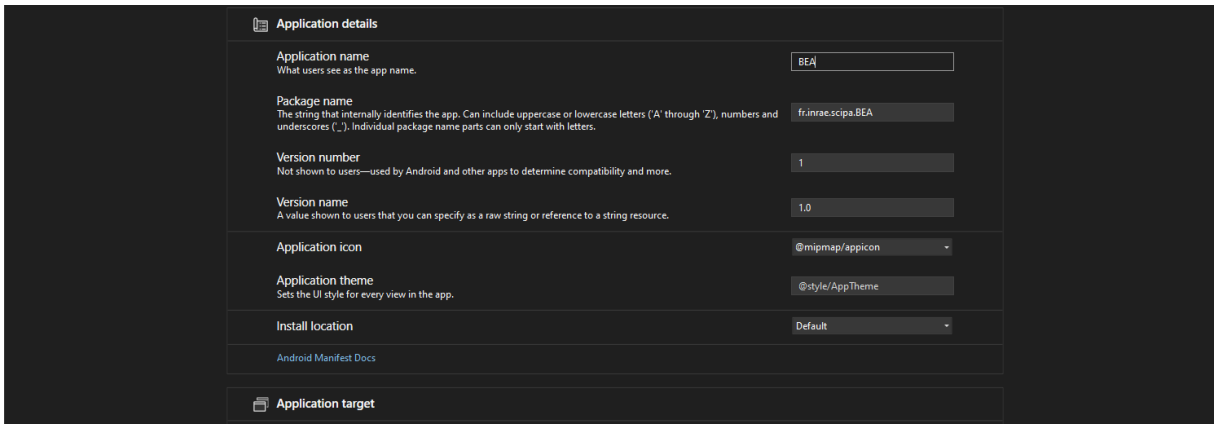
Déployer un projet .NET MAUI sur Android

Etape 1 : Nom de l'application, Nom du paquet, numéro de version et Icone de l'application

Passer en mode Release.

Sur Visual Studio 2022, ouvrir le fichier « AndroidManifest.xml » Soit via l'explorateur de solution, soit via la fenêtre de « propriété » du projet.



Modifier en définissant le nom de l'application, le nom du paquet

Le nom du paquet est le nom qu'aura notre paquet APK ainsi que le nom qui sera affiché dans informations détaillées de l'application. Pour nos applications, il conviendra de mettre : fr.inrae.sicpa.« Nom du programme »

Il est recommandé d'avoir une icône pour l'application. Les applications de distribution de type "marketplace" (ex. Google Play) empêchent de publier une application qui n'en a pas. La propriété Icon de l'attribut Application permet de définir une icône pour un projet Android.

Pour les icones, il est recommandé par l'équipe Android d'utiliser les dossiers mipmap fournis comprenant chacun deux fichiers : icon.png et launcher_foreground.png. Chaque dossier correspond à une densité d'écran (dpi). Heureusement, il n'est pas obligé de tout changer manuellement et il existe un outil mis à disposition par les développeurs Android. Plus d'informations sur le lien suivant :

https://developer.android.com/studio/write/image-asset-studio#create-legacy

Note : Le dossier mipmap-anydpi-v26 est utilisé pour avoir des icones adaptatives (au format XML) sur des appareils utilisant un niveau SDK supérieur a 26, ce qui ne nous concerne pas car nous visons un niveau d'APK 25.

Etape 2 : Version de l'application

Le versionnage est important pour la maintenance et la distribution des applications Android. Sans une certaine forme de gestion des versions, il est difficile de déterminer si et comment une application doit être mise à jour. Pour faciliter la gestion des versions, Android reconnaît deux types d'informations différentes :

Numéro de version : une valeur integer qui représente la version de l'application. Ce numéro s'incrémente de 1 à chaque nouvelle version et ne doit pas être diffusé aux utilisateurs. Il est

