

App Thinning

- [Introduction](#)
- [Slicing](#)
- [Bitcode](#)
- [On-Demand Resources](#)

Introduction

In iOS, App Thinning helps optimize your application by using:

- **Slicing:** Use an Asset Catalog to install assets on specific devices
- **Bitcode:** Submit partially compiled code that will be optimized by the App Store
- **On-demand resources:** Specific assets are stored in the App Store and downloaded by the application when they are needed, and purged by iOS when disk space is low

The optimizations help reduce the size of the application when the user installs it and reduces the application's memory footprint.

Titanium SDK only supports Slicing. For details, see the subsections below.

For more information about App Thinning, see [iOS Developer Library: App Thinning](#).

Slicing

Slicing is used to indicate which assets are supported by which devices, so when a user installs your application, only the necessary assets are included. iOS uses an Asset Catalog to mark images for specific devices.

Icons and launch images are added to the Asset Catalog. Because of this change, you cannot access icons or launch images from the filesystem or an `UIImageView` object.

If enabled, the Titanium SDK automatically adds all PNG and JPEG images to an Asset Catalog. Images that match a specific pattern (e.g. suffix `@2x`, `@3x`, etc.) will be grouped into one imageset. If there is no matching suffix, the image would be in an imageset by itself. For details, see [Images and UIImageView APIs](#). Because of this change, you cannot access these images from the filesystem, that is, you cannot access the file using its path or URL with the `Ti.FileSystem` API or equivalent.

By default, slicing is disabled. To enable slicing, add the `use-app-thinning` element to the `ios` element in the `tiapp.xml` file and set it to `true`.

tiapp.xml

```
<ti:app>
  <ios>
    <use-app-thinning>true</use-app-thinning>
  </ios>
</ti:app>
```

Bitcode

Bitcode produces an intermediate compiled version of your application which is submitted to iTunes Connect. The App Store will optimize and compile your application for specific architectures and future supported architectures. Using Bitcode, reduces the size of the package sent to iTunes Connect and provides future support for new architectures without having to re-submit your application.

Currently, bitcode is disabled for Titanium because all application frameworks require bitcode to be enabled, including third-party modules.

On-Demand Resources

On-demand resources are resources, such as large data and sound files, that are tagged by you into groups to be downloaded when the application needs them. The resources are stored and managed by the Apple servers. When the application needs a resource group, it will download it from the Apple servers. If the user's device is low on disk space, the on-demand resources will be purged by iOS.

Currently, the Titanium SDK does not provide a way to tag on-demand resources, but support for the feature is in development ([TIMOB-19069](#)).