

How we reduced Naukri iOS App size by 50% using App Thinning

With iOS 9, Apple has introduced the “App Thinning” feature which will save the device memory and downloading time for user. In previous version, developer were supposed to make universal build which includes all the assets of the devices. So, when application installed on the devices, it has all the resources regardless of the device specifications. With App Thinning, only required assets will be downloaded, that are required to run the application on the particular iOS device.

In countries like India, where majorly people doesn't have high speed internet, have limited data or concerned regarding the data consumption. App thinning will be beneficial for application and its users because it will reduce the data consumption, reduce the time and most important reduce the memory. So we have implemented “App Thinning” in our “Naukri India” Application to focus on those people those have limited Data and memory.

There are three main aspects of App Thinning:-

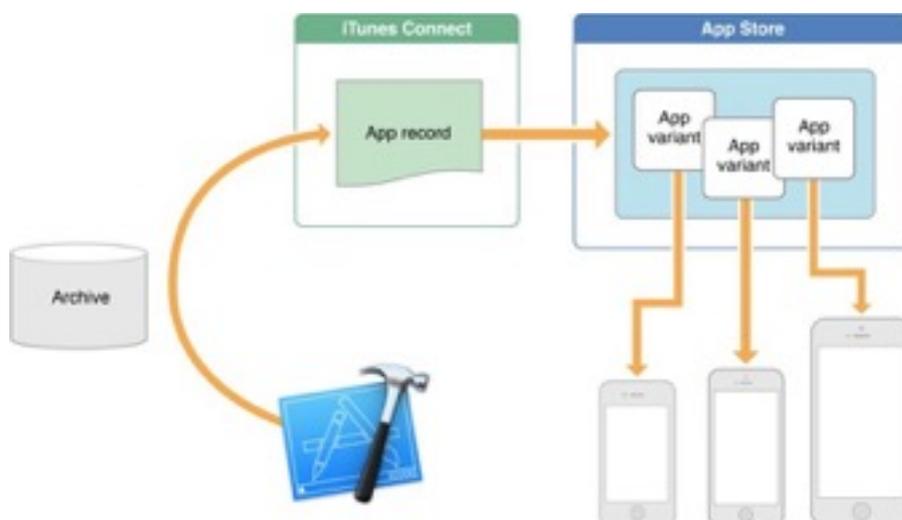
1. Assets slicing
2. On Demand Resources
3. Bit Code

App Slicing

According to Apple,

“Slicing is the process of creating and delivering variants of the app bundle for different target devices.”

App Slicing will provide only required assets to device (depending upon device configuration). Developer still have to upload the universal build (.ipa) files over the iTunes.



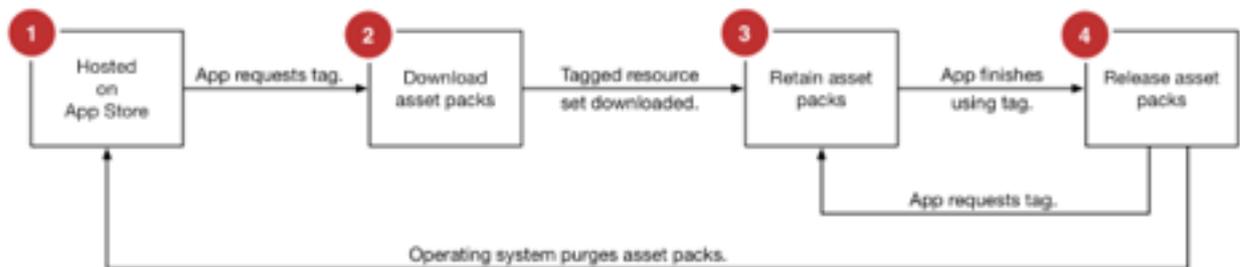
Source: <http://www.appcoda.com/app-thinning/>

AppStore will slice the application according the user’s device (depending upon the device configuration).

For example, User have a iPhone 6 Plus and download the application from Apple Store. Apple store will give the only iPhone 6 plus variant. So user will not get @2x assets, iPad Assets or iPhone 4s assets. So User will get smaller build and will download in less time & consume less data.

