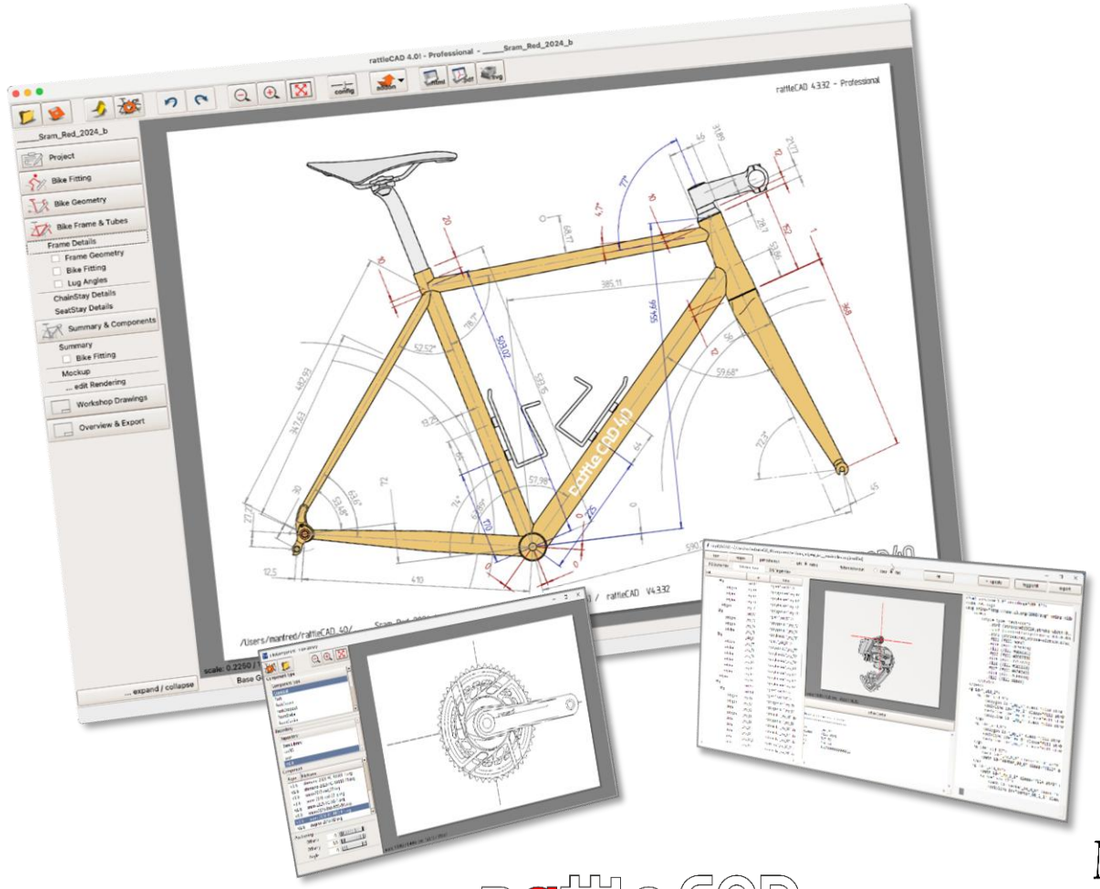


Tcl9

21st European
Tcl/Tk Conference

July, 10th – 11th 2025, Bologna, Italy

Tcl/Tk-based Desktop Apps for MacOS

… ready for Codesigning and Notarization

Mac™ OS

rattleCAD.com

Manfred ROSENBERGER

Resources:

https://www.little-feather.com
.../macos-framework/
.../starkit-app/

https://gitlab.com/little-feather
https://gitlab.com/little-feather /macos-framework.git
https://gitlab.com/little-feather/starkit-app.git

Many thanks in advance to:

- Kevin Walzer [https://www.codebykevin.com/]
- Paul Obermeier [https://www.tcl3d.org/bawt]
- Alexander Schoepe [https://www.sowaswie.de/tcl-tk]

- https://stackoverflow.com/questions/53777217

Words of Wisdome

… nothing lasts forever

„The only constant in life is change”