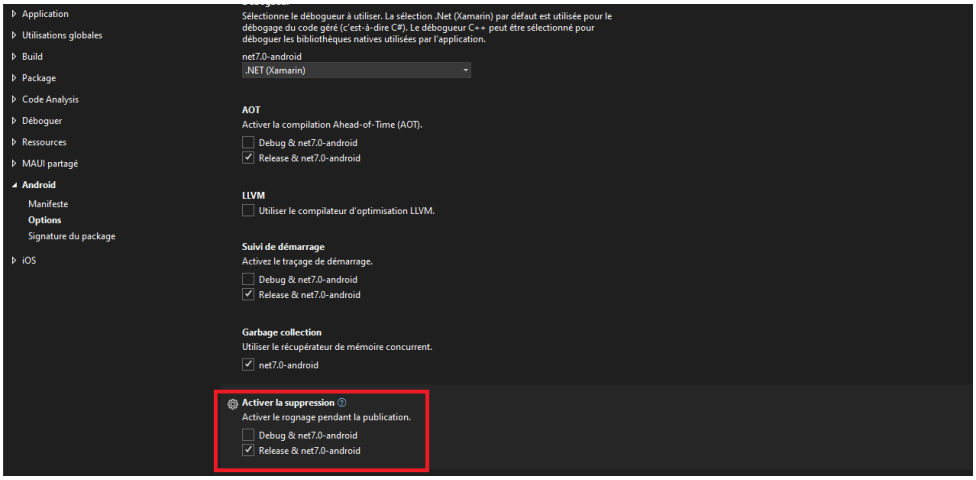
indépendant du Nom de version (ci-dessous). Cette valeur est utilisée en interne par Android pour gérer les conflits de version, par exemple.

Nom de version : un string qui est utilisé pour informer l'utilisateur de la version de l'application. C'est le nom qui sera affiché aux utilisateur (ex. "v1.2.0"). Cette valeur est purement indicative est n'est pas utilisée en interne par Android.

De la même façon que pour l'icône, on définit ces paramètres en allant dans les propriétés du projet (plus bas)

Etape 3 : Optimisation de la taille de l'APK

Optimiser la taille de l'APK, en laissant « Activer la suppression » sur Release

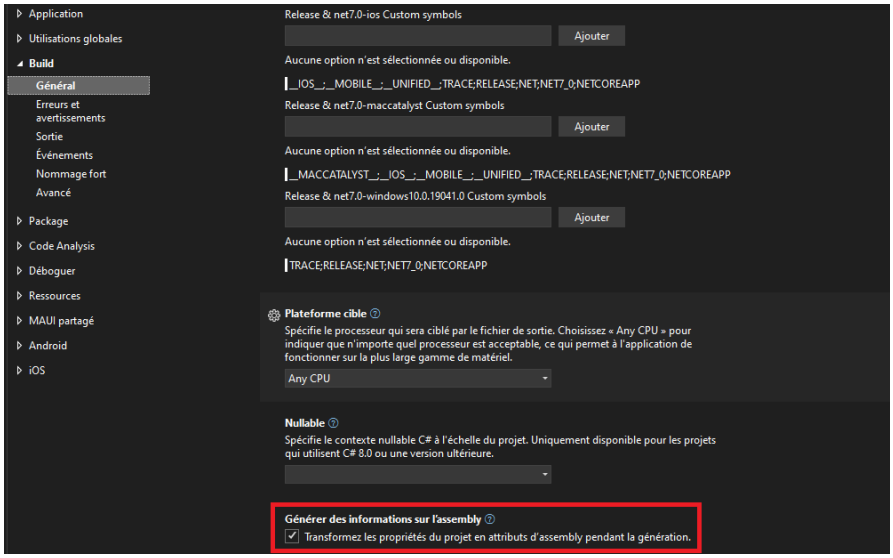


Source : https://learn.microsoft.com/fr-fr/dotnet/maui/android/linking?tabs=vs

Etape 4 : Protéger l'application

Il est important de désactiver toujours l'état de débogage dans une application publiée car il est possible (via JDWP) d'obtenir un accès complet au processus Java et d'exécuter du code arbitraire dans le contexte de l'application si cet état de débogage n'est pas désactivé.

Vérifier que la génération des informations sur l'assembly est cochée dans les propriétés du projet (voir image ci-dessous).



