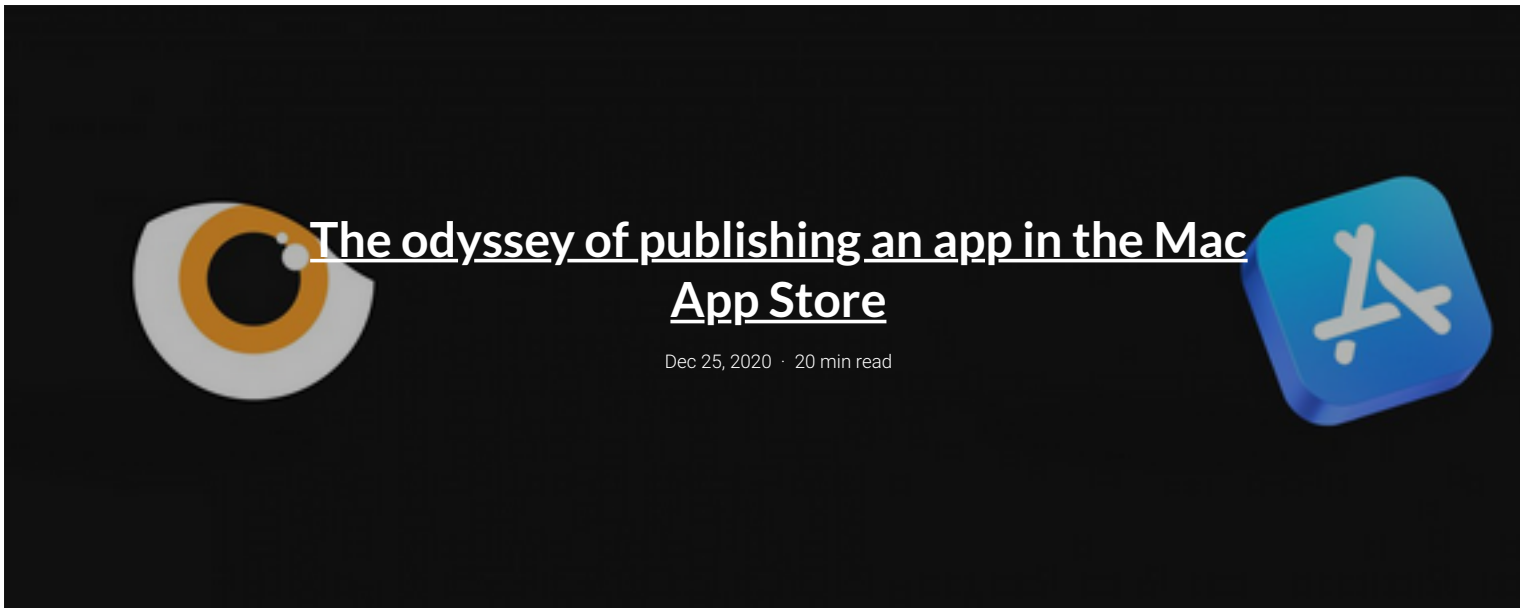


The odyssey of publishing an app in the Mac App Store

Dec 25, 2020 · 20 min read

TL;DR I wanted to post my app Hustl in the Mac App Store but it turned out to be a much more difficult process than I thought. This is a story 6 months in the making, and I'm not even sure it's been worth it. From multiple identity verifications blocking the process for days, to payment bugs, deciphering tax jargon and errors that go silent and break your app, Apple has done everything within their reach to turn what should be a simple process into a giant puzzle.

Many Mac developers choose to self-distribute their apps, that is, they choose to make the app only available via their website, without publishing it in the Mac App Store (MAS), like I did for my app Hustl. Which, at least in principle, sounds counter intuitive: why wouldn't you want to reach as many potential customers as possible?

A good example of this practice is the well-known design app Sketch. Their developers not only decided not to publish their app there, but they actually removed Sketch from the MAS altogether back in 2015.

Years ago, well before I placed a foot in the Mac software industry, I wondered why people did it that way. It didn't make a whole lot of sense. Apps are more accessible and better discoverable via the MAS, and it's not obvious why a developer would choose *not to* be on the store.

But there's a good reason.

Well, there's plenty of good reasons. Some of them I discovered quickly, like licensing, control and revenue.

- **Licensing**: when you're publishing on the MAS, your apps get distributed in accordance to Apple's EULA (End User License Agreement), not your own EULA. This means you can't limit what your users can or can't do with your app (like limiting a license to a single computer). You also can't issue your own licenses, and you have to use Apple's (somewhat rudimentary) licensing system.

- **Control**: when distributing outside the MAS, you have full control and ownership over distribution. You don't have to ask permission on what you can or can't do.

- **Revenue**: Apple takes a 30% cut off all MAS sales. They've just released the Apple Store Small Business Program, which reduces their fee to 15%, but it's still a significant cut.

There's many more reasons why developers choose not to publish in the MAS, and they are better explained here (2015).

When I first released Hustl a couple of years ago, I also chose to distribute it only via its website. But I didn't do it because of financial reasons or because I wanted to be fully in control of all the distribution channels. I did it out of pure convenience, because going for the MAS seemed like a huge amount of work for just testing the waters. And I was not wrong.

Deciding whether to sell on the MAS or not

I can't remember exactly what was my first approach to selling on the MAS, but since the very beginning it seemed like a lot of work for me and I ended up postponing it month after month. My rationale was something like "well, this seems like it'll be a lot of work and I'm not sure it's worth it for a product that I've just launched, so let's wait until it gets some traction and I'll re-evaluate this decision".

And so I did.

Fast forward to ~May 2020. Hustl has been on the market for ~2 years, plenty of people around the world love it and all of a sudden the idea of putting in the time to get it featured on the MAS starts to make more sense.

After all, the MAS is the primary app discovery mechanism for most Mac users.

It was not that uncommon that people would message me: "so where do I get Hustl?" after they did a quick search on the Mac App Store and found nothing. Many Mac users are still not used to buying software outside of the MAS, or they think externally distributed apps are not trustworthy.

So I thought it was time to make it happen.

This is what's needed just to get started as a business on the MAS

As a preface to this story: I own a company. In the very early days, when I was just testing things out, I started doing business as an individual (sole proprietorship) so I could validate demand without incurring in unnecessary costs. But I was no longer doing that – and I didn't know yet, but owning a company would complicate things with Apple.

So, May 2020. I want to publish my Mac app in the Mac App Store. How hard can it be? Most people would think of this process as just uploading the app to the platform, in a similar fashion as when you upload a video to YouTube or a picture to Twitter. Click "New", drag and drop your app, click "Upload" and voilà! You're done.

I had published previously in the iOS App Store – but it was a long time ago and I remembered little about the whole process. Also, it was an iOS app, not a Mac app; and I did it as an individual, which I guess made things a lot easier. I'd soon discover that even "cool" companies like Apple suffer from the worst of bureaucracy.

Past experiences aside, the current problem is: all profits are managed by Apple so I need to submit to them accurate tax and legal info. I only had a personal Apple account until now, and all revenue should go through my company so I needed to open a business Apple account to publish the app through there.

