Een applicatie voorbereiden voor release

# Applicatie icoon

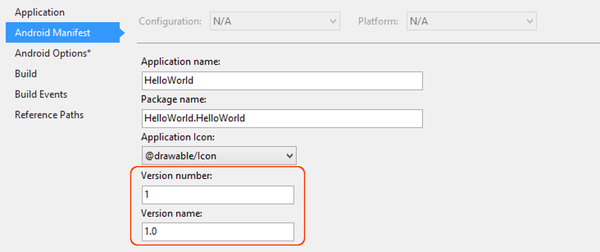
In visual studio versie 2013 en later, kan de applicatie icoon ingesteld worden door naar de project properties en daarna naar de sectie ’Andriod Manifest’ te gaan. In Andriod Manifest zie je een veld met Application Icon.

# 01-application-icon-sml

# Versie toevoegen

Een versie toevoegen is belangerijk om de applicatie te onderhouden. Android herkent twee verschillende types:

**Version Number:** Dit is de versie nummer

**Version Name:** Dit is de naam voor de versie, de versienaam word ook weergegeven in de Playstore.

# De applicatie beschermen

**Disable debugging**

Tijdens de development van de applicatie heb je debuggen nodig om verschillende technologieën met elkaar te laten communiceren, maar als je een applicatie wil releasen is dit een security threat.

Dit doe je makkelijk door deze conditional statement aan AssemblyInfo.cs toe te voegen.

#if DEBUG

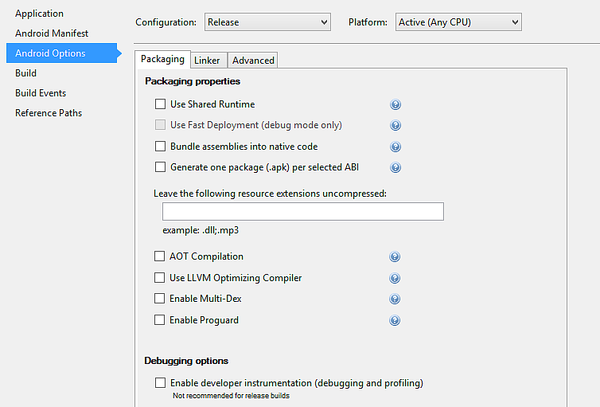
[assembly: Application(Debuggable=true)]

#else

[assembly: Application(Debuggable=false)]

#endif

# De properties voor de apk instellen

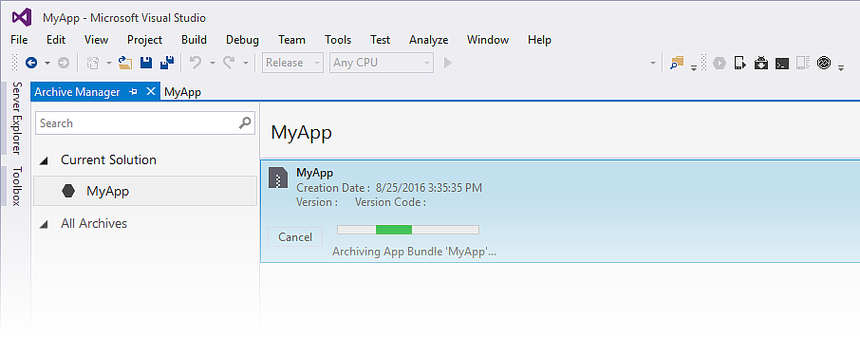
Om de properties in te stellen ga je in visual studio naar de project properties en daarna naar ‘Android Options’

# Compileren

Als je alle voorgaande stappen voltooid hebt kun je gaan compileren. Ga naar Build > Rebuild Solution om te verifieren dat de applicatie succesvol build in Release mode. (Deze stap produceert nog geen APK)

