General Instructions for Deploying WIT_stable

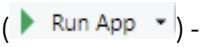
Preparing for Deploying App Locally

If running in Windows, verify the following prior to running:

- Check that you have at least R 4.3 (preferably 4.4) installed.
- Verify RStudio is up to date as well.
- Unzip the app folder.
- Make sure Rtools is installed and up to date.
    - You can verify if (and which version of) Rtools is installed by running the following in the RStudio console: `Sys.which("make")`.
    - For R version 4.4, you should have the corresponding Rtools 4.4 installed so `Sys.which("make")` should yield:

      `"C:\\rtools44\\usr\\bin\\make.exe"`
    - Link to download updated Rtools: https://cran.r-project.org/bin/windows/Rtools/.
        - Make sure to add to PATH.
        - Use this helpful "Installing Rtools" Guide if you need additional instruction.
- Make sure you have Java installed on your device.
    - See **Debugging** section (end of document) for instructions if you do not have Java installed.

Deploying the App

- I recommend opening the `NNCTAnetworkFunction.proj` file (from File Explorer) in RStudio.
    - Can also opt to **Open Project** from within RStudio.
- When it opens in RStudio, it may give warnings about `renv` being out of sync - disregard them, as `renv::restore()` has yet to be run.
- From the **Files** pane of RStudio, open the `app.R` script.
- In the top right of the script, there will be a **Run App** button () - click it. It will take a little time to call all the packages. If prompted to install `shiny`, click **Yes**.
    - Side tip? You can try running `data_import_local.R` found in the `R` folder before clicking **Run App**, just to make sure the packages load okay.
        - Disclaimer: I haven't tried this route but seems potentially promising as it will get the packages loaded up.
- You may need an internet connection to install the packages the first time, but I'm not sure. To err on the side of caution, I recommend you try running it without an internet connection after you get it working on your device, just to check.

Everything else should run as you've seen it in the website deployment.

Debugging

A few known errors when deploying in Windows:

ERROR 1

```
error downloading 'https://packagemanager.posit.co/cran/latest/src/contrib/PACKAGES'
[curl: (35) schannel: next InitializeSecurityContext failed: Unknown error (0x80092012)
- The revocation function was unable to check revocation for the certificate.]
```

Potential Cause

The error message indicates a problem with the SSL certificate verification process when attempting to access the specified CRAN repository via HTTPS. This issue can arise due to several reasons such as network configuration issues, outdated CA certificates on the system, or problems with the repository's SSL certificate. The specific error `schannel: next InitializeSecurityContext failed: Unknown error (0x80092012) - The revocation function was unable to check revocation for the certificate` suggests that the system is unable to verify the revocation status of the SSL certificate presented by the server.

Recommended Solution

Enter `options(repos = c(CRAN = "https://cran.r-project.org"))` in the `.Rprofile` file.

ERROR 2

```
** libs
Error in system(paste(MAKE, p1(paste("-f", shQuote(makefiles))), "compilers"),  :
  'make' not found
* removing 'C:/xxx/xxx/AppData/Local/R/win-library/4.4/.renv/1/sf'
install of package 'sf' failed [error code 1]
```

Potential Cause

Rtools is not installed or, if installed, outdated.

Recommended Solution

Go to https://cran.r-project.org/bin/windows/Rtools/ to install the latest Rtools.

- When installing it, make sure to add it to PATH!

Once installed, you can verify you properly added it to PATH by running `Sys.which("make")` in the console. If it is properly installed and added, you should have the following output:

```
> Sys.which("make")
                                                    make
"C:\\rtools44\\usr\\bin\\make.exe"
```

ERROR 3

```
Error: package or namespace load failed for ‘mailR’:
 .onLoad failed in loadNamespace() for 'rJava', details:
  call: fun(libname, pkgname)
  error: JAVA_HOME cannot be determined from the Registry
```

Potential Cause

“The error tells us that there is no entry in the Registry that tells R where Java is located. It is most likely that Java was not installed (or that the registry is corrupt).”

*Source: **R-statistics blog: How to load the {rJava} package after the error "JAVA_HOME cannot be determined from the Registry"***

Recommended Solutions

Fix 1:

This error is often resolved by installing a Java version (i.e. 64-bit Java or 32-bit Java) that fits the type of R version that you are using (i.e., 64-bit R or 32-bit R).

1. Install 64-bit Java from **https://java.com/en/download/manual.jsp**.

   Note that it is necessary to ‘manually download and install’ the 64-bit version of JAVA. By default, the download page gives a 32-bit version.

2. Then in Windows **cmd** run:

   ```
   setx PATH "C:\Program Files\Java\jre1.8.0_211\bin\server;%PATH%"
   ```

   – Make sure the file path is correct!

Fix 2:

Another possible solution is trying to re-install **rJava**.

```
remove.packages("rJava")
install.packages(“rJava”)
```

Fix 3:

If that doesn’t work, you could also manually set the directory of your Java location by setting it before loading the library:

```
Sys.setenv(JAVA_HOME="")
Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jre7') # for 64-bit version
Sys.setenv(JAVA_HOME='C:\\Program Files (x86)\\Java\\jre7') # for 32-bit version
library(rJava)
```