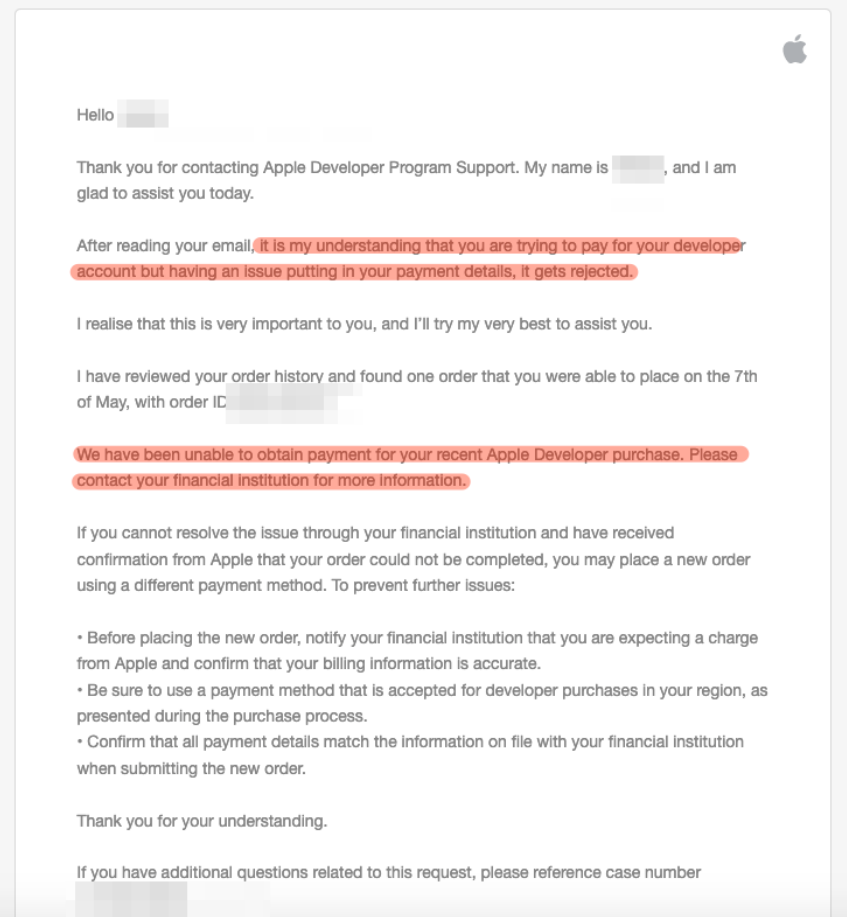
I go ahead and open the new Apple biz account. As soon as I click “Create Account”, I get a mail from Apple. The email conversation went like this:

– Apple: we need to verify your identity via phone
– Me: ok
– Apple: call us then
– Me: it says “Phone support is currently unavailable”
– Apple: nvm let us know when and we’ll call you back
– Me: sure, I’m available tomorrow 5pm
– Apple: ok we’ll call you in 2-4 business days, we didn’t really care about your availability that much lol

A week goes by, I get the call, I get verified, I apply to enroll in the developer program, I pay the $99 fee.

Apple: “we’ll verify your payment in 1-3 business days”

Days go by, I get another Apple email: «your payment didn’t go through, contact your financial institution.»



Hello

Thank you for contacting Apple Developer Program Support. My name is , and I am glad to assist you today.

After reading your email, it is my understanding that you are trying to pay for your developer account but having an issue putting in your payment details, it gets rejected.

I realise that this is very important to you, and I'll try my very best to assist you.

I have reviewed your order history and found one order that you were able to place on the 7th of May, with order ID

We have been unable to obtain payment for your recent Apple Developer purchase. Please contact your financial institution for more information.

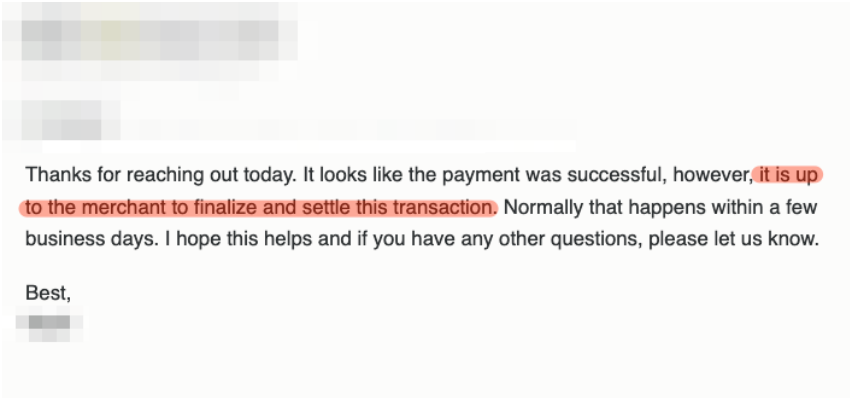
If you cannot resolve the issue through your financial institution and have received confirmation from Apple that your order could not be completed, you may place a new order using a different payment method. To prevent further issues:

• Before placing the new order, notify your financial institution that you are expecting a charge from Apple and confirm that your billing information is accurate.
• Be sure to use a payment method that is accepted for developer purchases in your region, as presented during the purchase process.
• Confirm that all payment details match the information on file with your financial institution when submitting the new order.

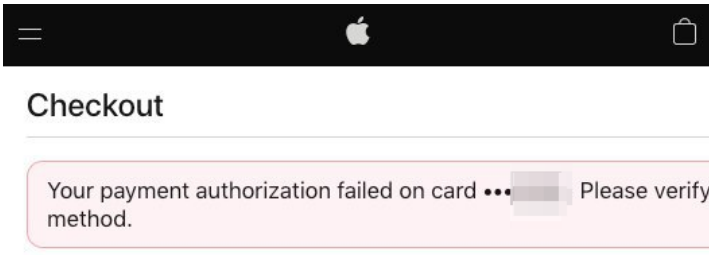
Thank you for your understanding.

If you have additional questions related to this request, please reference case number

I reach out to my bank. They say the payment went through initially but that the merchant didn’t finalize the transaction, and that I should contact Apple.



Apple suggested trying a different payment method, and so I did. I tried almost every card I own. They all got the same problem, which always ended up in this notification days later:



The message Apple gave me on Checkout went like: “Your payment authorization failed on card •••****. Please verify your information and try again, or try another payment method”. Which was not very descriptive of them.

Using different credit cards didn’t make sense either. This was a business expense and I *needed* to use my business CC, plus my billing information was my business’ and (although I tried) it didn’t make sense changing it to match my personal info.

I Googled about this error for days, and everyone on the internet was completely clueless:

- It might be due to unpaid bills / outstanding balance you need to set up (none, this was a new account)
- It might just be your credit card (I tried like 5 cards, none worked)
- It might be that the country that issued the CC doesn’t match your account’s country (mine did)
- It might be that the country you’re in where you’re accessing the Apple Developer website to do the payment doesn’t match the country of the account (tried a VPN, also didn’t work)
- It might be that your billing information is not accurate or has a typo (I tried changing my business info for my personal info, also didn’t work)
- It might be that the bank is rejecting the transaction or labeling it as fraudulent (mine didn’t, I contacted them; plus I tried like 5 different banks, and ffs this is Apple, it didn’t make sense to flag Apple as fraudulent)

So after 15 days of trying payment methods, contacting Apple and banks and just trying to set up my account I decided this was not worth it and I just stopped it. Everyone on the internet was completely clueless, Apple was not being helpful and I thought “well, fuck it”. I would continue distributing my apps outside the MAS and using my personal account to issue the signing

certificates.