Pour générer le fichier « Nom du projet ». AssembyInfo.cs

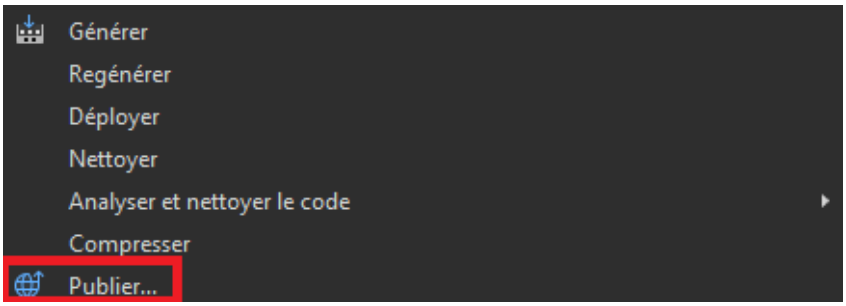
Exemple : BEA.AssembyInfo.cs

Chercher le fichier dans « Nom du projet » « /obj /Release » pour désactiver l'état de débogage en dans un build « Release » avec l'instruction ci-dessous.

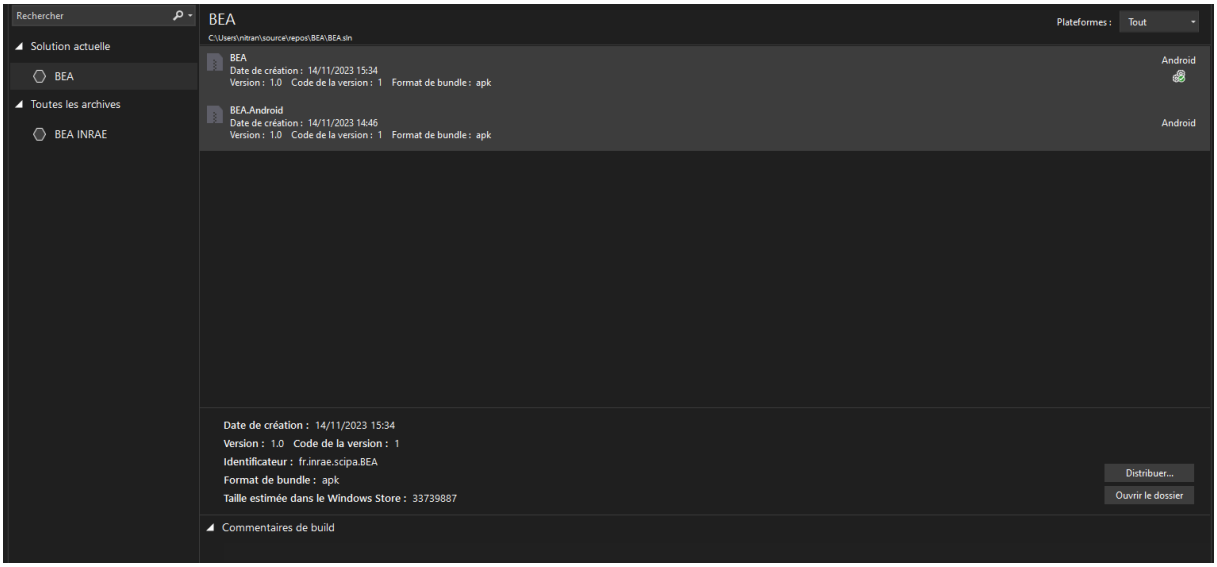
```
#if DEBUG
[assembly: Application(Debuggable=true)]
#else
[assembly: Application(Debuggable=false)]
#endif
```

Etape 6 : Archiver pour distribution

On accède au processus de distribution en cliquant droit sur le projet Android dans l'explorateur de solutions et en sélectionnant "Publier…" :