Heraclitus

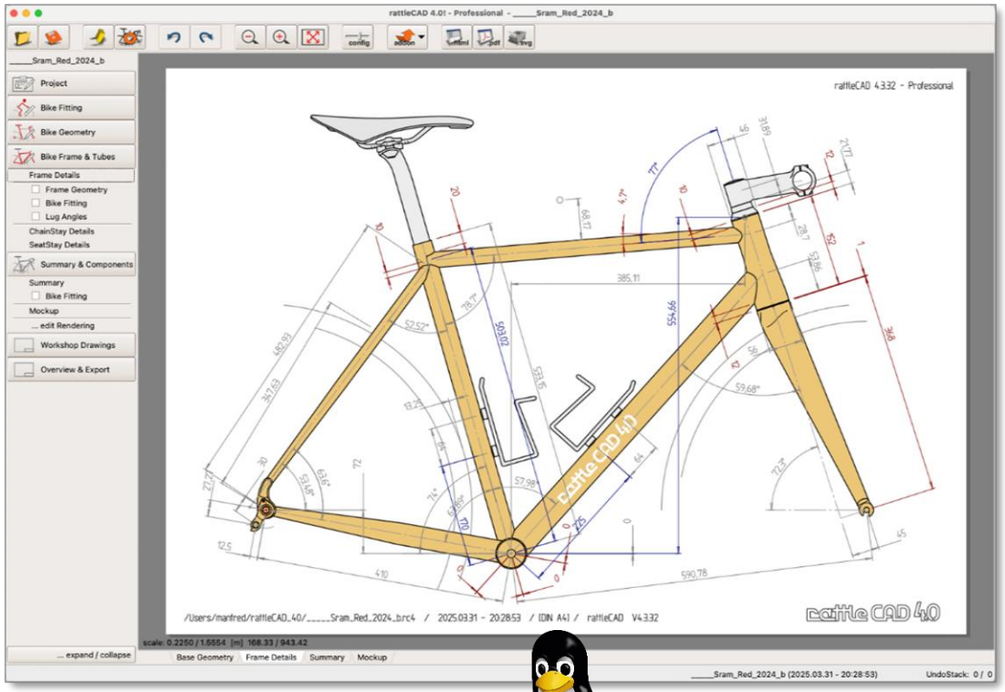
https://crossroadsantigua.org/the-only-constant-in-life-is-change-heraclitus/

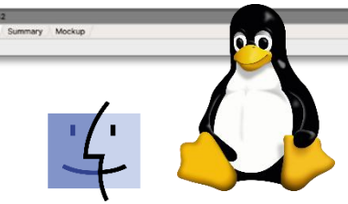
Quality Requirements

- stable
- extensible
- scaleable
- usable
  Start with a **Desktop-Icon**
  not by command-line

- free of Malware
  Codesigning
  MacOS – Notarization

- portable
  easy to install on different plattforms
  Linux (shell-Script & tar-Content)
  MS Windows (InnoSetup)
  MacOS (**Disk Image**, .dmg)





Mac OS

Windows 11

Troublemaker

… once a year

MacOS Sequoia

- Rollout
  - Sept, 16th, 2024
- Features (wikipedia.org)
  - AI
  - Safari
  - Window Control
  - iPhone Mirroring
  - Gatekeeper

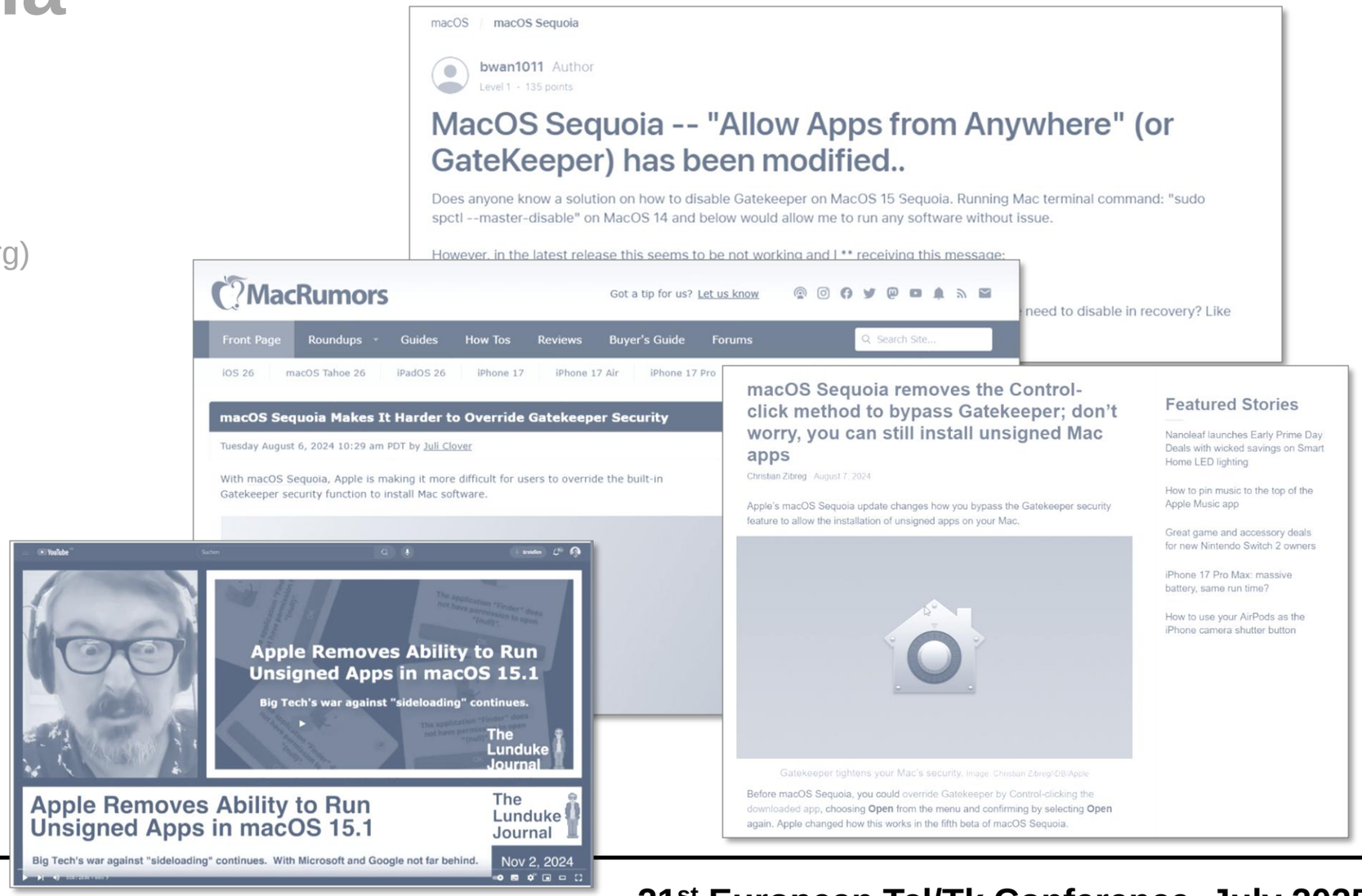
```
uname -a
            Darwin
            MinivonManfred.home
            24.5.0
            Darwin Kernel
            Version 24.5.0:
            Tue Apr 22 19:48:46 PDT 2025;
            root:xnu-11417.121.6~2/RELEASE_ARM64_T8103
            Arm64


sw_vers
            ProductName:            macOS
            ProductVersion:         15.5
            BuildVersion:           24F74
```