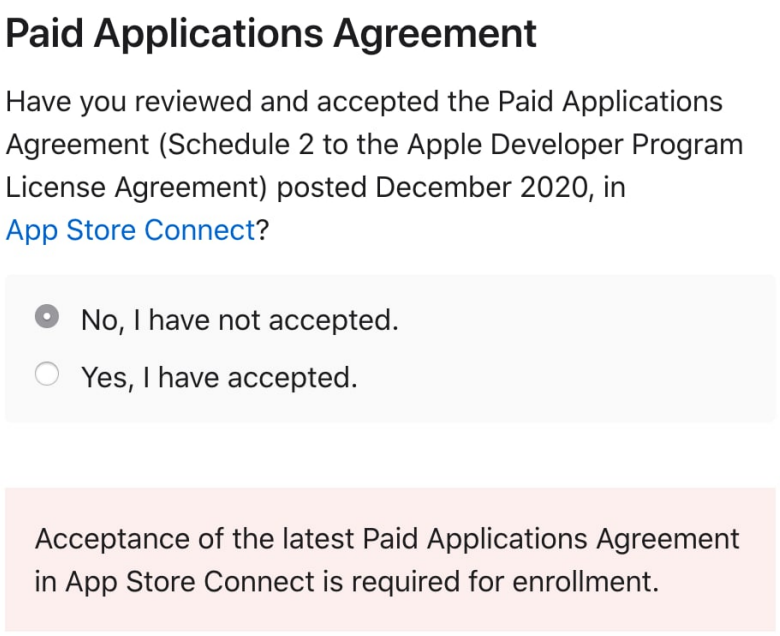
Let's give Apple a second chance

Fast forward to a few weeks ago. With renewed motivation, I tried to go through the process again. So I wait ~5 business days to get verified, I make the payment, I contact my bank and Apple to let them know in advance I don't want any trouble this time.

And I don't know if this was just a bug they fixed since May or if me warning them beforehand did the trick – but I was surprised to discover the payment went through, and I got accepted into the Apple Developer Program one morning. No weird errors, no nothing. Just an email: "Congrats, you're in".

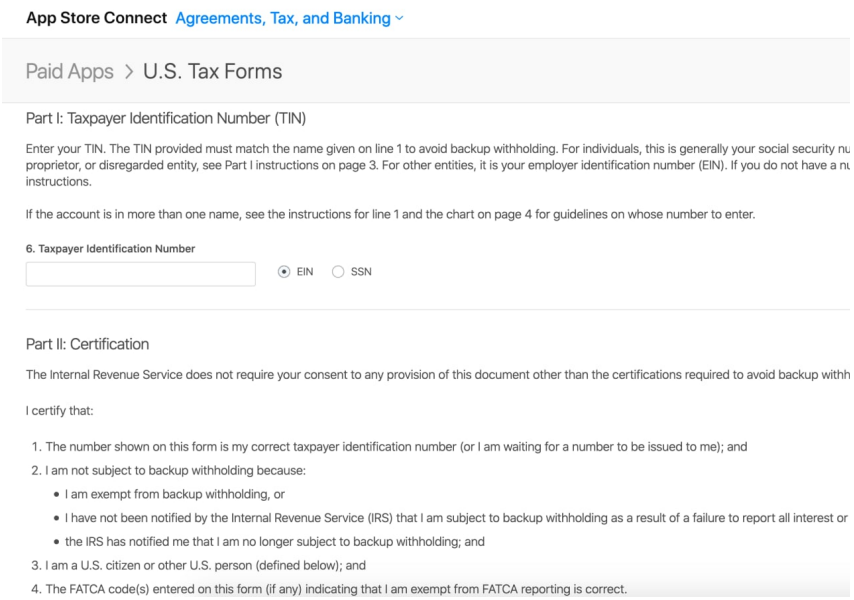
– Me: OK, let's set things up so I can publish my app.

– Apple: no wait, you need to read all these agreements and when you're done with that you need to submit your tax info

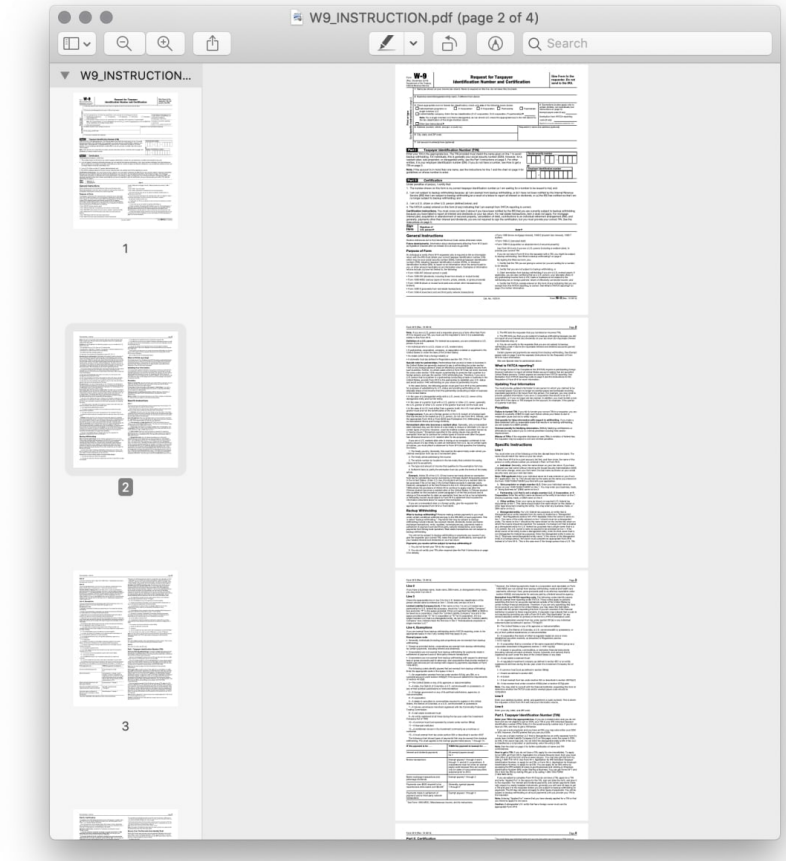


You know you're in for a couple of hours when you read legalese like "Schedule X to the Y document".

So I read the thing, I accept it and after that they prompt me with the less friendly UI I've ever seen from Apple:



If just seeing the screen capture seems boring, imagine having to read through it using 100% of your attention so you understand what they're actually requesting and infer the correct steps to take action. I dive deep into the form they request, and I discover I now need to read all of this document:



The US IRS W9 form instruction Apple needs you to read in order to complete their US Tax Forms page in the Apple App Store Connect dashboard / Apple Developer dashboard

I do, and after that I go ahead to fill up the Apple form. They ask me for a US social security number. I, as an individual, don't have any, so they say: "Read part I of page 3". This is what part I of page 3 looks like:

Part I. Taxpayer Identification Number (TIN)

Enter your TIN in the appropriate box. If you are a resident alien and you do not have and are not eligible to get an SSN, your TIN is your IRS individual taxpayer identification number (ITIN). Enter it in the social security number box. If you do not have an ITIN, see *How to get a TIN* below.

If you are a sole proprietor and you have an EIN, you may enter either your SSN or EIN. However, the IRS prefers that you use your SSN.

If you are a single-member LLC that is disregarded as an entity separate from its owner (see *Limited Liability Company (LLC)* on this page), enter the owner's SSN (or EIN, if the owner has one). Do not enter the disregarded entity's EIN. If the LLC is classified as a corporation or partnership, enter the entity's EIN.

Note. See the chart on page 4 for further clarification of name and TIN combinations.

How to get a TIN. If you do not have a TIN, apply for one immediately. To apply for an SSN, get Form SS-5, Application for a Social Security Card, from your local SSA office or get this form online at *www.ssa.gov*. You may also get this form by calling 1-800-772-1213. Use Form W-7, Application for IRS Individual Taxpayer Identification Number, to apply for an ITIN, or Form SS-4, Application for Employer Identification Number, to apply for an EIN. You can apply for an EIN online by accessing the IRS website at *www.irs.gov/businesses* and clicking on Employer Identification Number (EIN) under Starting a Business. You can get Forms W-7 and SS-4 from the IRS by visiting IRS.gov or by calling 1-800-TAX-FORM (1-800-829-3676).