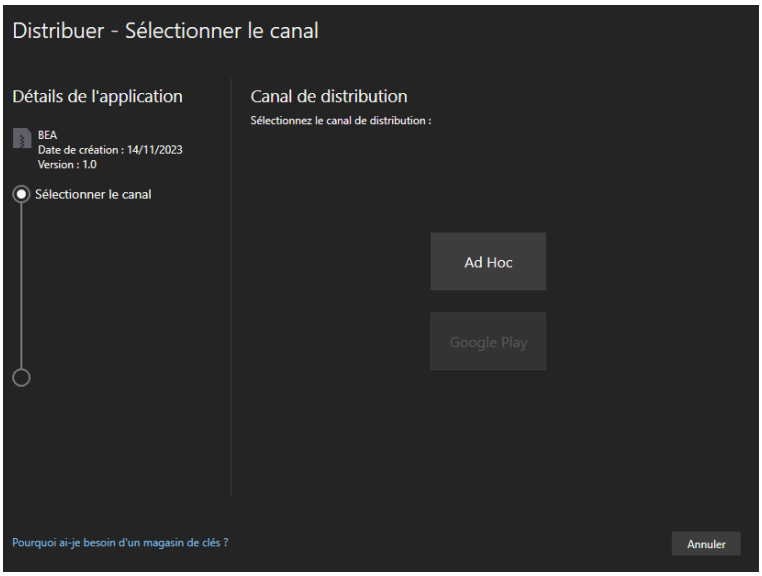
"Publier…" lance **le Gestionnaire d'archives** et commence le processus d'archivage du bundle d'applications



Etape 7 : Distribution

Lorsqu'une version archivée de l'application est prête à être publiée, sélectionnez l'archive dans le gestionnaire d'archives et cliquez sur le bouton "Distribuer…" :

La boîte de dialogue Canal de distribution présente des informations sur l'application, une indication de la progression du workflow de distribution et un choix de canaux de distribution. Lors de la première exécution, deux choix sont présentés :

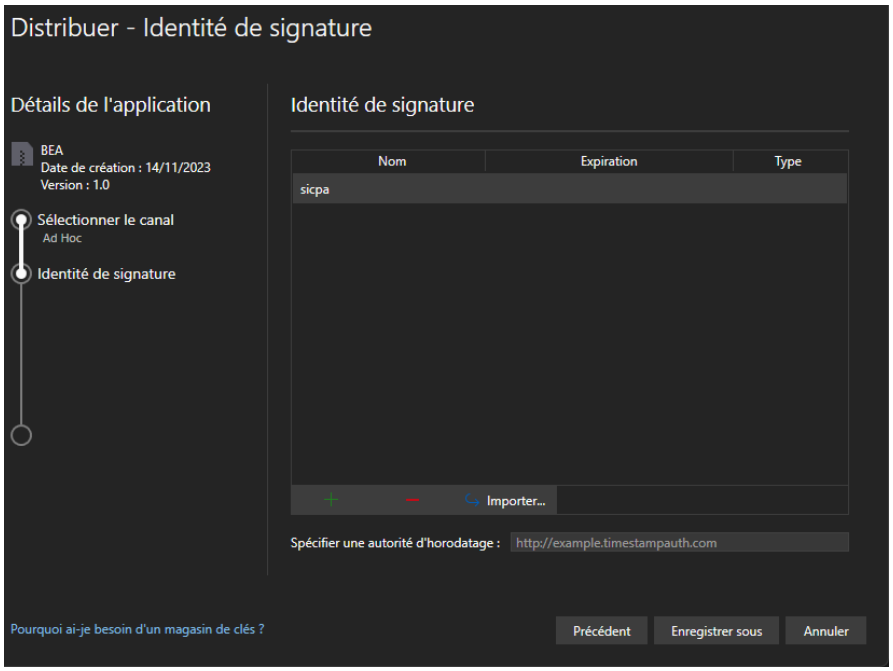


Ad-Hoc : Enregistre sur le disque un APK signé qui peut être chargé sur des appareils Android. C'est un bon moyen de créer un APK pour les tests. On utilisera cette option car on ne souhaite pas passer par Google Play.