Gatekeeper

MacOS Sequoia

- Rollout
  - Sept, 16th, 2024
- Features (wikipedia.org)
  - AI
  - Safari
  - Window Control
  - iPhone Mirroring
  - **Gatekeeper**



… unsigned Apps works on
MacOS Sonoma (MacOS 14)

… ,but does not work on
MacOS Sequoia (MacOS 15)
anymore

Current Situation

```
rattleCAD-4.3.22-Professional.app
└── Contents
    ├── Info.plist
    ├── MacOS
    │   ├── rattleCAD-4.3.22-Professional.kit
    │   ├── rattleCAD_4.3-MacOS.sh
    │   └── vanillawish-20240307-macosx-x86_64
    ├── PkgInfo
    ├── PlugIns
    │   ├── extend__createMiter.tcl
    │   ├── extend__mockup3D.tcl
    │   ├── createMiter
    │   ├── forkReplacement
    │   ├── mockup3D
    │   └── stemBuilder
    └── Resources
        └── appIcon.icns
```

… this approach does not work anymore on MacOS Sequoia (MacOS 15)

rattleCAD-4.3.22-Professional.app/Contents/Info.plist

```
<plist version="1.0">
    <dict>
        <key>CFBundlePackageType</key>              <string>APPL</string>
        <key>CFBundleExecutable</key>               <string>rattleCAD_4.3-MacOS.sh</string>
        <key>CFBundleName</key>                     <string>rattleCAD-4.3-Professional</string>
        <key>CFBundleIdentifier</key>               <string>com.rattlecad.rattleCAD-Professional</string>
        <key>CFBundleVersion</key>                  <string>4.3.22</string>
        <key>CFBundleShortVersionString</key>       <string>4.3</string>
        <key>CFBundleIconFile</key>                 <string>appIcon</string>
        <key>CFBundleDevelopmentRegion</key>        <string>English</string>
        <key>CFBundleDisplayName</key>              <string>rattleCAD-4.3-Professional</string>
        <key>CFBundleGetInfoString</key>            <string>rattleCAD-4.3-Professional 22 Copyright 2024 … </string>
        <key>CFBundleInfoDictionaryVersion</key>    <string>6.0</string>
        <key>CFBundleSignature</key>                <string>unset</string>
        <key>LSEnvironment</key>    <dict>
                <key>APP_BUNDLER</key>              <string>com.rattleCAD.appBuilder 0.1</string>
            </dict>
        <key>LSHasLocalizedDisplayName</key>        <false/>
        <key>LSUIElement</key>                      <false/>
        <key>NSAppleScriptEnabled</key>             <false/>
        <key>NSHumanReadableCopyright</key>         <string>__NSHumanReadableCopyright__</string>
        <key>NSMainNibFile</key>                    <string>MainMenu</string>
        <key>NSPrincipalClass</key>                 <string>NSApplication</string>
```

rattleCAD-4.3.22-Professional.app/Contents/MacOS/rattleCAD_4.3-MacOS.sh

```
#!/bin/sh

cd "${0%/*}"

exec ./vanillawish-20240307-macosx-x86_64 rattleCAD-4.3.22-Professional.kit $@
```

Findings

Findings

MacOS Binaries:

- Mach-O Universal / Fat Binaries
  - Container for `arm64` & `x86_64` Binaries

```
otool -L .../Wish


./build/myApp.app/Contents/MacOS/Wish (architecture x86_64):

        @executable_path/../Frameworks/Tk.framework/Versions/8.6/Tk
        @executable_path/../Frameworks/Tcl.framework/Versions/8.6/Tcl
        /usr/lib/libSystem.B.dylib
        /System/Library/...


./build/myApp.app/Contents/MacOS/Wish (architecture arm64):

        @executable_path/../Frameworks/Tk.framework/Versions/8.6/Tk
        @executable_path/../Frameworks/Tcl.framework/Versions/8.6/Tcl
        /usr/lib/libSystem.B.dylib
        /System/Library/...
```

