

# Repositories and Releases

The OpenJFX project maintains 2 main development repositories. We utilize the "Branching with Clones" method as described in [A Guide to Branching in Mercurial](#) by Steve Losh. The "master" repo is the one from which each weekly promoted build is made. You can think of it as the "golden master". The master repository must always be buildable. We never work directly against master. Instead, developers push their fixes into the "-dev" repository.

We have one "mainline" development repository, the "jfx-dev" repository. The URL for this repo is listed here:

- <http://hg.openjdk.java.net/openjfx/jfx-dev/rt>

## Naming / History

*The following is only for historical interest. There is no longer a separate OpenJFX release. As of JDK 8, we follow the JDK release numbers precisely.*

Releases of OpenJFX are based on street names or neighborhoods in San Francisco. This tradition started with JavaFX 1.1 and carries on to this day. The names we have used for releases past, present, and future are listed here.

Release	Version
Franca	1.1
Marina	1.2
Soma	1.3
Presidio	2.0
Lombard	8.0
Van Ness	9.0

Some have questioned, why bother with release names, why not just stick with numbers? Long story short, it is so that when release numbers change on us, we don't have to adjust everything in the project (such as tags in JIRA).

From JavaFX 1.0 until 1.3, JavaFX was developed internally at Sun as proprietary software. Each of these so-called minor releases were actually major and occurred quickly over a short period of time. Each release contained new functionality and the not-so-occasional backward incompatible change.

JavaFX 2.0 was the first major release from Oracle. It was the first release that was based entirely on Java, and where the JavaFX Script language had been removed. A lot of the backend implementation code was already in Java (JavaFX 1.x was half JavaFX Script and half Java), but all of the JavaFX Script code had to be rewritten as Java. Most of the API carried over intact, but the implementation was essentially brand new.

Starting with JavaFX 2.0, Oracle began shipping JavaFX with the Oracle JDK and later the Oracle JRE. Oracle also announced that JavaFX would be open sourced, and would move toward becoming a standard part of Java. The release number for Lombard was bumped up from 3.0 to 8.0 in order to reflect this change. Lombard works with Java SE 8, and is co-bundled with Oracle's JavaSE 8 implementation (both the JDK and JRE). To help keep the version numbering consistent, we decided to bump up the version number.

From here, every release of JavaFX will map exactly to a corresponding release of Oracles JavaSE. Specifically the [OpenJDK Milestones](#) and release dates correspond to OpenJFX as well. This makes it easy to know into which release you will find fixes or features from OpenJFX. We typically have not given names to update releases (2.1, 2.2, etc).