /opt/esup-sgc), démarrez STS et créez un nouveau workspace dans /opt/workspaces/esup-sgc par exemple



Lorsque STS est démarré faites "File -> Import" puis choisissez "Maven -> Existing Maven Projects" et naviguez dans le dossier des sources où se trouve le pom.xml



Cliquez sur "Finish", le projet s'importe puis vous devez obtenir l'arborescence suivante :



Editer la configuration

Dans STS l'auto-complétion est déclenchée par la combinaison CTRL + Espace

L'auto-complétion va nous permettre de modifier facilement la configuration. On prend comme exemple le fichier src/main/resources/META-INF/spring/applicationContext-services.xml. Si l'on souhaite ajouter un service CardIdsService

A la saisie dans le nouveau bean au niveau du '<bean class="org.esupportail.sgc.services.cardid.' l'auto-complétion va proposer CnousCardIdService, ComueNuAccessControlCardIdService, GenericCardIdService ...

De même, pour la saisie des propriétés l'auto-complétion fonctionne et propose les noms des attributs de la classe sélectionnée.

Cette méthode permet de sécuriser l'édition des fichiers de configuration.

Versionning de la configuration

D'une manière générale il est conseillé de faire des commits GIT locaux pour sauvegarder vos modifications de configuration.

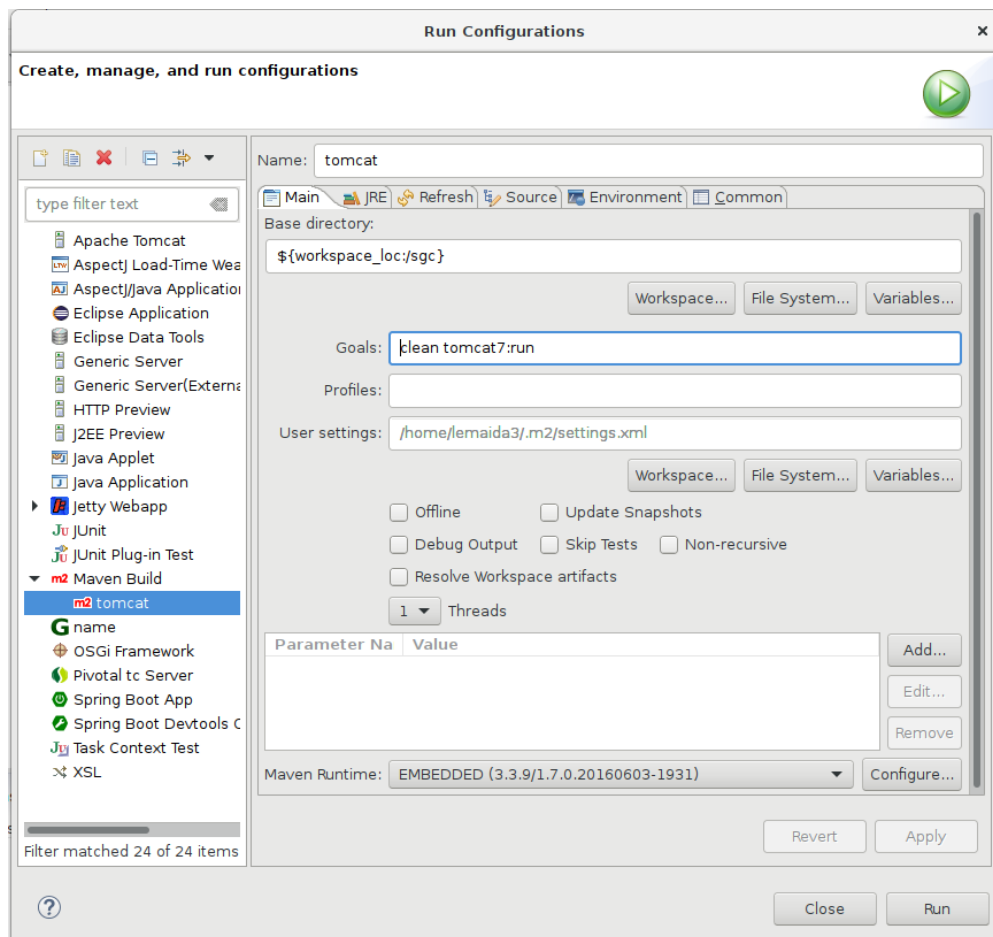
Lorsque vous avez fini la configuration du fichier vous pouvez exécuter les commandes suivantes depuis la racine de vos sources

```
git add src/main/resources/META-INF/spring/applicationContext-service.xml
git commit -m "config prod univ-ville-fr"
```