MacOS Notarization - Process:

- … does not allow Binaries with Payloads
- … does not allow Shell-Scripts

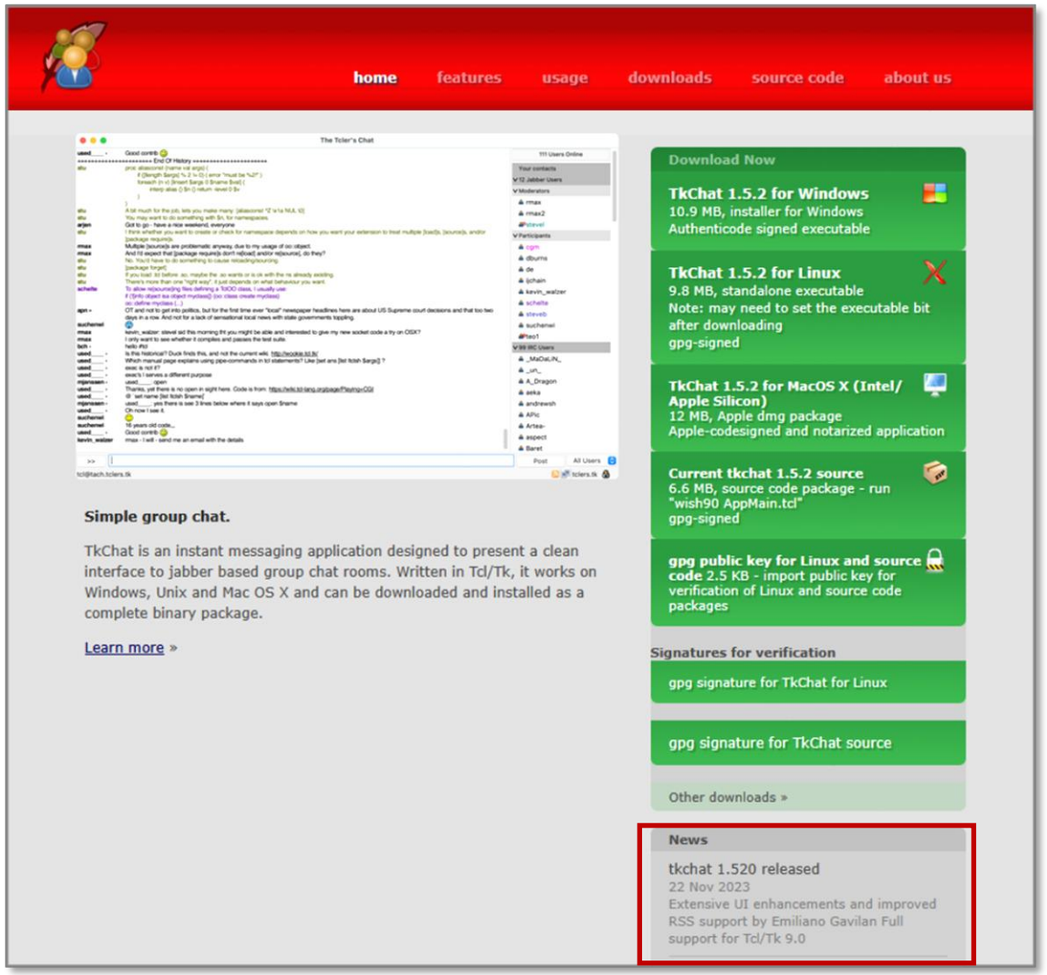
Conclusion:

- No
  - tclkit
  - zip-kit (e.g. vanillawish)
  - shell-Script

https://www.symbolcrash.com/2019/02/26/mach-o-universal-fat-binaries/

Solution Approach

Borrow Approach from: TkChat 1.5.2




```
TkChat_1.5.2__9.0.app
└── Contents
    ├── Frameworks
    │   ├── Tcl.framework
    │   ├── Tk.framework
    │   ├── libcrypto.3.dylib
    │   └── libssl.3.dylib
    ├── Info.plist
    ├── MacOS
    │   └── TkChat
    └── Resources
        ├── Credits.html
        ├── Scripts
        │   ├── AppMain.tcl
        │   ├── TkChat.ico
        │   ├── bin
        │   ├── lib
        │   ├── main.tcl
        │   ├── tclkit.ico
        │   └── tclkit.inf
        ├── Wish.icns
        ├── Wish.sdef
        └── tkchat.icns
```

