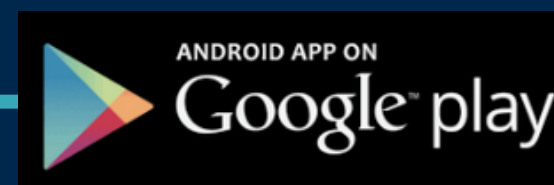


Self-Service: For New Mobile Apps

Step 1



To Start: Send App Information

(1) Email mobilepublishing@umich.edu the following:

- **Name of the mobile app**
- Indicate any app capabilities needed (i.e. push notifications)
- Is it Free or Paid?
- Plans to commercialize?

(2) Submit Disclosure Form

- Inventor = anyone who has made a material contribution in the conception, development and ultimate reduction to practice of the tool.
- This will become the Apple SKU #

3) Review and sign the Mobile App Publishing Responsibility Agreement

Step 2

Who Is on the Mobile App Team?

The Two Roles to Assign:

- **App Manager - MUST BE UM EMPLOYEE**
 - Main point of contact (Project Manager)
 - Oversees the app in the storefronts
- **Developer - internal or external to UM**
 - Sign and upload build files

Send the email addresses and assigned roles to mobilepublishing@umich.edu to create the Users:

App Store Connect:

- Ensure your umich account is setup as your Apple ID (if not, [submit ITS 4Help Ticket to get it setup](#))

Google Play:

- Provide either your personal gmail OR create a new gmail account for your app

NOTICE: Prior to Publishing, it is required the App Manager check-off with these key UM teams to ensure your app is aligned with UM Standards, compliance oversight and security requirements. After this point, you will be ready to publish.

Step 3

Checkpoints:

Click on the names of each group to be routed to their sites

- 1) UM Branding** - guidelines for designing app icon if using the block M
- 2) IRB and IRB MED** - for human subjects research apps only
- 3) ITS Information Assurance** - security screening must be done prior to publishing for production
- 4) ITS Accessibility**

Step 4

Now You Have Access!

The App Manager can begin to fill out details about the app (i.e. description, category, privacy policy), upload screenshots, etc.

The Developer can upload build files.

Here is what is needed to sign and upload:

- App Store Connect:** ITS WeMoAD will share via dropbox
- (1) signing certificate
 - (2) push key (if applicable)

Google Play: The Developer can use own signing certificate

Step 5

Resources

[App Store Connect](#) & [Google Play](#)