La modification du fichier applicationContext-service.xml sera versionnée ce qui permet de conserver les traces de toutes les modifications (procédure à faire à chaque modification). De plus lors d'une mise à jour d'ESUP-SGC (git pull) la configuration ne sera pas écrasée.

Compilation / lancement

Pour compiler ou lancer le projet il faut créer une configuration de lancement de type "Maven Build" dans "Run -> Run Configurations" comme suit :



Android Studio

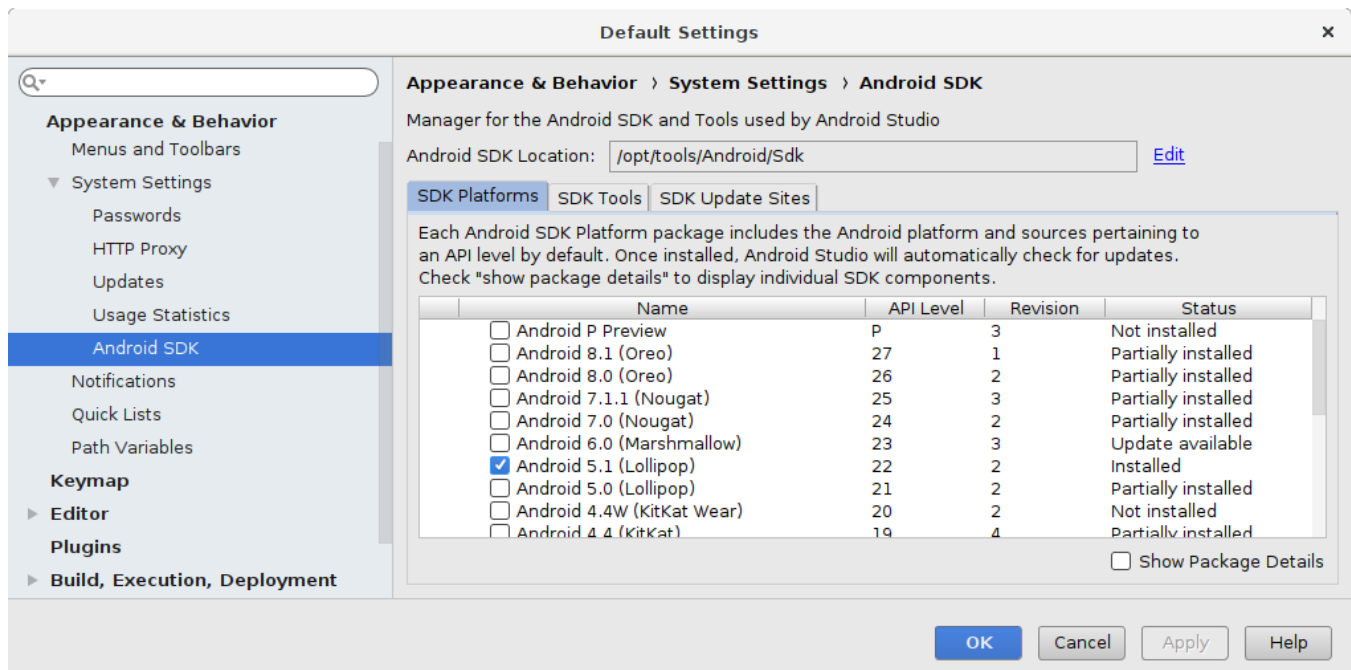
Esup-Nfc-Tag-Droid est développé sous Android Studio <https://developer.android.com/studio/index.html>