Frameworks:
Contains any private shared libraries and frameworks used by the executable

http://tkchat.tcl-lang.org/

Solution: Build Tcl/Tk - Framework

tclbuild.sh

```
#!/bin/bash

macosxminver=10.9
sver=868cp
ver=8.6.8
mver=8.6
tclmver=$mver
tkmver=$mver
SRCDIR=$HOME/t/tcl${sver}
INSTLOC=$HOME/t/localB

if [[ $1 ≠ "" ]]; then
  INSTLOC=$1
fi

if [[ -d $INSTLOC ]]; then
  rm -rf $INSTLOC
fi
mkdir $INSTLOC

cd $SRCDIR

test -d build && rm -rf build

cd $SRCDIR
cd tcl${ver}
if [[ $? -eq 0 ]]; then
  f=library/init.tcl
  if [[ ! -f $f-orig ]]; then
    cp -pf $f $f-orig
  fi
  cp -pf $f-orig $

  make -C macosx \
      PREFIX="" \
      CFLAGS_OPTIMIZE="-O2 -mmacosx-version-min=${macosxminver}" \
      INSTALL_ROOT=$INSTLOC install

  cd $SRCDIR

  chmod u+w $INSTLOC/bin/tclsh${tclmver}
  install_name_tool -change \
      "/Library/Frameworks/Tcl.framework/Versions/${tclmver}/Tcl" \
      @executable_path/../Library/Frameworks/Tcl.framework/Versions/${tclmver}/Tcl \
      $INSTLOC/bin/tclsh${tclmver}
fi

cd $SRCDIR
cd tk${ver}
cd $SRCDIR
find $INSTLOC -type f -print0 | xargs -0 chmod u+w
exit 0
```

Find precompiled or compile Tcl/Tk Frameworks for macOS

Dec 14, 2018

https://stackoverflow.com/questions/53777217

Manual

Expected Outcome

starkit

myApp.kit

distributable App

myApp.app @ myApp-8.6.dmg

https://gitlab.com/little-feather/starkit-app.git

Process - Step by Step

myApp.kit

myApp-8.6.dmg

Process

1 - Build Tcl/Tk Framework

```
./install_8.6
├── Applications/...
├── Library/
│   ├── Frameworks/
│   │   ├── Tcl.framework/...
│   │   └── Tk.framework/...
│   └── Tcl/...
├── bin/
├── include/
└── man/
```

2 - Build Application Bundle

```
./myApp.app
└── Contents
    ├── Frameworks/
    │   ├── Tcl.framework
    │   └── Tk.framework
    ├── MacOS/
    │   └── Wish
    ├── Resources/
    │   ├── Scripts
    │   │   ├── bin
    │   │   │   └── myApp.tcl
    │   │   ├── lib
    │   │   │   ├── vfs1.4.3
    │   │   │   └── vlerq4.1
    │   │   ├── AppMain.tcl
    │   │   └── myApp.kit
    │   └── Tk.icns
    └── Info.plist
```