# Archiveren voor Publishing

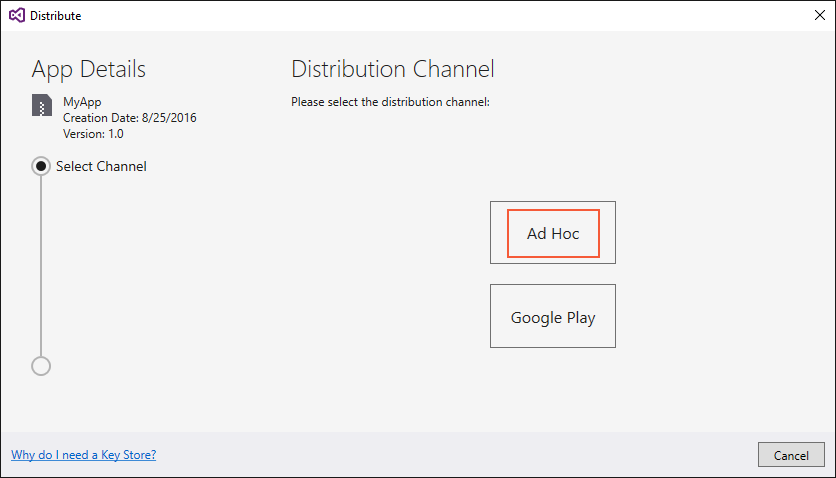
Om het publishing proces te beginnen moet je rechts klikken in de Solution Explorer > Archive.

Nu komt er een nieuw scherm, dit is de Archive Manager

# Signing

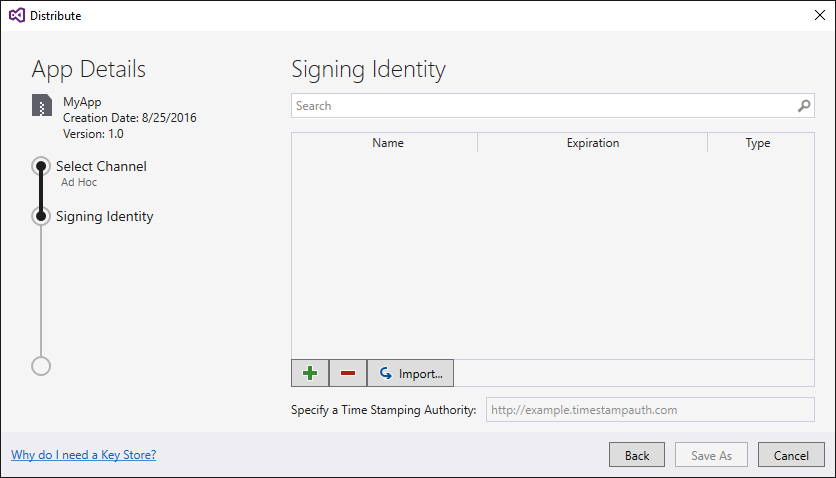
Het volgende is een uitleg van hoe je een “Android signing Identity” aanmaakt, een nieuwe signing certificaat voor android applicaties maakt en hoe je de gearchiveerde app ad hoc op je schijf.

