

ORKA-ILCRoot Releases

Brett Viren

December 11, 2013

Contents

1 Overview	1
2 Versions	2
3 Branching and tagging	2
4 Release procedure	3
4.1 Release contents	3
4.2 Release branch	3
4.3 Release candidate tag	3
4.4 Build configuration	4
4.5 Iteration	4
5 Redmine	4
This document describes procedures and conventions for making ORKA-ILCRoot releases.	

1 Overview

A release identifies specific versions of the software and its dependencies which are expected to meet some criteria. For ORKA-ILCRoot the following release types are identified:

feature a version which has had some additional functionality added since the last release

bug fix a version which fixes bugs found in a prior release

candidate a release made towards producing a feature or a bug-fix release.

The release involves identifying specific commits made to two, related software repositories:

`orka-ilcroot` holding the ORKA-ILCRoot software

`orka-ilcroot-worch` holding the worch build configuration

2 Versions

ORKA-ILCRoot adopts a version naming scheme of: `vX-Y-Z` with each digit interpreted as indicating, in decreasing order:

X an overall architecture version. This digit is incremented to indicate large or backwards-incompatible changes have been made.

Y new feature development. Incremented in the course of "normal" development of new features.

Z a bug fix. No new features added. The first feature release is simply a bug-fix release with `Z=0`.

When any higher order digit is incremented all lower order digits are set to 0.

3 Branching and tagging

Releases are recorded as git branches tags in the two repositories. Both are expected to follow these branch conventions for the production of releases:

`master` used for normal development

`vX-Y` branch from `master` created initially for feature release `vX-Y-0` and holding all subsequent bug fix releases (`Z>0`)

Any additional branching policy is allowed in personal, cloned repositories but it is discouraged to push any other branches to the central Redmine git repositories.

A release is indicated by a git tag in either of the two repositories. The `orka-ilcroot` repository has release related tags of the form:

`vX-Y-Z` a final release

`vX-Y-Z-rcN` the N'th release candidate made prior to `vX-Y-Z`

It is expected that initial validation of any release may fail and so one or more release candidates (RC) are made before a final release. The first RC that successfully validates becomes the final release. Note, the `rcN` label starts at `rc0` and counts up as high as required. The version string `vX-Y-Z` is considered greater than `vX-Y-Z-rcN` for all N.

The tags on `orka-ilcroot-worch` are meant shadow the tags placed on `orka-ilcroot`.

`vX-Y-Z+buildM` the M'th iteration on the configuration to build final release `vX-Y-Z` of ORKA-ILCRoot.

`vX-Y-Z-rcN+buildM` the M'th iteration on configuration to build release candidate `vX-Y-Z-rcN`.

4 Release procedure

With the above understanding, the release procedure is as follows:

4.1 Release contents

Consensus is first reached among developers and release manager regarding:

- identifying the "target commit" in the `orka-ilcroot` git repository `master` branch
- the `vX-Y-Z` version string for the release
- the versions of all dependencies

The target commit is the one that is expected to work and contain all desired features or bug fixes for the release.

4.2 Release branch

If the release is a new feature, a `vX-Y` branch is created at the target commit. A branch of the same name is made in `orka-ilcroot-worch` at an appropriate location.

4.3 Release candidate tag

A tag `vX-Y-Z-rc0` is made on the target commit.

4.4 Build configuration

Given the RC tag the build configuration from `vX-Y` branch is modified to build the RC. Any required changes needed to build the dependencies are also made. Local validation is performed. If failures occur which can be fixed by build configuration changes they are simply made in place. Once successful the `orka-ilcroot-worch` is tagged with `vX-Y-Z-rc0+build0` and a public RC is made for more extensive testing.

4.5 Iteration

If the target commit is found to fail validation, fixes to `orka-ilcroot` are committed to the `vX-Y` branch, a new RC is tagged and a new build configuration is attempted. The first iteration would result in `vX-Y-Z-rc1+build0`.

5 Redmine

When a new release is made its name should be added to ORKA-ILCRoot's Redmine as a "Version". Under the **Settings** area find the "Versions" tab and click "New version" and fill in the form. For **Name** use the targeted `vX-Y-Z` string. It will then become available for Issues in the "Target version" drop-down. Planned release can also be listed in order to file issues against them.