Pré-requis

- Java

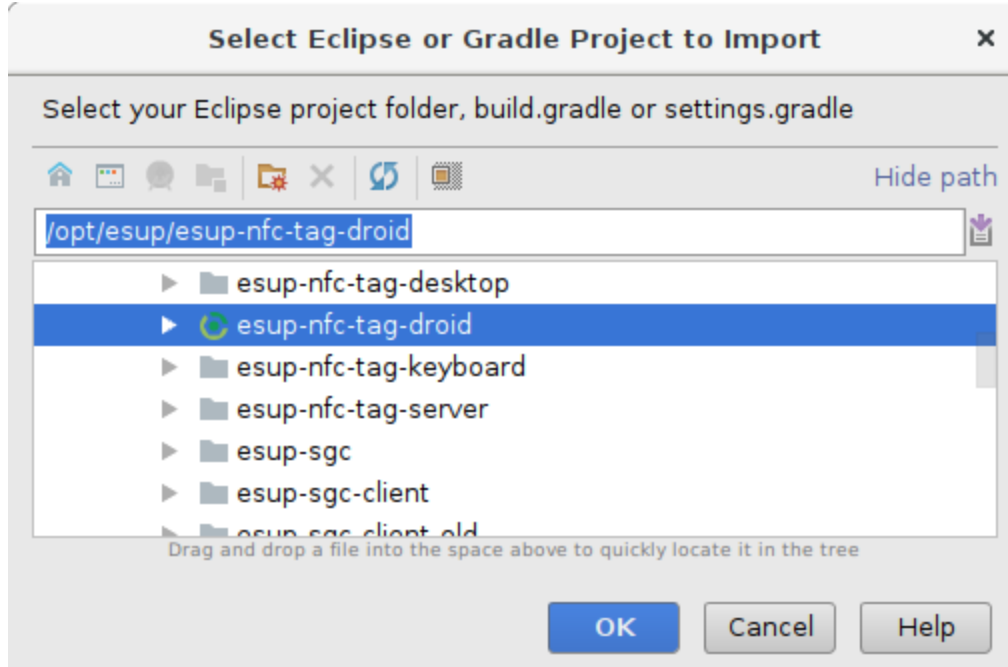
- Smartphone Android
- SDK 5.1

Pour vérifier l'installation de SDK, il faut ouvrir SDK Manager depuis le menu "Configure"



Importer le projet

Après avoir cloné le projet (dans /opt/esup-nfc-tag-droid), choisir "Import project"



Ouvrir le fichier build.gradle et contrôler que la synchro est bonne. Si cela n'est pas le cas cliquer sur "Sync Now" ou "Try Again" (le cas échéant fixer les problèmes Gradle)



Si la version de gradle n'est pas bonne l'ide propose de fixer le projet

The versions of the Android Gradle plugin and Gradle are not compatible.

Please do one of the following:

- Update your plugin to version 2.4. This will require changes to build.gradle due to API changes.

- Open migration guide

- Downgrade Gradle to version 3.5.