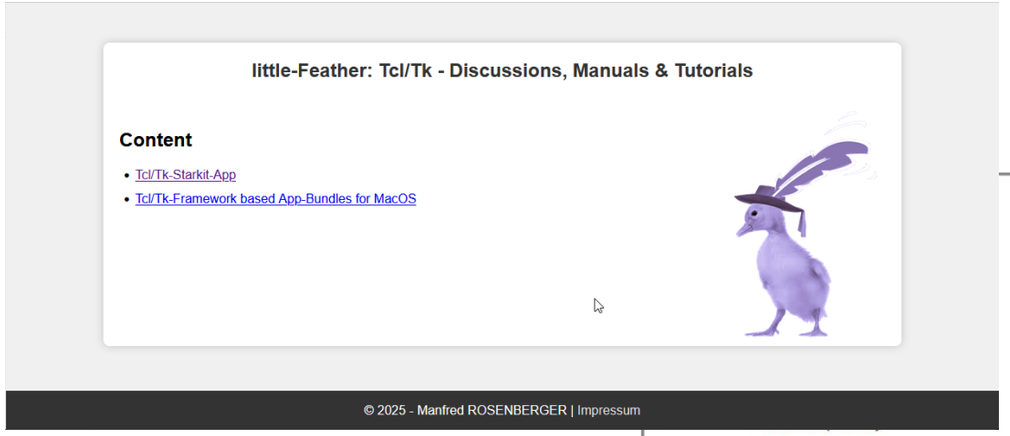
3 - Codesigning & Notarization

3.1 - Codesigning & Create .zip for Notarization

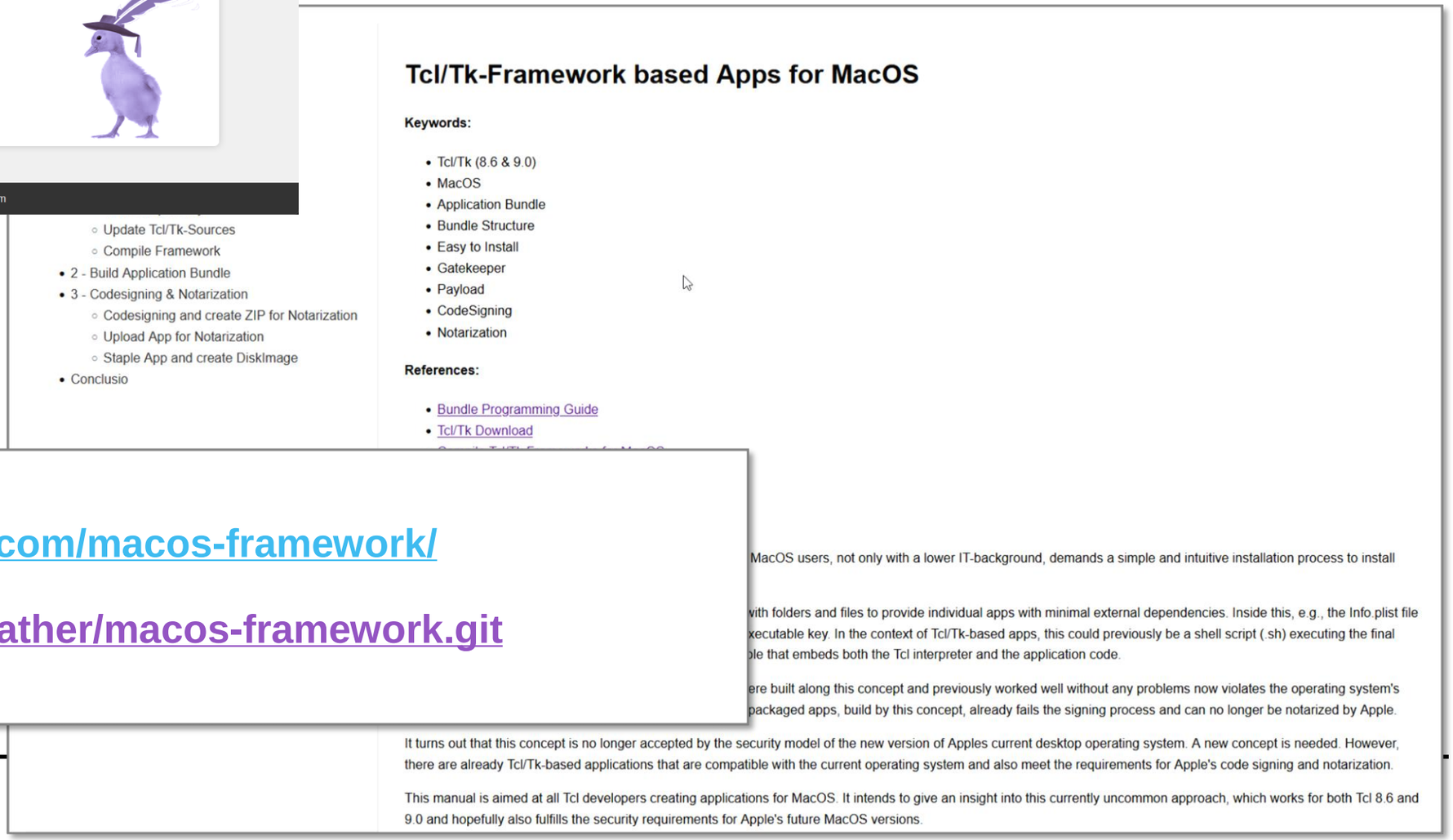
3.2 - Upload .zip for Notarization

3.3 - Staple App and create .dmg

Documentation



https://www.little-feather.com



https://www.little-feather.com/macos-framework/

https://gitlab.com/little-feather/macos-framework.git

1 - Build Tcl/Tk Framework

```
./buildTcl__8.6.16.sh
_____________________________________________________________________________



ver=8.6.16
mver=8.6



cd tcl${ver}

    make -C macosx \
        PREFIX="" \
        CFLAGS_OPTIMIZE=" -arch x86_64  -arch arm64  -mmacosx-version-min=${macosxminver}" \
        INSTALL_ROOT=$INSTLOC install

    install_name_tool -change \
            "/Library/Frameworks/Tcl.framework/Versions/${tclmver}/Tcl" \
            @executable_path/../Library/Frameworks/Tcl.framework/Versions/${tclmver}/Tcl \
        $INSTLOC/bin/tclsh${tclmver}



cd tk${ver}

    make -C macosx \
        PREFIX="" \
        CFLAGS_OPTIMIZE=" -arch x86_64  -arch arm64  -mmacosx-version-min=${macosxminver}" \
        INSTALL_ROOT=$INSTLOC install

    install_name_tool -change \
            "/Library/Frameworks/Tk.framework/Versions/${tkmver}/Tk" \
            @executable_path/../../../../Tk \
       $INSTLOC/Library/Frameworks/Tk.framework/Versions/${tkmver}/Resources/Wish.app/Contents/MacOS/Wish

    install_name_tool -change \
            "/Library/Frameworks/Tcl.framework/Versions/${tclmver}/Tcl" \
            @executable_path/../../../../../../../Tcl.framework/Versions/${tclmver}/Tcl \
        $INSTLOC/Library/Frameworks/Tk.framework/Versions/${tkmver}/Resources/Wish.app/Contents/MacOS/Wish
```

