logo

QE Framework – Release Procedure

Andrew Rhyder  
Andrew Starritt

12 February 2018

Copyright (C) 2013, 2014, 2015,2018 Australian Synchrotron.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation;with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the associated document ‘QE Framework – QEGui and User Interface Design’.

Contents

[Introduction 2](#__RefHeading___Toc184_1630634933)

[Procedure 2](#__RefHeading___Toc186_1630634933)

# Introduction

This document contains details on creating releases of the QE framework. It is intended to be used by those developing and maintaining the QE framework, not GUI or application developers using the QE framework.

This document, as of February 2018, has been significantly updated to reflect the relocation of the public home of QE framework from SourceForge to Github. This document has been internally relocated from the *qeframework* repository to the *qtepics.github.io* repository.

# Procedure

1. Ensure that you are allowed to commit changes to <https://github.com/qtepics>, if not please consult site administrator.
2. Ensure all documentation is up to date (.PDFs/.HTML generated from .DOCX for example) and checked into Perforce.
3. Ensure any local changes in Perforce required for the release are commited to the appropriate git repository and pushed to the GitHub site.
4. Run *qtepics.github.io/trunk/tools/qe\_git\_test\_build –all*  within at least one environment to ensure GitHub head files are at least build-able. Note: a successful build does not mitigate the need for through testing of any new features. If build fails, investigate, fix and restart procedure.
5. Using your preferred Perforce client workspace, checkout (mark for edit) the following file:

//ASP/tec/gui/qeframework/trunk/  
 qeframeworkSup/project/common/QEFrameworkVersion.h

Update version numbers to new release version. Update QE\_VERSION\_MAJOR, QE\_VERSION\_MINOR, QE\_VERSION\_RELEASE, and QE\_VERSION\_STAGE. The later should be set to QE\_VERSION\_STAGE\_PRODUCTION,

Commit this version change to Perforce.

1. Using your preferred Perforce client workspace, checkout (mark for edit) the following file:

//ASP/tec/gui/qtepics.github.io/trunk/source/release\_notes.docx  
//ASP/tec/gui/qtepics.github.io/trunk/release\_notes.html

Update the release notes to create a new section for the new release. This will provide a summary of the changes since the last release. While in principle this is available from a number of sources, the easiest is the GitHub logs, e.g.

git log r3.5.2..HEAD

provides a summary of the changes since the last release (tagged r3.5.2 in this example).  
This is summary can then be used to update the release notes file. Note this will have to be run for each of the repositories, not just qeframework.

Save the modified release notes document and also save as HTML to update the html version.

Commit the updated release notes files to perforce.

1. Create Perforce component labels, E.g., for each component...

cd …/tec/gui/qeframework  
create\_component\_label qeframework/3-5-3-01 trunk  
create\_release\_label qeframework/3-5-3-01

Note the additional -01 – Used for internal only releases.

1. Commit the three new files (version, release notes docx and release notes html) to the qeframework and qtepics.github.io repositories.
2. Tag *each* git repository (except qeBinaries) with the new release tag.  
   This should be consistent with the new major, minor and release versions specified in QEFrameworkVersion.h as per step (e) above, then push the latests files and tags to the GitHub site, e.g.:

git tag r3.5.3  
git push  
git push –-tags

1. Run *qtepics.github.io/trunk/tools/qe\_git\_test\_build -ffmpeg in all available environments.* This not verify that the previous commit did not break anything, it now allows you to run  
   *qtepics.github.io/trunk/tools/qe\_generate\_prebuilt* . This will generate a tar archive of the just released versions of qeframework and qegui. The file generated will be of the form:

centos7-qe3.5.3-qt5.9-r3.15.5-linux-x86\_64.tar.gz

Verify that the qe version is as expected. Copy the file to the local qeBinaries git . and add to the repository.

git add centos7-qe3.5.3-qt5.7-r3.15.5-linux-x86\_64.tar.gz  
git commit -m “some message”  
git push

Note: *qe\_git\_test\_build* and *qe\_generate\_prebuilt* is Linux centric. Equivalent Windows tools do not exist yet.

1. email tech-talk.