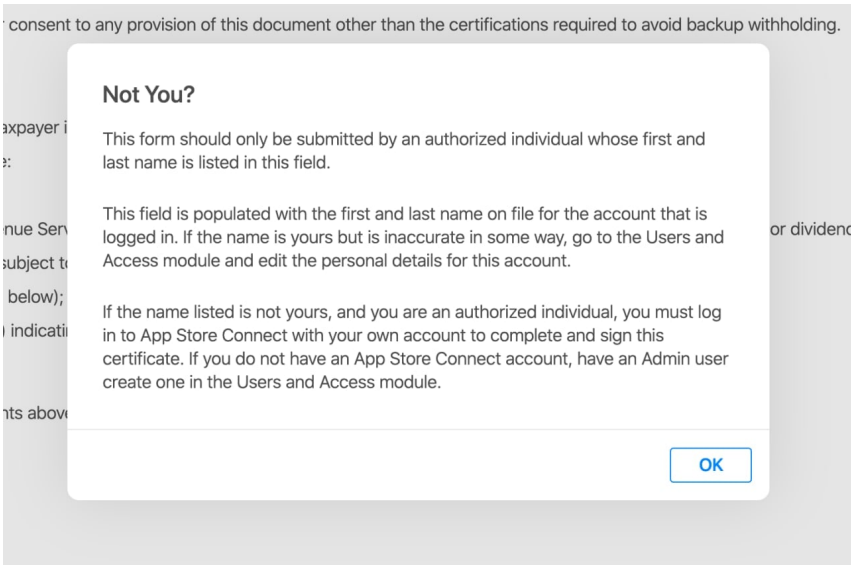
If you are asked to complete Form W-9 but do not have a TIN, apply for a TIN and write "Applied For" in the space for the TIN, sign and date the form, and give it to the requester. For interest and dividend payments, and certain payments made with respect to readily tradable instruments, generally you will have 60 days to get a TIN and give it to the requester before you are subject to backup withholding on payments. The 60-day rule does not apply to other types of payments. You will be subject to backup withholding on all such payments until you provide your TIN to the requester.

Note. Entering "Applied For" means that you have already applied for a TIN or that you intend to apply for one soon.

Caution: *A disregarded U.S. entity that has a foreign owner must use the appropriate Form W-8.*

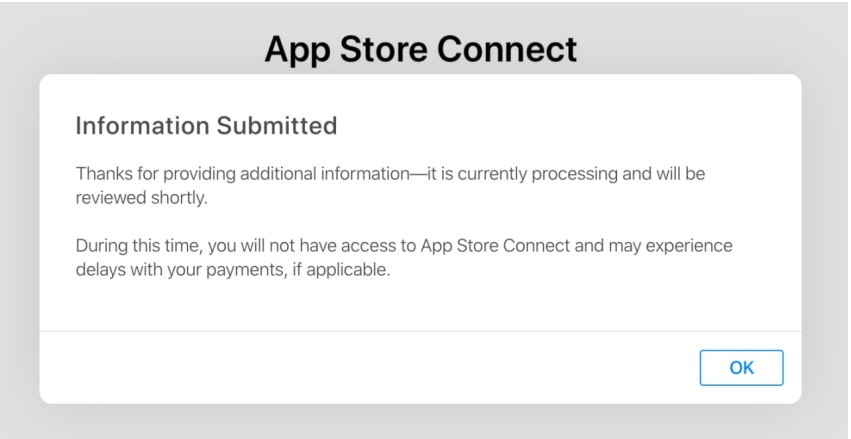
Nice, so I read this and it says I need a different form than the one Apple said I needed.

I lose another couple of hours with this. All to reach a point where they tell me it might not actually be the company info they need but my info.



So I need to log in with a personal account, link it to the biz account, and go through the process again from that account.

I do, they require me to verify my identity (again!!!) and I do.

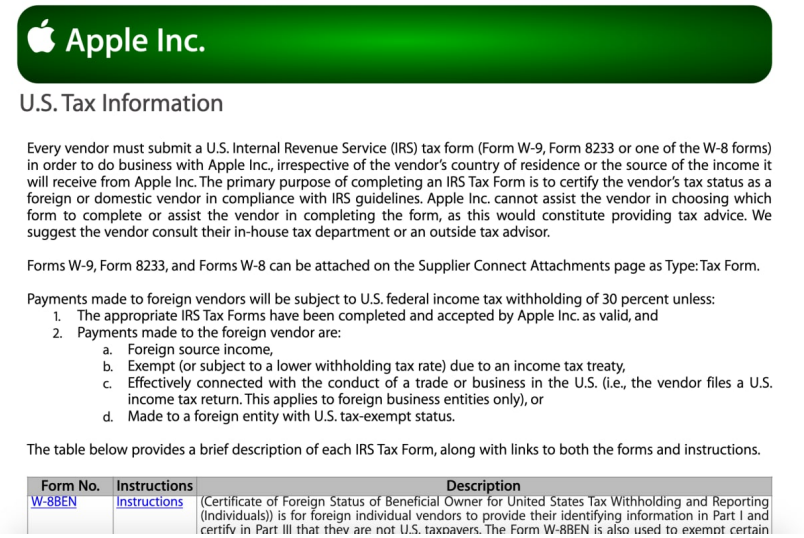


And now they tell me I need to wait God only knows how long until they verify my info and allow me to fill up my tax info, so I can accept the business terms and conditions, so I can set up my business Apple account, so I can finally use it to start getting my app ready.

Like, how much more difficult publishing an app in the App Store can be?

This alone was easily a full week's worth of my time, and just to be able to start the process of publishing an app in the store. And I'm not even halfway there yet.

All of this was really frustrating, but it was also kinda fun to discover the absolutely ugly parts of Apple. Take this legal document, for example:



Apple Inc.

U.S. Tax Information

Every vendor must submit a U.S. Internal Revenue Service (IRS) tax form (Form W-9, Form 8233 or one of the W-8 forms) in order to do business with Apple Inc., irrespective of the vendor's country of residence or the source of the income it will receive from Apple Inc. The primary purpose of completing an IRS Tax Form is to certify the vendor's tax status as a foreign or domestic vendor in compliance with IRS guidelines. Apple Inc. cannot assist the vendor in choosing which form to complete or assist the vendor in completing the form, as this would constitute providing tax advice. We suggest the vendor consult their in-house tax department or an outside tax advisor.

Forms W-9, Form 8233, and Forms W-8 can be attached on the Supplier Connect Attachments page as Type: Tax Form.

Payments made to foreign vendors will be subject to U.S. federal income tax withholding of 30 percent unless:

1. The appropriate IRS Tax Forms have been completed and accepted by Apple Inc. as valid, and
2. Payments made to the foreign vendor are:
   a. Foreign source income,
   b. Exempt (or subject to a lower withholding tax rate) due to an income tax treaty,
   c. Effectively connected with the conduct of a trade or business in the U.S. (i.e., the vendor files a U.S. income tax return. This applies to foreign business entities only), or
   d. Made to a foreign entity with U.S. tax-exempt status.

The table below provides a brief description of each IRS Tax Form, along with links to both the forms and instructions.

Form No.	Instructions	Description
W-8BEN	Instructions	(Certificate of Foreign Status of Beneficial Owner for United States Tax Withholding and Reporting (Individuals)) is for foreign individual vendors to provide their identifying information in Part I and certify in Part III that they are not U.S. taxpayers. The Form W-8BEN is also used to exempt certain

To me, it's crazy to see how different the back office is from what people usually see, even for cool companies like Apple. Lawyers are not designers, after all.

This is what bureaucracy is. A never ending process of redundant validations in which you need to make your way through very complicated forms with the constant fear of doing something wrong.

This is bad, but it's debatable whether it's avoidable or not. I think it may just come inevitably with scale. Like, once you have, say, 1.000+ employees in 50+ different countries, you *need* to have some sort of standard procedure in place so you can replicate your business practices across all places you operate. But that comes along with complexity, rigidity and bureaucracy. I think that's partly why it's so nice, fun and easy when companies are small / indie.