In the “Archive ” selecteer je de “Distribution Channel” daarin select je Ad-Hoc

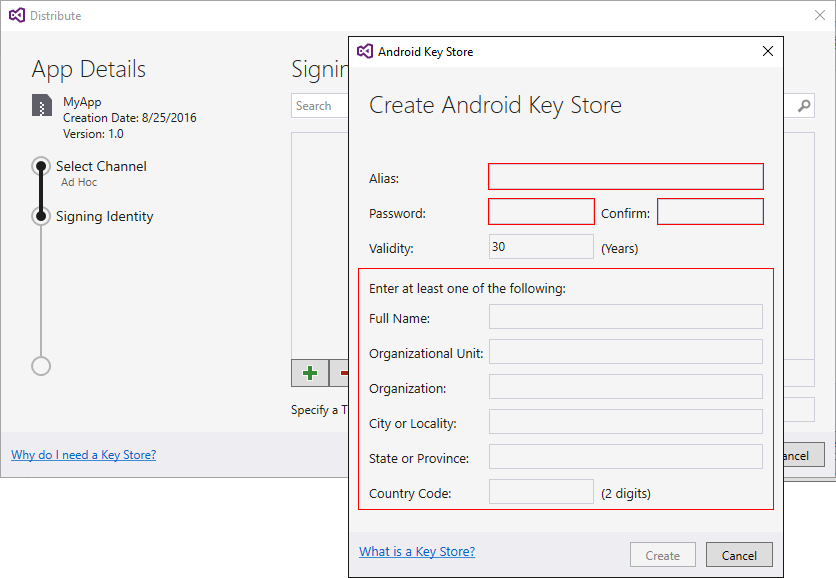


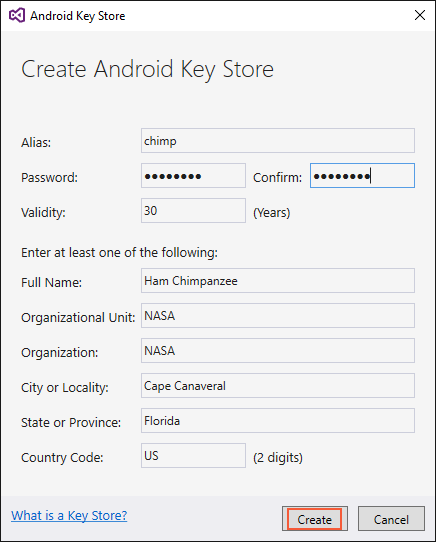
# Nieuwe Certificaat Aanmaken

Nadat Ad-Hoc is geselecteerd opent Visual Studio de “Signing Identity” pagina van het dialoog (Te zien in de screenshot). Om de .Apk bestand te publiceren, moet het eerst gesigneerd zijn met een signing key (ook wel certificaat).

Een bestaande certifcaat kan gebruikt worden door op de knop **Import** te klikken en dan doorgaan met het signeren van de APK. Heb je geen bestaande certificaat klik dan op de “+” knop om een nieuwe certificaat te maken

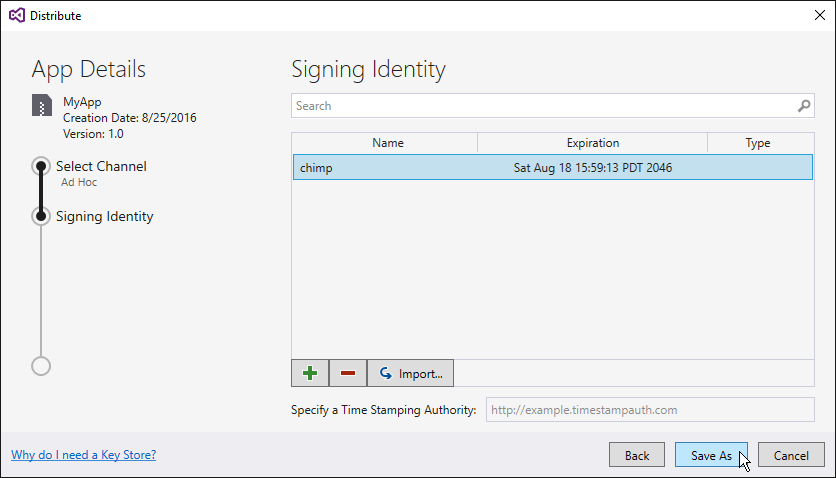
De “Create Android Key Store” wordt nu weergeven; gebruik deze dialog voor het maken van een nieuwe signing certificaat die gebruikt kan worden voor je Android applicaties. Voor de benodigde