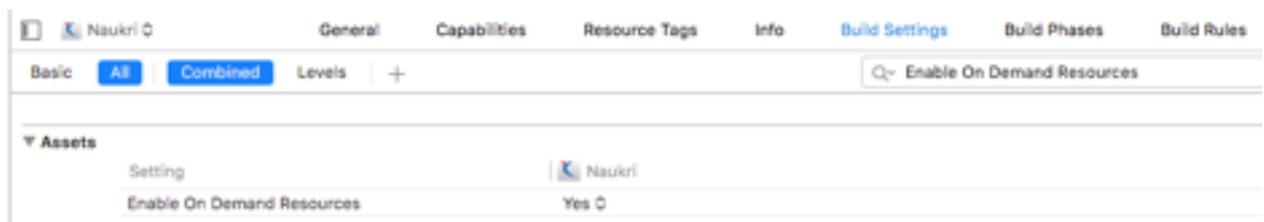
On Demand Resources

On-Demand Resources are files that are downloaded when they are needed and that are also cleared from devices when their purpose fulfilled.



Source: <http://arstechnica.com/apple/2015/06/app-thinning-will-be-a-major-boon-for-8gb-and-16gb-iphones-and-ipads/>

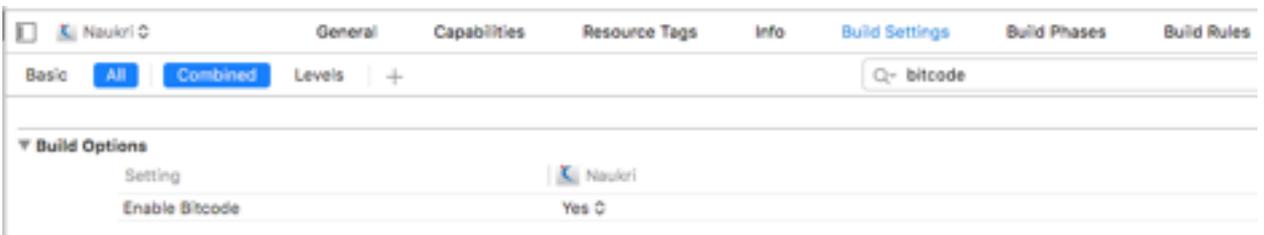
We can enable the On Demand Resources by changing the “Enable On Demand Resources” boolean to Yes under the Xcode Settings (Build Settings).



Bitcode

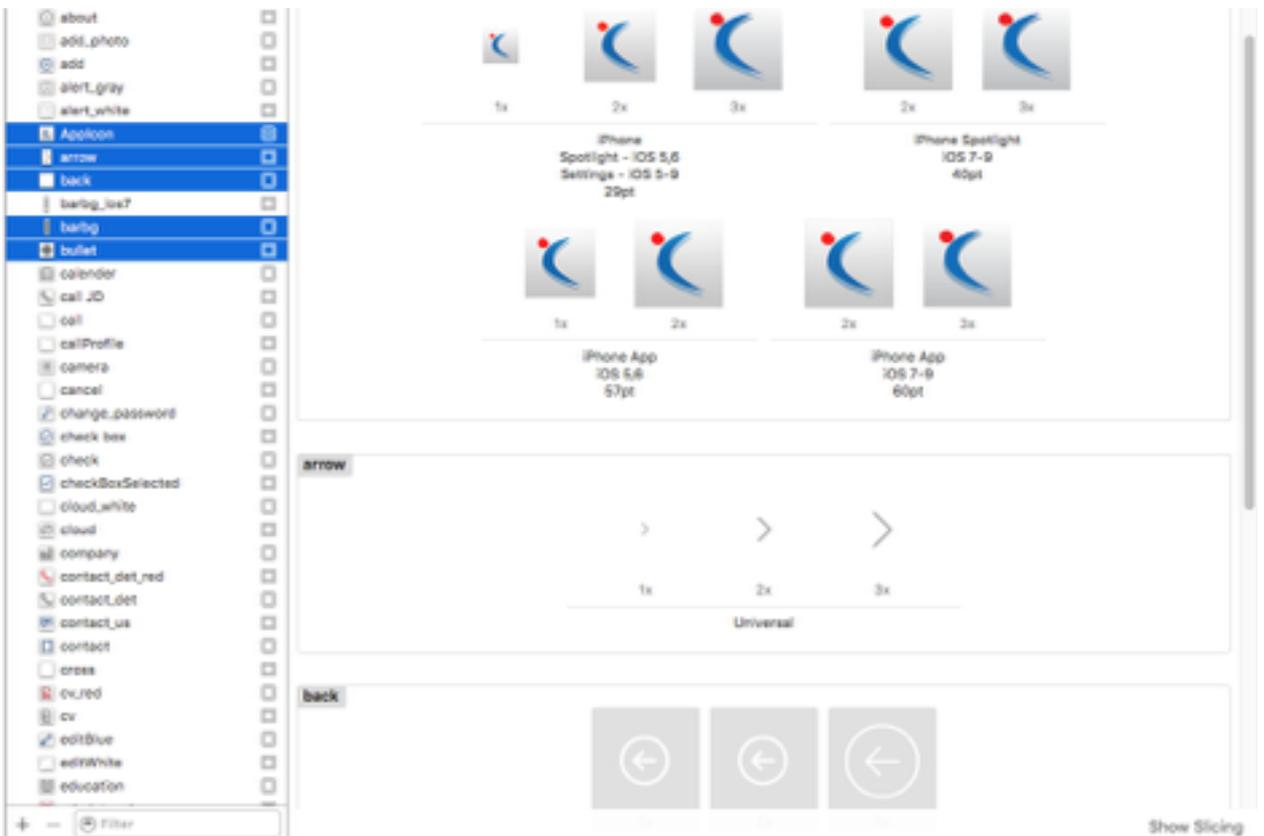
Bitcode optimizes the application before the application download. It optimizes the application according to the devices configuration and architecture. Bitcode make the downloadable build smaller by eliminating unnecessary optimizations and download only relevant optimization. Using Bitcode is enabled by default but still optional for iOS apps (expect it to become required in the future, though). It is, however, required for native Apple Watch apps.