Also, a big contributor to this mess is the IRS code. As Sean said when I was sharing this story with friends: "American red tape is the worst tape". US IRS code definitely is one of the most complicated things I've ever had to navigate. My local IRS code is not any better though. I always want to cry from frustration to do the easiest things. And the worst part is – it's like they make it

ambiguous on purpose. Like you never really know if what you're doing is right or not – and it feels like no one really knows.

The technical part: adapting and uploading the app to the MAS

We're only halfway through.

And what's ahead of us is arguably the most difficult part yet: the technical challenges of adapting the app to the App Store, and making everything work so Apple accepts the binaries.

As a first step, I started removing my own license verification system from the app. Copy protection is an issue – whether you're going to post your app in the MAS or not. The thing is Apple doesn't really allow you to implement your own copy protection mechanism, and it explicitly says so in the App Store Guidelines

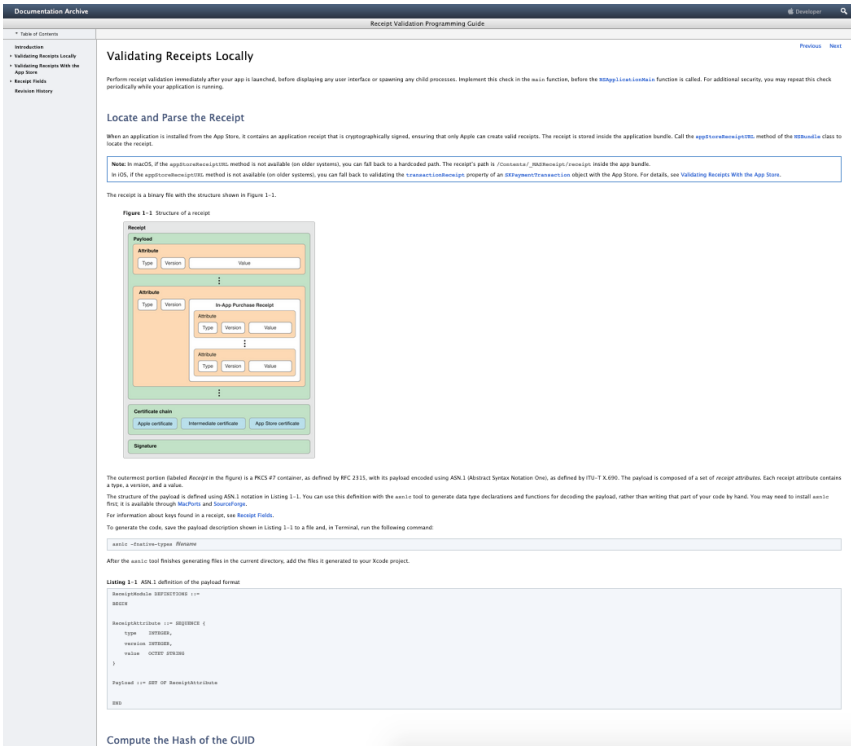
> **2.4.5 Apps distributed via the Mac App Store have some additional requirements to keep in mind:**
> **(vi) They may not present a license screen at launch, require license keys, or implement their own copy protection.**
>
> ***The part of the Apple App Store Review Guidelines that basically gets you rekt. Source.***

My worry: that without my own copy protection mechanism if literally just one user downloads my app they could just go to their Applications folder, copy it and paste Hustl in any site or forum and publish it so everyone would get access to it for free. Yes, this is sadly the way it would work.

> **"It is up to the software developer to check the receipt or check a developer website to see if the software has permission to run on that device. Some software does that, other software does not. And some software checks and if the program is not 'entitled' to run on the device does the following – run for about a week and then start to bug-up and fuck with the user who stole it. The 'fuck' could be simply to announce – "all your work will be lost if you do not buy a legit copy" or it could be something much more nefarious and suitable only for thieves – after all, isn't that what you are at that point?"**
>
> ***Someone said this on the Apple Developer forums***

I spent a good deal of time wondering how people solved the piracy problem in the MAS in an elegant manner.

Until I found a rather obscure documentation page from Apple:



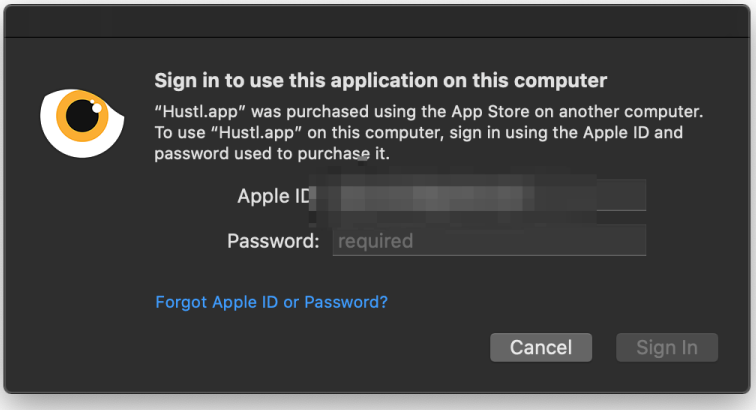
It basically says that when apps are downloaded from the MAS, Apple adds a _MASReceipt folder inside your app, which holds the Apple-issued receipt of the purchase, and explains what's inside that folder and how to validate its contents correspond to a good purchase made from the computer from which the app is running.

I would have never found that piece of information if it wasn't for Marc, who clued me in by sharing with me this repo he found (thanks! 🙌)

That did the trick. But it wasn't self-evident how to make it work properly:

- You have to test the app doesn't launch when it doesn't have a valid receipt
- You have to test the app doesn't launch when it's run from a computer different from the computer to which the receipt was issued (i.e.: copying and pasting the app from one computer to another shouldn't work)
- You have to test the app works normally when it's issued a valid receipt (which raises the question: where and how do you get a valid Apple-issued receipt for your app if it's not yet in the MAS?)
- You have to make sure the app will work normally and correctly for the Apple reviewers the first time they open it – or otherwise you might get rejected from the MAS

These questions were not obvious at all, and I found that these were not clearly documented anywhere. The part that bugged me the most was discovering how to get a valid receipt issued by Apple to test my app without being in the Mac App Store. Turns out Apple generate those when the receipt verification fails and you get this screen:



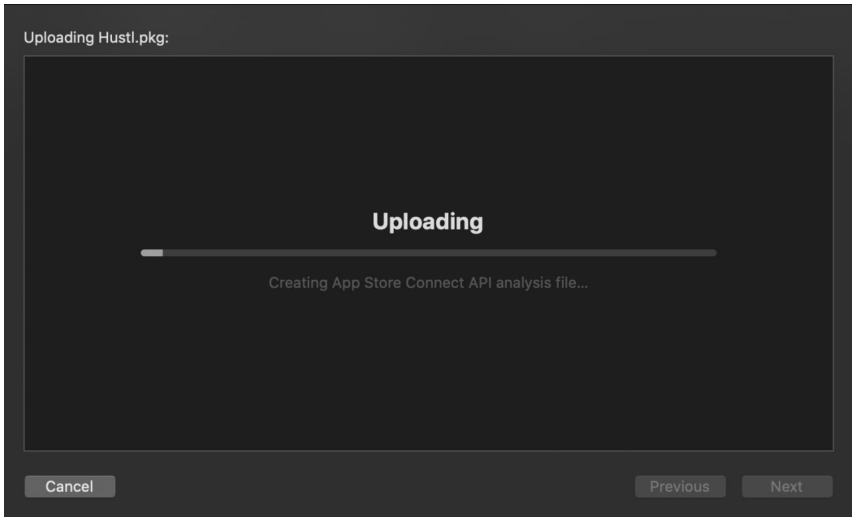
This is the "error" you get when locally validating the MAS receipt for your app while developing before having submitted your app to the Mac App Store: «"X.app" was purchased using the App Store on another computer. To use "X.app" on this computer, sign in using the Apple ID and password used to purchase it.»

