

# Setup/Configurations for Mac Mini

## Connecting

*\*You will need to be at our office or on the VPN to do this*

1. Download [VNC Viewer](#)
2. Make sure the Mac you're wanting to connect to is actually plugged in (check the ethernet cable) and turned on
  - The two QA ones are by all the other QA testing machines (as of 10/25/2022)
  - The Dev one is by McKay's space - in the Ops corner (as of 10/25/2022)
3. Once it's open, add a new connection see credentials
  - The dev mac currently is the only machine that has all the signing certificates for the Mac Desktop Client
    - The dev mac is often turned off
  - The QA mac 1 is much faster than any of the others, but doesn't have the signing certificates for the Desktop Client
    - When setting up the QA mac within VNC, turn Encryption > Prefer Off
    - Make sure no one from QA is using it

## Credentials

### DEV MAC

**IP Address:** 10.10.0.60

**Login Info In Last Pass:** Development Mac Mini

### QA MAC 1

**IP Address:** 10.10.0.68

**Login Info In Last Pass:** QA Mac Mini 1

### QA MAC 2 (not recommended)

**IP Address:** 10.10.0.67

**Login Info In Last Pass:** QA Mac Mini 2

# Before you get started

You will need the following installed on your machine (Dev mac and QA mac 1 have these already):

- XCode
  - **Note:** Be careful with updating xCode versions, as it is likely to break building in release mode for the mac desktop client
  - Make sure you open XCode and check all the boxes to install the things needed
- Visual Studio for mac with:
  - .NET
  - .NET WebAssembly Build Tools
  - macOS (Cocoa) - *all versions listed*
  - Android
  - iOS
- Our codebase

## Version Control

**IMPORTANT** - the master branch for the Mac Desktop Client is different from windows. You're looking for *origin/mac-master*. Branch off of that unless instructed otherwise

Sometimes git through Visual Studio for mac is really dumb, especially with changing branches. The good news is that you can use git through terminal where needed.

- If using Visual Studio for Mac for version control, make sure that when you bring the branch into your local environment, that you also track the branch, this will prevent a few issues

### Useful Git Commands

- Force change branches => `git switch -f <branch-name>`
- It tries rebasing on branch swap sometimes => `git rebase --quit`

## NuGet

You will probably need to add our UtopiaNuget at some point to a project on mac. The process is slightly different than what is described in [our other wiki page](#).

1. Go to the NuGet packages in the solution and select Configure Sources
  -  and or type unknown
2. Go in to ADO and select Personal Access Tokens
  -  and or type unknown
    1. Select + *New Token*
    2. Give it a name
    3. Change the expiration to be far in the future
    4. Give full access scope
    5. Click *Create*
    6. Save the token you are given and don't lose it
      - You will not be given it again and will need to regenerate another one if you lose it, potentially breaking other connections
3. Add a new package source
  1. Name it UtopiaNuget
  2. The location is  
[https://pkgs.dev.azure.com/eFileCabinet/Utopia/\\_packaging/UtopiaNuget/nuget/v3/index.json](https://pkgs.dev.azure.com/eFileCabinet/Utopia/_packaging/UtopiaNuget/nuget/v3/index.json)
  3. The username is your ADO username
  4. The password is the token you generated in step 2
  5. Click *Add Source*
4. Click *OK*
5. At this point, if you select UtopiaNuget as the package source and are on the browse page, you should be able to see all of our packages we offer.

## Troubleshooting

- If you finish all the steps successfully, and can see all the packages we offer, but restoring the packages still gives you a problem, then Visual Studio for mac messed up on [adding a nuget reference to the global nuget file](#).
  - The fix is to search for the hidden file NuGet.Config and add the reference to the new package source manually
    -  and or type unknown
    -  (ignore the double reference)

# Mac Desktop Client

**IMPORTANT** - the master branch for mac is different from windows. You're looking for *origin/mac-master*. Branch off of that unless instructed otherwise

If you're feeling courageous and want to mess with getting the signing certificates working on a machine other than the dev mac machine, then the credentials are found in last pass and is named Mobile Dev Apple Account.

## Helpful Tips

It may be beneficial to run a local instance of Utopia to connect to for the Desktop Client.

- [See our page on how to do this](#)
- It is only possible to do from the office since you're remoting into the mac for testing

## Building in Debug Mode

*\*If you're building on a machine that doesn't have the signing certificates (any other than Dev Mac Machine), you will need to set signing to false in the project file.*

To be able to build, you need to first have built two other projects on the machine:

- UtopiaAPIClient
- SharedLogic

It is very likely that you will need to update their references to UtopiaSharedClasses in NuGet. I recommend going into the UtopiaWindowsDesktopClient to update the packages and build them.

## Building in Release Mode

*\*As of right now the signing certificates are only on the Dev Mac machine. I've tried putting it on the other mac, and they are there, but they require a password per use, which we don't have.*

To build in release mode, it's as simple as putting it in release mode in visual studio, updating the version number in the Product.plist file, and running build. Once done, then go into the bin folder to get the dmg file

```
../UtopiaMacDesktopClient/RubexClient/bin/Release/RubexInstaller.dmg
```

Send this to QA to test your changes. If they sign off on it then you're good to have it replace the dmg file in Utopia. Make sure you update the version number in the json file in the Utopia project after updating the dmg file!

## Note

There is a chance that building it won't sign it properly or build the dmg. I never had this problem, but if you do then you will need to follow these steps:

Monterey 12.0.1 has build issues when trying to build an installer from Visual Studios when running the release mode for Rubex Desktop Mac Application. Had to manually build the files.

There is a Product.plist file in the obj/release folder that has a property called "os", delete this entry and run the below commands (with the proper version).

```
productbuild --product /Users/dev/UtopiaClientApplications/UtopiaMacDesktopClient/RubexClient/obj/Release/Product.plist --component /Users/dev/UtopiaClientApplications/UtopiaMacDesktopClient/RubexClient/bin/Release/Rubex.app /Applications --sign "Developer ID Installer: eFileCabinet, Inc. (J82X76G6Y8)" /Users/dev/UtopiaClientApplications/UtopiaMacDesktopClient/RubexClient/bin/Release/Rubex-1.0.4.pkg
```

```
appdmg /Users/dev/UtopiaClientApplications/UtopiaMacDesktopClient/RubexClient/InstallerItems/spec.json /Users/dev/UtopiaClientApplications/UtopiaMacDesktopClient/RubexClient/bin/Release/RubexInstaller.dmg
```

# iOS Mobile

*\*I haven't touched this much but we do have a [helpful page already built](#). I don't know how up to date it may be.*

---

Revision #23

Created 19 July 2022 13:36:16 by Bryce Holloway

Updated 9 February 2023 20:20:49 by Caleb Baker