We can switch on the BitCode Feature in our project by enabling the “Enable BitCode” boolean to Yes under the Xcode Settings

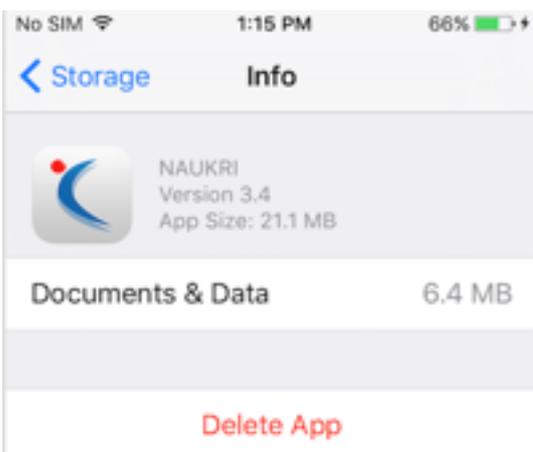


Testing with App Thinning

To implement the App Thinning, we must use the “**Asset catalogs**”. The Asset Catalog contains 1x, 2x, and 3x versions of the app icon and other images also. Assets Catalogs will look like:-

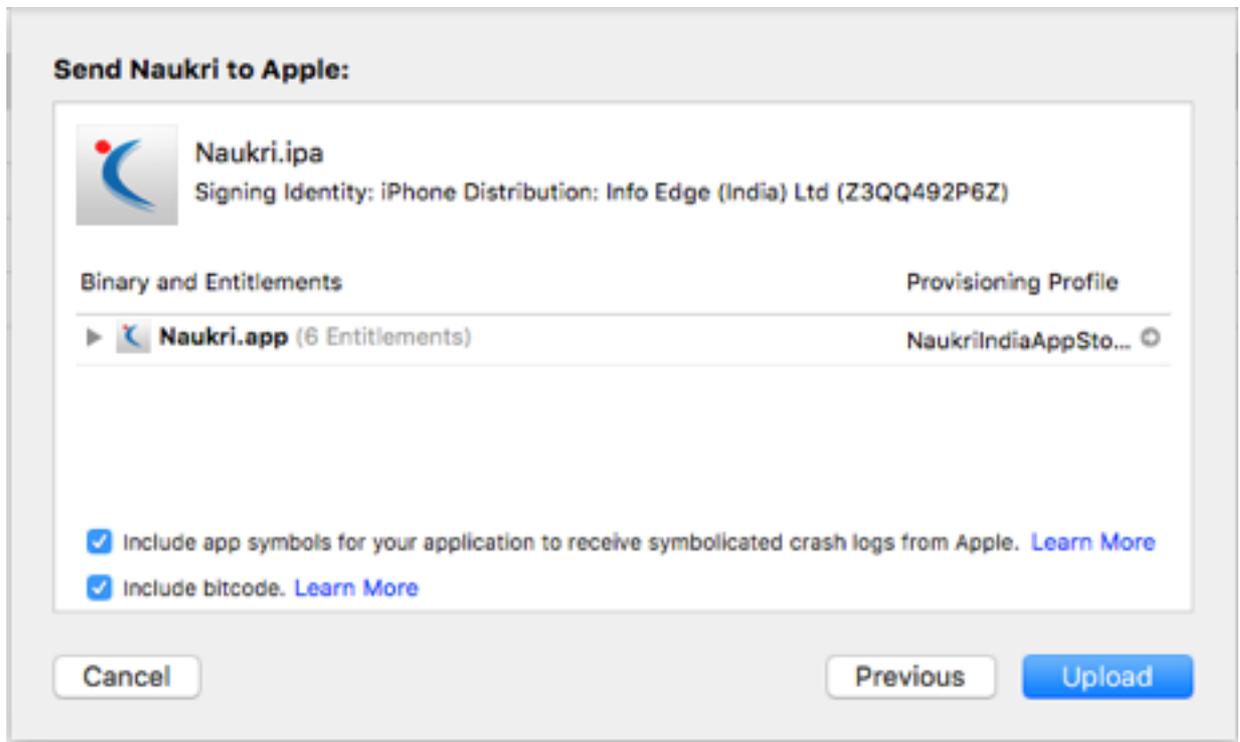


First We run the application over the device, see the memory taken by application. We search “Manager Storage” under the “Setting section”. We scroll down to notice the memory.

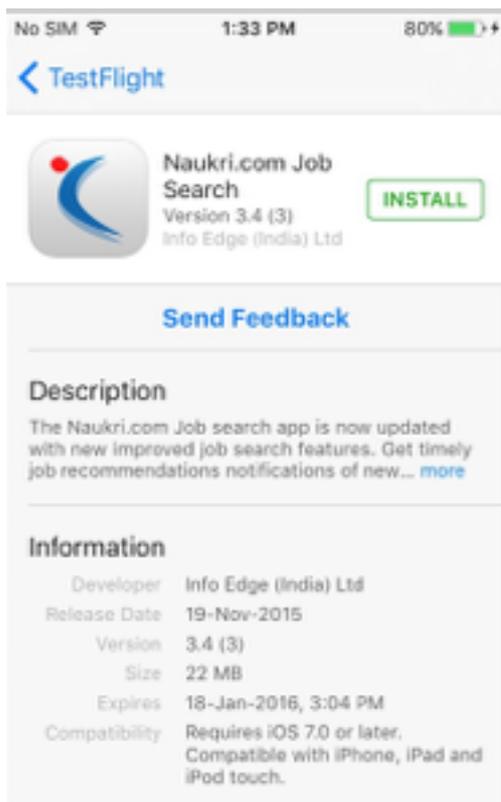


In our case, we run the application over the iPhone 5c, memory taken by application is 21.1 MB. This time, device have fat version of the application (Build that have all resource irrespective the device configuration). Xcode doesn't automatically handle the app variant and app thinning.

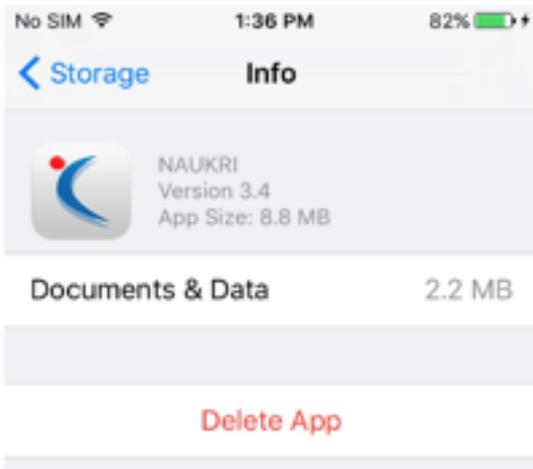
Next, we created the build of the application by clicking on the “Archive” option in “Product” Tab. While uploading the application over iTunes Store, we have to enable the boolean “Enable BitCode” to Yes.



After uploading the build over iTunes Store, we will have our new build version in list of the build over iTunes Connect. We select the new build and send the invitation through TestFlight.



After the getting the invitation, we can download the new version of application over the device through TestFlight.



We have installed the application and visited the “Manager Storage” and checked the application. Now its 8.8MB. This time, application have only device specific resources or relevant resources.

For iPhone 5c, App thinning saves around 12.3 MB (58%).

We ran same steps over various devices. We got following analysis data. On a average, App thinning saves 30-50% memory.

iOS Version & iPhone Device	App Size (Before App Thinning)	App Size (After App Thinning)
9.0 (iPhone 5c)	23.4 MB	8.8 MB
9.1 (iPhone 5s)	14.1 MB	9.5 MB
9.1 (iPhone 6)	19.1 MB	9.5 MB
9.1 (iPhone 6 Plus)	19.2 MB	9.6 MB

Effect of App Thinning over Naukri India iOS App

References: –

1. [Working with App Thinning in iOS 9](#)
2. [App thinning in iOS 9: Explained](#)