BUT, big caveat, MAS test receipt generation only works if:

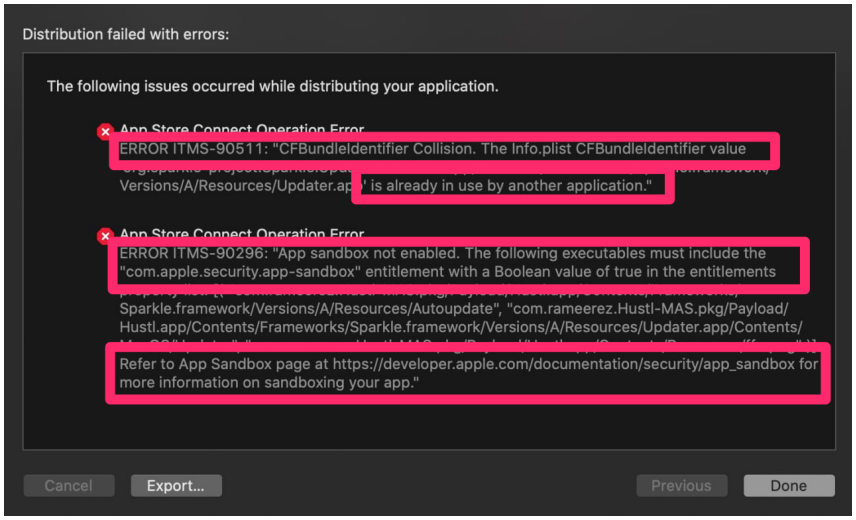
- You've created an app in App Store Connect with the same bundle id that the app you're launching and trying to get a receipt for; AND:
- You've created a Sandbox Tester account on App Store Connect and you're using those Apple ID / Password credentials when prompted (for reference, you create one by going to App Store Connect → Users and Access → Sandbox → Testers → Add new)

Then Apple will issue you a test MAS Receipt and your app will hopefully pass the copy protection tests.

I won't talk about all the errors that came up during the development of the MAS-side parts of the app, because I assume that's just part of the process. Let's just say I put 4-6 hours worth of engineering work into adapting the app for the Mac App Store and we magically reach a point where the app is ready to be uploaded to Apple. And so I do:



I go get myself a well-deserved coffee, and when I come back I find this on my screen:



The errors read something like:

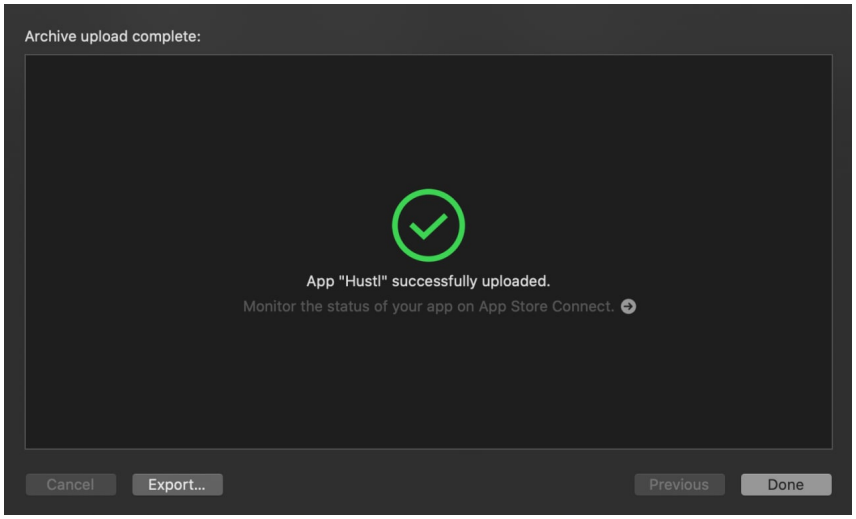
> **Distribution failed with errors. The following issues occurred while distributing your application.**
>
> **❌ App Store Connect Operation Error. ERROR ITMS-90511: "CFBundleIdentifier Collision. The Info.plist CFBundleIdentifier value *** is already in use by another application"**
>
> **❌ App Store Connect Operation Error. ERROR ITMS-90296: "App sandbox not enabled. The following executables must include the "com.apple.security.app-sandbox" entitlement with a Boolean value of true in the entitlements property list: ***. Refer to App Sandobx page at https://developer.apple.com/documentation/security/app_sandbox for more information on sandboxing of your app.**

Please note that this is a perfectly functional and working app that's being delivered via several different distribution channels, none of which had rose any issues so far. People download and use the app without a problem. It shouldn't raise any errors, especially because Apple already notarizes my app before I distribute it anywhere. So here we go.

Apple is complaining my app bundle still contains a library I actually no longer need, and I'm using a binary that apparently wasn't sandboxed and Apple needs me to sandbox it.

So again I'm faced with the problem of how to do this (there's no documentation for this, or at least not that I've found easily, and all the info I had was through StackOverflow questions and GitHub issues). I had to create and modify an entitlements file, add several steps to the build process and a couple more things I don't even remember.

When I think I've finished, I re-upload it to Apple:



Yes!

That's it. I immediately go to the App Store Connect dashboard to finish filling up the app listing and get it ready for the review process.

But then I'm faced with more bureaucracy.

Bureaucracy round two: Apple limitations

There's a couple of things that I was sort of expecting, but they were the straw that broke the camel's back.

First of all: naming. App names on the MAS are limited to 30 characters. I wanted to name my app "Hustl: Timelapse Screen Recorder" but that didn't fit, leaving me with: "Hustl: Timelapse Screen Record". Which was not exactly optimal. So I had to put more time into thinking of a good name.

Then it was pricing. I sell my app for $29, but believe it or not, Apple does not allow developers to set arbitrary prices for their apps. They tell you what fixed pricing they have available, and you get to choose from their fixed tiers.

Some of Apple's pricing tiers for apps in the App Store

The closest it got to $29 was either $28.99 or $29.99. Either way, I have to face the question of which one to choose and change the whole pricing of the product everywhere just to stay consistent with the App Store pricing.

And not only that: pricing differs wildly for each currency, and there's nothing I can do against that. So for example, the $29.99 USD tier gets automatically converted as 32.99€ for European users.

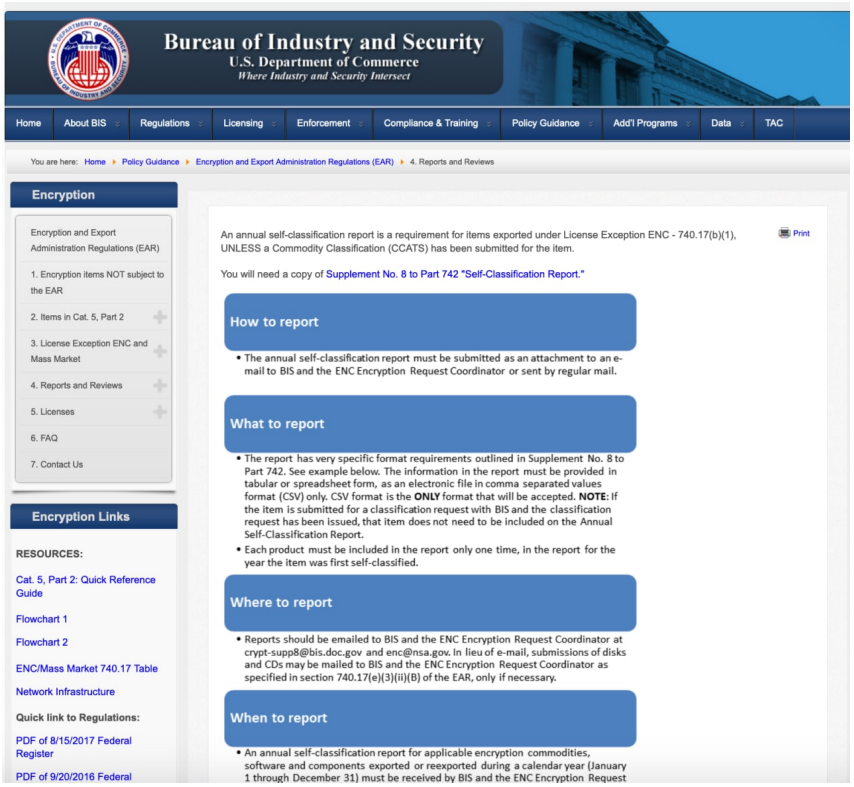
So I not only need to change the pricing on my website (and maybe for other developers, all other distribution channels), but I also potentially need to account for users visiting from other parts of the globe if I want the pricing to stay consistent across territories.