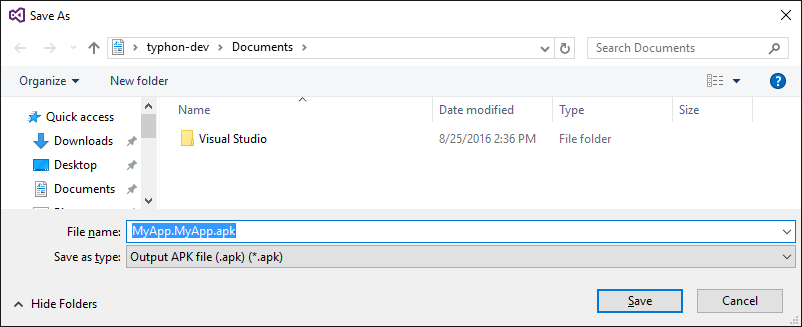
informatie in ( te zien in rood in de onderstaand afbeelding:Het volgende illustreert het soort informatie dat gegeven moet worden. Click “Create” om de nieuwe certificaat aan te maken

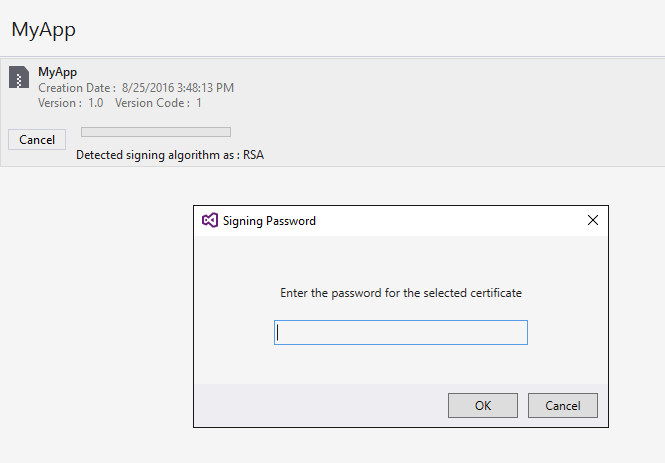


De keystore wordt opgeslagen in de volgende locatie:

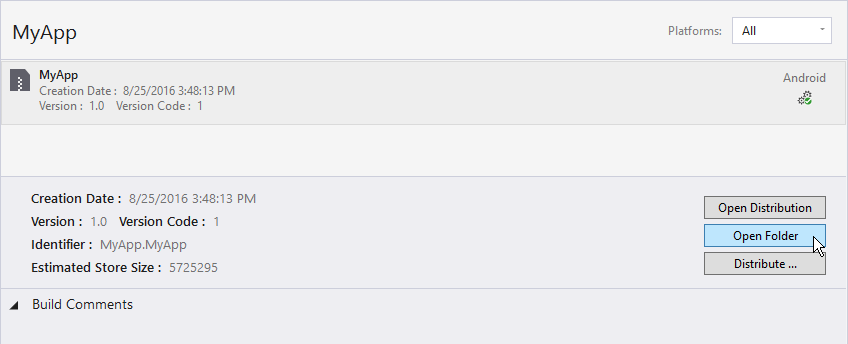
**C:\Users\USERNAME\AppData\Local\Xamarin\Mono for Android\alias\alias.keystore**

Nadat de knop “Create ” is aangeklikt wordt er een nieuwe key store opgeslagen en te zien onder “Signing Identity” zoals te zien is in de volgende screenshot. Om de ad-hoc te publishen, selecteer je de nieuwe signing identity om te gebruiken voor het signen en click je “Save As” om die te publisheren. 

Als volgende toont de “Archive Manager” de publishing progressie. Wanneer het publishing process voldaan is, opent de ”Save As” dialoog en vraagt voor een locatie om de gegenereerde .APK bestand opteslaan.

Navigeer naar de gewilde locatie en klik op opslaan. Als de key password onbekend is zal de “Signing password” dialoog voor komen, deze laat je een wachtwoord voor het certificaat selecteren

Klik vervolgens op “Open Folder”



Dit zorgt ervoor dat de map opent met het APK bestand erin. Het bestand is nu gesigneerd en gereed voor distrubutie.