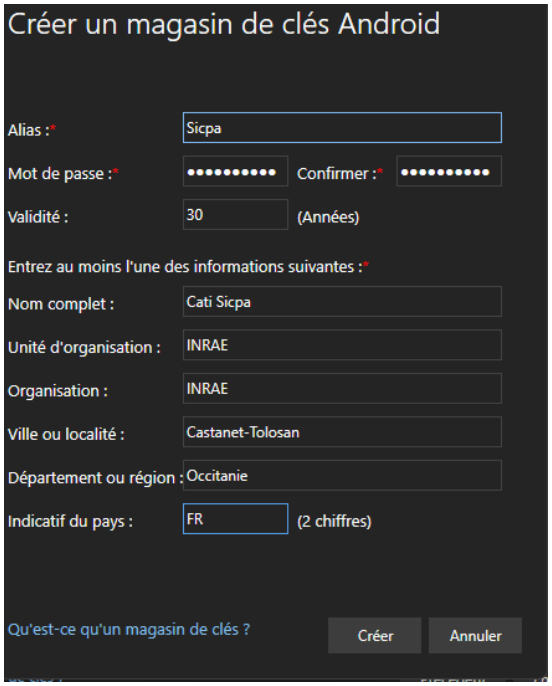
Google Play : Publie un APK signé sur Google Play

Etape 8 : Signer le paquet d'application Android

En utilisant la méthode ad hoc, l'APK résultant peut être chargé dans les appareils Android sans passer par une application de distribution. En sélectionnant l'option "Ad Hoc", Visual Studio ouvre la page Identité de Signature de la boîte de dialogue, comme le montre la capture d'écran suivante. Pour publier le .APK, il doit d'abord être signé avec une clé de signature (également appelée certificat).



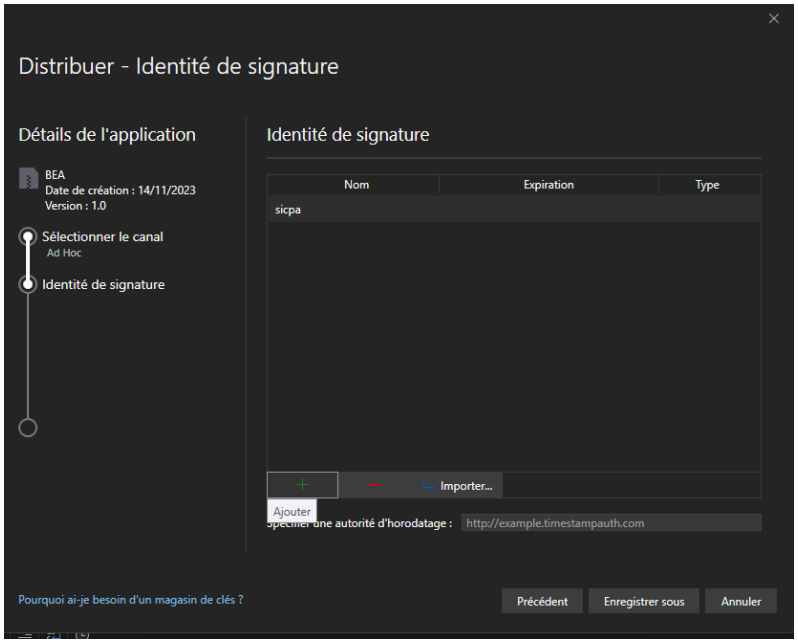
Un certificat existant peut être utilisé en cliquant sur le bouton Importer puis en procédant à la signature de l'APK. Sinon, cliquez sur le bouton + pour créer un nouveau certificat. La boîte de dialogue Créer un magasin de clés Android s'affiche ; utilisez cette boîte de dialogue pour créer un nouveau certificat de signature qui pourra être utilisé pour signer des applications Android. Saisissez les informations requises comme l'indique l'exemple suivant :



Le keystore obtenu se trouve à l'emplacement suivant :

C:\Users\USERNAME\AppData\Local\Xamarin\Mono for Android\Keystore\ALIAS\ALIAS.keystore

Lorsque vous cliquez sur Créer, un nouveau magasin de clés (contenant un nouveau certificat) est enregistré et répertorié sous Identité de signature, comme le montre la capture d'écran suivante. Pour publier ad hoc, sélectionnez l'identité de signature à utiliser pour la signature et cliquez sur Enregistrer sous pour publier l'application pour une distribution indépendante.



Ensuite, le gestionnaire d'archives affiche la progression de la publication. Lorsque le processus de publication est terminé, la boîte de dialogue Enregistrer sous s'ouvre pour demander l'emplacement où le fichier .APK généré doit être stocké.

Source : https://learn.microsoft.com/fr-fr/dotnet/maui/android/deployment/publish-ad-hoc