At this point, I remind myself that if I didn't want to implement this distribution channel, it wasn't because I didn't want to grow Hustl's reach, but because of all the downsides it came along with. Unless I was pretty certain this would be a good idea, it wouldn't have made a lot of sense to do it – especially not in the early days.

But we're this far, and we're not going to give up now.

I make decisions and keep filling the app review submission form, when I'm faced with this after selecting my app version:

Apparently using any sort of encryption in your app is a big deal for Apple. If your app uses encryption of any sort, you should submit a form to the US government two times a year informing them of how you use encryption in your app. Even an HTTPS call might get you in trouble. For reference, this is only the beginning of what you're supposed to read, understand and act upon two times a year:



So that's what, another 2 hours of just reading and understanding what's needed and if you need to do something – and then actually take the steps to comply with this? I'm starting to feel really, really frustrated at this point.

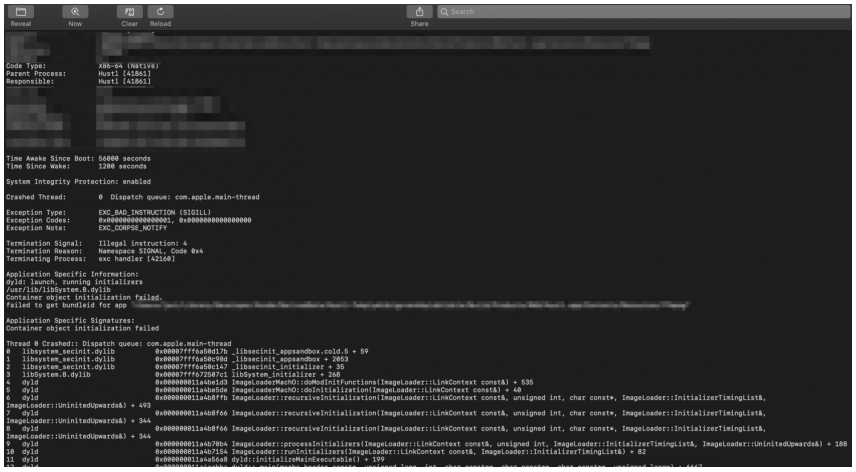
Anyway, the app listing is looking better than ever and I'm getting ready to click that shiny "Submit for Review" button. But before I do that, I want to make sure my app is working alright. So I go to the version of the app I uploaded, double click it to open it, use it… and it's not working.

Not that it's crashing: it looks like it's working normally. But something is wrong, because it's not completing its core task. Everything looks like it's working but the user is left with nothing. There's no errors, the whole process just crashes silently.

Off to debug it.

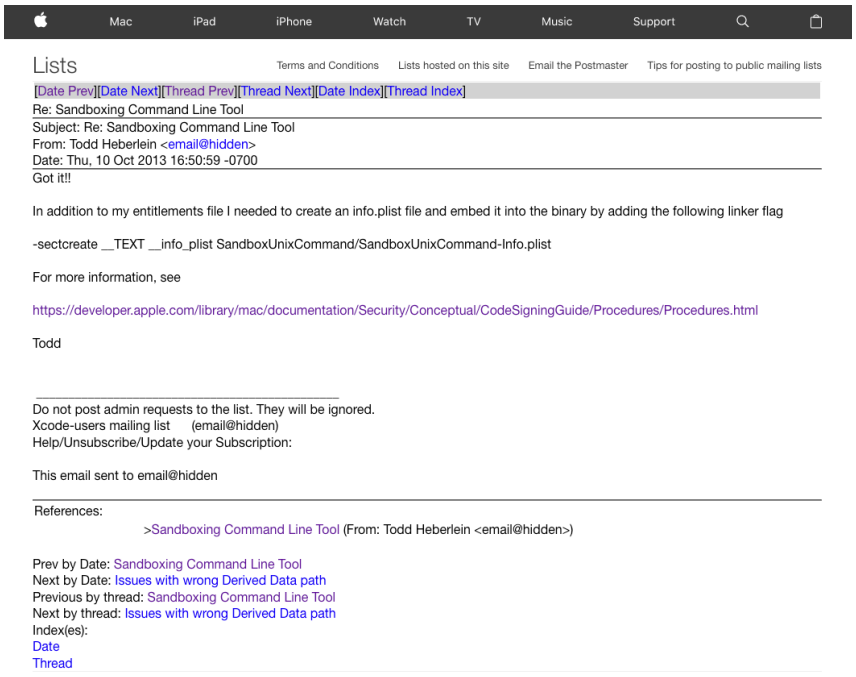
Technical problems round two: Apple broke my app

I lose another couple of hours trying to find the root cause of the error, until I find a nasty-looking error log like this one.



The important bit is the "failed to get bundleid for app" part, but technical jargon aside, this was not easy to solve.

I ended up browsing a 10+ years old mailing list in which someone found a compiler flag you need to set in order for this to work correctly – something that should be documented by Apple but isn't.



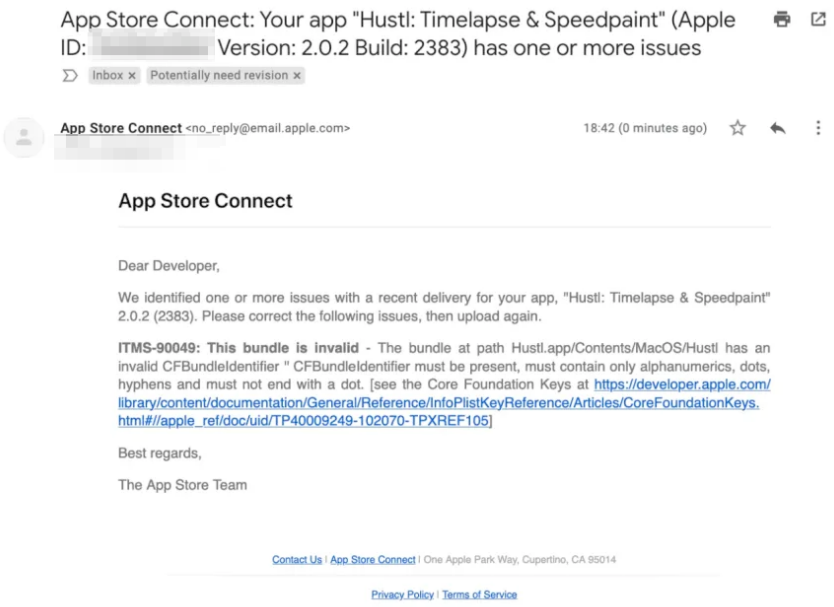
Not kidding, key information to solve the problem often lies in weird, obscure, old-fashioned systems like this one (and thank God someone actually posted it)

There were a bunch of other technical reasons why this didn't work, but we'll leave that for another

post.