2 - Build Application Bundle

```
./buildApp__A__makeApp__8.6.sh
_____________________________________________________________________________________



APP_Name="myApp"
mver=8.6



cp -R       ${REPO_Dir}/myTemplate.app                                                  ./${APP_Name}.app

for libDir in ./${APP_Name}.app/Contents/Resources/Scripts/lib/__${mver}/*; do
    mv $libDir ./${APP_Name}.app/Contents/Resources/Scripts/lib
done
rm -Rf      ./${APP_Name}.app/Contents/Resources/Scripts/lib/__*



rsync -av   ./install_${mver}/Library/Frameworks                                        ./${APP_Name}.app/Contents
    # rsync -av   ./install_${mver}/Library/Tcl                                         ./${APP_Name}.app/Contents/Library



cp          ./install_${mver}/Library/Frameworks/Tk.framework/Resources/Wish.app/Contents/MacOS/Wish    ./${APP_Name}.app/Contents/MacOS/
cp          ./install_${mver}/Library/Frameworks/Tk.framework/Versions/Current/Resources/Tk.icns        ./${APP_Name}.app/Contents/Resources/



install_name_tool -change \
        @executable_path/../../../../Tk \
        @executable_path/../Frameworks/Tk.framework/Versions/${tkmver}/Tk \
    ./${APP_Name}.app/Contents/MacOS/Wish

install_name_tool -change \
        @executable_path/../../../../../../../../Tcl.framework/Versions/${tkmver}/Tcl \
        @executable_path/../Frameworks/Tcl.framework/Versions/${tkmver}/Tcl \
    ./${APP_Name}.app/Contents/MacOS/Wish



rm -Rf      ./${APP_Name}.app/Contents/Frameworks/Tk.framework/Versions/${tkmver}/Resources/Documentation
rm -Rf      ./${APP_Name}.app/Contents/Frameworks/Tk.framework/Versions/${tkmver}/Resources/Wish*
```

3.1 - Codesigning & .zip for Notarization

./buildApp__B__createZip.sh

```
REPO_Dir=$(pwd)
APP_Name="myApp"

cp -Rf ${APP_Name}.app ./build/${APP_Name}.app

vtool -show-build ./build/${APP_Name}.app/Contents/MacOS/Wish
otool -L          ./build/${APP_Name}.app/Contents/MacOS/Wish

xattr -cr ./build/${APP_Name}.app

python3 -mmacholib standalone ./build/${APP_Name}.app

tkmver=$(echo $(ls -l ${APP_Name}.app/Contents/Frameworks/Tk.framework/Versions/Current) | awk -F ' -> ' '{print $2}')

rm -rf ./build/${APP_Name}.app/Contents/Frameworks/Tcl.framework/Versions/${tkmver}/Resources/Documentation
rm -rf ./build/${APP_Name}.app/Contents/Frameworks/Tk.framework/Versions/${tkmver}/Resources/Documentation

find ./build/${APP_Name}.app  -name "*.a"       -exec rm -rf {} \;
find ./build/${APP_Name}.app  -name "*debug"    -exec rm -rf {} \;
find ./build/${APP_Name}.app  -name "*.sh"      -exec rm -rf {} \;

rm -rf ./build/${APP_Name}.app/Contents/Frameworks/Tk.framework/Versions/${tkmver}/Resources/Wish.app
rm -rf "./build/${APP_Name}.app/Contents/Frameworks/Tk.framework/Versions/${tkmver}/Resources/Wish Shell.app"

security unlock-keychain login.keychain

find ./build/${APP_Name}.app -type f  -name "*.bundle"  -exec \
    codesign  --verbose  -f ...

codesign  --verbose  -f ... ./build/${APP_Name}.app/Contents/Frameworks/Tk.framework/Versions/Current
codesign  --verbose  -f ... ./build/${APP_Name}.app/Contents/Frameworks/Tcl.framework/Versions/Current
codesign  --deep  --verbose=2 ... \
            --options runtime  \
            --entitlements ${REPO_Dir}/sources/entitlements.plist \
       ./build/${APP_Name}.app

/usr/bin/ditto -c -k --keepParent ./${APP_Name}.app ../${APP_Name}.app.zip
```

- **otool**
  … displays specified parts of object files or libraries.

- **vtool**
  … displays and edits build and source version numbers embedded in the Mach-O(5) file format.

- **xattr**
  … display and manipulate extended attributes of one or more files, including directories and symbolic links.

- **python -mmacholib**
  … can be used to analyze and edit Mach-O headers

- **codesign unlock-keychain**
  … access Developer-ID

- **codesign**
  …. codesigning utility

- **ditto**
  … dopy directory hierarchies, create and extract archives

3.2 - Upload App.zip for Notarization

buildApp__C__submitZip.sh

```
APP_Name="myApp"
SIGN_Profile="myNotarizationProfile"


codesign -dvv --entitlements -  ./build/${APP_Name}.app


spctl    --assess --verbose     ./build/${APP_Name}.app


xcrun notarytool submit \
             "./${APP_Name}.app.zip" \
           --keychain-profile $SIGN_Profile \
           | tee log_c_submit.log
```

