# X.Org Version Numbering Schemes

The XFree86 Project, Inc

Updated for X.Org by Keith Packard,
Kevin E. Martin, and Alan Coopersmith

November 2010

**Abstract**

X.Org has adopted the same basic numbering scheme used by the XFree86 Project, Inc. for their releases. The actual numbers are different, but the basic scheme is the same. This document reflects the policy that X.Org uses.

# Table of Contents

odule ersions .....
eleases, e elopment treams, and ranches .....
urrent ersion umbering cheme .....
e elopment ranch .....
table ranch .....
Finding the X.Org X er er ersion From a lient .....

# Module Versions

tarting ith the X11 . release, each module has its o n ersion number. For those ithout a natural starting point, the ersion numbers started at 1. . For instance, the X11 . release included the orgserver 1. module. s modules are released independently from the rest of the indo system, the module ersion is the most accurate source of ersion information. For instance, there are many X ser er releases in a year, but generally only one indo system release, so an X ser er ersion number such as 1. . is more informati e than the X11 . ersion for the indo system atamari release.

nfortunately, up through the X ser er 1. release, the X ser er used the indo ystem ersion hen reporting its ersion number in log les, the - ersion option, and the connection setup string displayed by dpyinfo . This as corrected ith X ser er 1. , hich caused the isible ersion number string to appear to jump bac ards from . to 1. .

# Releases, Development Streams, and Branches

X.Org has t o release branches for the X ser er soft are, and se eral other modules ith acti e ongoing de elopment. First is the trun of the git repository. This is the main de elopment stream, here all ne or and or for future releases is done.

econd is the stable bug branch for the latest full release. It is created around the time of the release. The branch ill be named for the release ersion, such as `server-1.9-branch` for the X ser er 1. . series of releases. Fi es for bugs found in the release ill be added to this branch as ell as the trun , and updates to this release if any ill be cut from this branch. imilar stable branches are present for pre ious full releases.

The X.Org Foundation is planning to ma e full releases from the main de elopment stream at regular inter als in the 6-1 month range. The feature free es for these releases ill usually be - months before the release dates. This general plan is a goal, not a binding commitment. The actual release inter als and dates ill depend to a large degree on the resource a ailable to X.Org. pdate bug releases ill be made on an as-re uired basis, depending also on the a ailability of resources, and ill generally be limited to serious bug and security es. e features ill not usually be added in update releases.

side from actual releases, snapshots of the acti e release branches are tagged in the git repository from time to time. ach such snapshot has an identi able ersion number.

# Current Version Numbering Scheme

tarting ith the main de elopment branch after X11 6. , the X.Org ersions are numbered according to the scheme outlined here.

The ersion numbering format is `M.m.P.s`, here *M* is the major ersion number, *m* is the minor ersion number, *P* is the patch le el, and *s* is the snapshot number. Full releases ha e *P* set to ero, and it is incremented for each subse uent bug release on the post-release stable branch. The snapshot number *s* is present only for bet een-release snapshots of the de elopment and stable branches.

# Development Branch

Immediately after forming a release stable branch, the patch le el number for the main de elopment branch is bumped to , and the snapshot number is reset. The snapshot number is incremented for each tagged de elopment snapshot. The git tag for snapshots is `xorg-server-M.m.P.s` . hen the de elopment branch enters feature free e, the snapshot number is bumped to . stable branch may be created for the ne t full release at any time after the feature free e. hen it is, the branch is called `server-M.m-branch` . The snapshot number is incremented from there until the release is nalised. ach of these snapshots is a release candidate . hen the release is nalised, the minor ersion is incremented, the patch le el is set to ero, and the snapshot number remo ed.

ere s an e ample hich sho s the ersion number se uence for the de elopment leading up to ersion 1.8

`1.7.99.1` The rst snapshot of the pre-1.8 de elopment branch.

`1.7.99.23` The t enty-third snapshot of the pre-1.8 de elopment branch.

`1.7.99.900` The start of the 1.8 feature free e.

`1.7.99.903` The third 1.8 release candidate.

`1.8.0` The 1.8 release.

`1.8.99.1` The rst pre-1. de elopment snapshot, hich is the rst main branch snapshot after creating the 1.8 stable branch.

# Stable Branch

fter a full release, the stable branch for the release ill be maintained ith bug es and important updates until the ne t full release. ny snapshots on this branch are considered release candidates, hich is indicated by setting to a number abo e . The snapshot number is incremented for each release candidate until the update release is nalised. The patch le el alue *P* is incremented for each update release.

ere s an e ample hich sho s a ersion number se uence for a 1.8. stable branch

`1.8.0` The 1.8 release.

`1.8.0.901` The rst pre 1.8.1 snapshot.

`1.8.0.903` The third pre 1.8.1 snapshot, also no n as the third 1.8.1 release candidate.

`1.8.1` The 1.8.1 release.

`1.8.1.901` The rst pre 1.8. snapshot.

`1.8.2` The 1.8. release.

# Finding the X.Org X Server Version From a Client

The X.Org X ser ers report a `VendorRelease` alue that matches the X.Org ersion number. There ha e been some cases of releases here this alue asn t set correctly. The rules for interpreting this alue as ell as the no n e ceptions are outlined here.

s noted abo e, the ersion reported by `VendorRelease` changed from the indo system ersion to the X ser er ersion starting in the org-ser er 1. release.

For all X.Org de elopment and release ersions using this numbering scheme, the `VendorRelease` alue is *MMmmPPsss*. That is, ersion *M.m.P.s* has `VendorRelease` set to `M * 10000000 + m * 100000 + P * 1000 + s`.

The follo ing is a code fragment ta en from `xdpyinfo.c` that sho s ho the `VendorRelease` information can be interpreted.

```
    if (strstr(ServerVendor(dpy), "X.Org")) {
        int vendrel = VendorRelease(dpy);

        printf("X.Org version: ");
 printf("%d.%d.%d", vendrel / 10000000,
        (vendrel /   100000) % 100,
        (vendrel /     1000) % 100);
 if (vendrel % 1000) {
     printf(".%d", vendrel % 1000);
 }
    }
```