After fixing everything, I re-upload the App to Apple. Everything seems to work fine until I try to select the newly uploaded app to send it for review.

My app doesn't show up in the interface now. I reload the page trying to look for the error, when suddenly I get an email from Apple:


App Store Connect: Your app "Hustl: Timelapse & Speedpaint" (Apple ID: [redacted] Version: 2.0.2 Build: 2383) has one or more issues

Inbox × Potentially need revision ×

App Store Connect <no_reply@email.apple.com> 18:42 (0 minutes ago)

App Store Connect

Dear Developer,

We identified one or more issues with a recent delivery for your app, "Hustl: Timelapse & Speedpaint" 2.0.2 (2383). Please correct the following issues, then upload again.

ITMS-90049: This bundle is invalid - The bundle at path Hustl.app/Contents/MacOS/Hustl has an invalid CFBundleIdentifier '' CFBundleIdentifier must be present, must contain only alphanumerics, dots, hyphens and must not end with a dot. [see the Core Foundation Keys at https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html#//apple_ref/doc/uid/TP40009249-102070-TPXREF105]

Best regards,

The App Store Team

Contact Us | App Store Connect | One Apple Park Way, Cupertino, CA 95014

Privacy Policy | Terms of Service

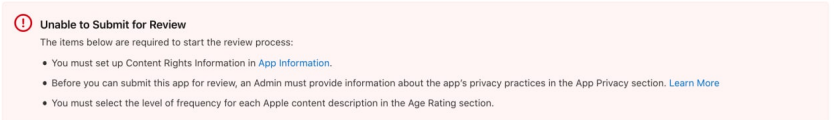
This was just one of ~10 emails I ended up getting. Turns out Apple doesn't perform all the required code verifications in your local computer, as it would be reasonable to think, but instead waits until the binaries hit their servers to run a suite of tests against them and then notifies you of any errors via email.

The problem with this is that it's extremely time consuming. Just to get that email, I had to invest 10-15 minutes:

- Fixing issues (~5 minutes)
- Compiling / Archiving the app (3-5 minutes)
- Signing the app (2-3 minutes)
- Uploading the app (2-3 minutes)
- Waiting until the app gets tested by Apple's servers (2-3 minutes)

The aftermath is: 10-15 minutes can easily go by from the moment you fix the error until the moment you receive the next piece of feedback of whether what you did was correct or not. That's just a lot of time for a feedback loop that could be almost instant.

So I manage to fix everything, the app runs fine, Apple seems to be happy with the binaries… I hit "Submit for Review" and…


Unable to Submit for Review

The items below are required to start the review process:

- You must set up Content Rights Information in App Information.
- Before you can submit this app for review, an Admin must provide information about the app's privacy practices in the App Privacy section. Learn More
- You must select the level of frequency for each Apple content description in the Age Rating section.

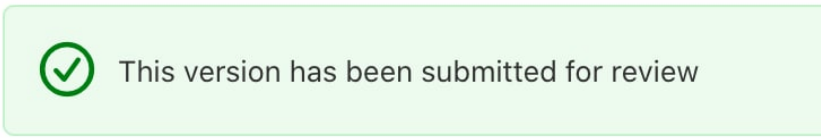
I wish I was kidding.

This is just bad UX.

These errors could *so easily* be displayed just in the right moment while the user is filling up the form, rather than waiting until someone clicks the Submit button. It's so frustrating. It makes you want to stop the process altogether.

Fortunately these errors were not so difficult to solve in comparison, so I went through each one of these rather smoothly and clicked Submit again.

This time, I got this:

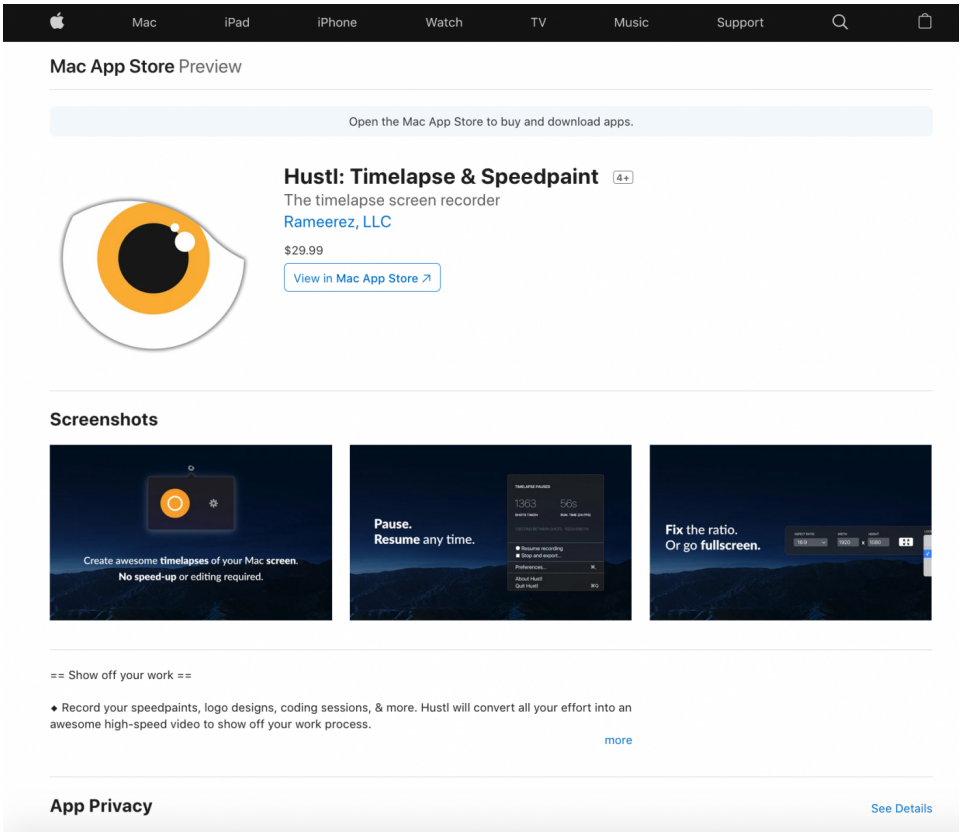


I wish I could tell you I was happy to see that green message, but I was just exhausted.

I was completely tired, and so annoyed I couldn't enjoy the moment. It felt like "OK, so what's going to be next? What am I doing wrong this time?".

Not everything was excruciating pain, though: Apple reviewed my app in what it felt like a record time of just 2 hours and it was up and ready for sale in the App Store sooner than I could even finish dinner (pro trick: I submitted my application hours before Apple was closing their App Store operations worldwide for the Christmas holidays, so I think they just were in a hurry and that's why they reviewed my app so quickly)

So I'm happy to announce, after all this odyssey: this is the official Mac App Store link to get Hustl 😎



Go check out Hustl on the Mac App Store

Finishing remarks

I still think an email from Apple might arrive at any moment, stating that my app is invalid in some weird way and inquiring I need to do God-only-knows-what in order to make it acceptable for Apple.

It's like imposter syndrome but for apps. And it feels horrible.

Some questions arise after going through this hell:

- Has this been worth it? (ask me in a year)
- All of this – just to get just 70% of the profits? Apple takes a 30% cut of all App Store sales (they're reducing it now to 15% with their new Small Business Developer Program, but it's still unlaunched)
- How could this process be improved?
- I've also dealt with uploading apps to Google stores before, and it's by no means anything compared to this. With Google, everything is easier and more intuitive. In contrast, apps in Google's app stores tend to be of a much lower quality than those of Apple. Does that outcome really justify the pain I had to go through? (IMHO: it does for the users, it doesn't for the developers)

Oh, and by the way!

I'm most active on Twitter – follow me to stay in the loop.