- **codesign**
…. codesigning utility

- **spctl**
… manage the security assessment policy subsystem

- **xcrun**
… run Xcode tools

3.3 - Staple Ticket to App and create DiskImage

buildApp__D__stapleApp.sh

```
APP_Name="myApp"
SIGN_Profile="myNotarizationProfile"


SUBMIT_ID=$(grep "id:" log_c_submit.log | awk '{print $2}' | sort | uniq)
echo "        ${SUBMIT_ID}"


xcrun notarytool log "$SUBMIT_ID" \
             --keychain-profile $SIGN_Profile \
           | tee log_d_status.log


head -10 log_d_status.log

SUBMIT_Status=$(grep \"status\": log_d_status.log | grep Accepted | awk '{print $2}')
echo "        ${SUBMIT_Status}"


xcrun stapler validate          ./build/${APP_Name}.app

xcrun stapler staple            ./build/${APP_Name}.app

xcrun stapler validate          ./build/${APP_Name}.app

spctl -vvv --assess --type exec ./build/${APP_Name}.app


hdiutil create \
              -volname "${DMG_Name}" \
              -srcfolder ./${APP_Name}.app \
              -ov \
              -format UDZO \
              ../${DMG_Name}.dmg
```

- **xcrun**
  … run Xcode tools

- **spctl**
  … manage the security assessment policy subsystem

- **hdiutil create**
  … create and manipulate disk images

Result

distributable App

myApp.app @ myApp-8.6.dmg

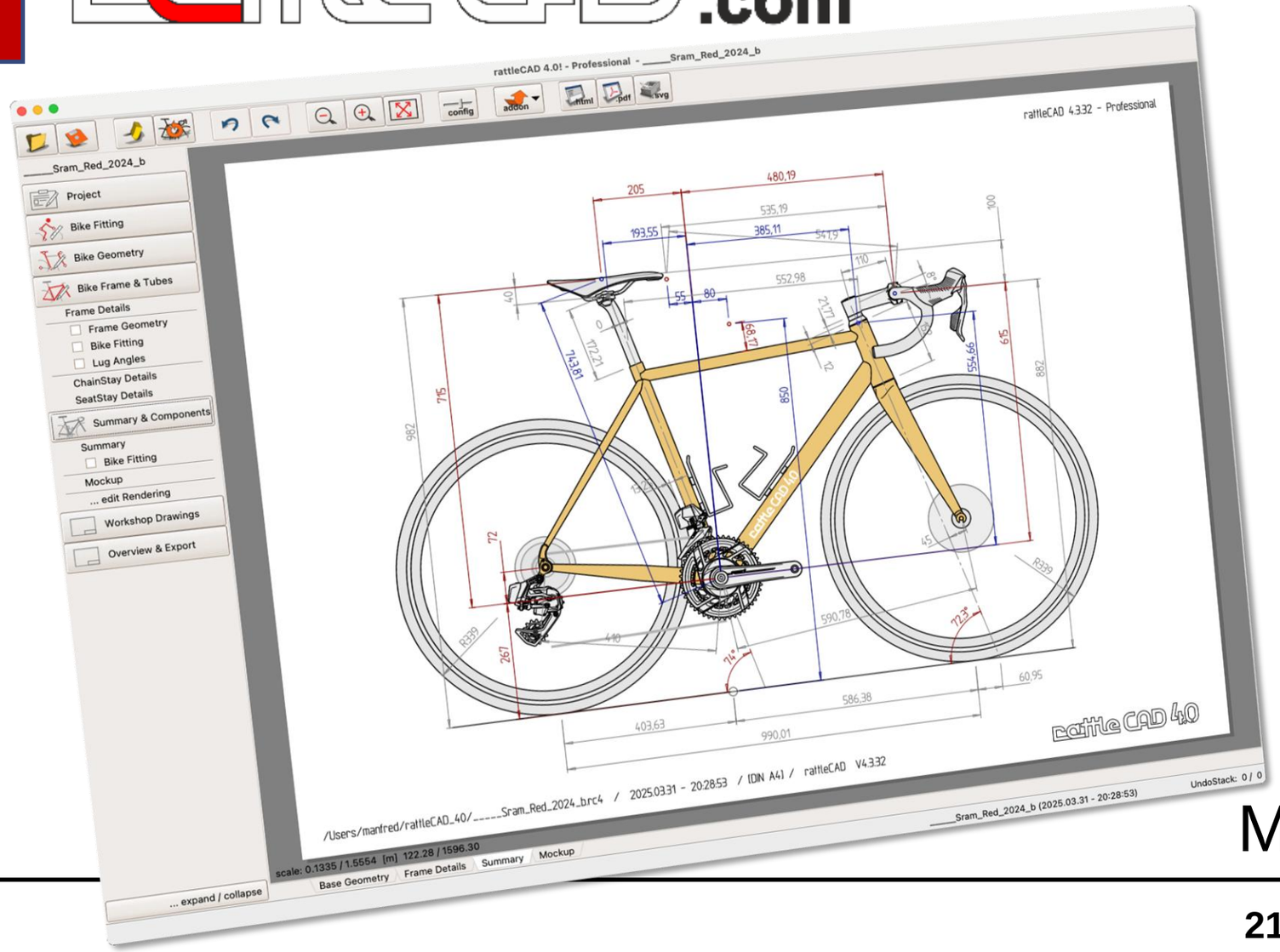
https://www.little-feather.com/
https://www.little-feather.com/macos-framework/
https://www.little-feather.com/starkit-app/

https://gitlab.com/little-feather/macos-framework.git
https://gitlab.com/little-feather/starkit-app.git

Questions ???

Thanks!

rattleCAD.com



Manfred **ROSENBERGER**

21st European Tcl/Tk Conference, July 2025