- Fix Gradle wrapper and re-import project

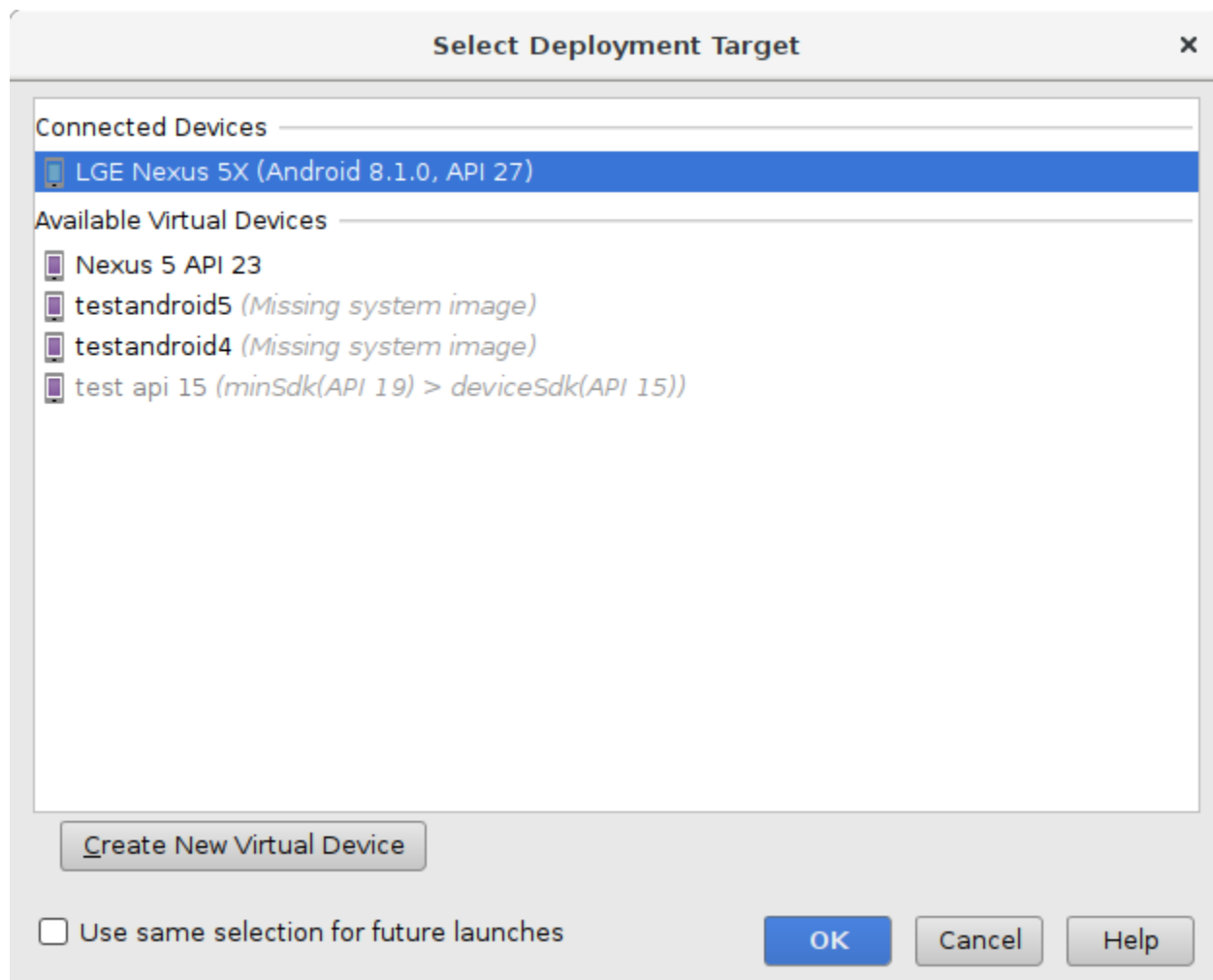
ou

Upgrade build tools

Lorsque le project compile correctement la barre d'outils présente le bouton d'execution



L'application peut etre lancée soit sur un emulateur Android paramétré depuis l'écran "Select deployment Targe" (voir dans "Create New Virtual Device") soit sur un téléphone Android avec degobage USB activé



Editer la configuration

Modifier l'adresse d'Esup-Nfc-Tag-Server dans esupnfc.tag.properties

```
esupNfcTagServerUrl=https://esup-nfc-tag.univ-ville.fr
```

Compilation

Cliquer sur "Build -> Generate Signed APK" puis renseigner le keystore

Visual Studio

Esup-CNOUS-Client est développé sous Visual Studio Community <https://www.visualstudio.com/fr/>

Pré-requis

- Windows
- Framework .net 2.0 minimum

Importer le projet

Après avoir cloné le projet (dans c:\opt\esup-nfc-tag-droid), faire "Fichier -> Ouvrir un projet" puis choisir le fichier sln

Compilation

Choisir "Release" au niveau de la barre d'outils" puis cliquer sur "Générer -> Générer la solution" (raccourci Ctrl+Maj+B)

Dans le dossier c:\opt\esup-crous-client\bin\Release on trouve l'executable CreationCarteCrous.exe ainsi que le fichier de configuration CreationCarteCrous.exe.config