

## Software Release and Support Policy

This document describes the **XRootD** policy for managing its software releases and release support policy. Use this information to help plan when to perform major **XRootD** software updates at your site.

### Software releases

**XRootD** software releases are organized into **major**, **minor** and **patch** versions, with the intent that installing minor version releases with the same major version do not take long to perform, cause significant downtime, or break dependent software. New major versions can be more disruptive, and may substantially change or remove software components. Releases are assigned version numbers, such as "3.2.1", where:

- The first number designates a **major version**. Major versions may introduce binary incompatibility with previous major versions and may require code dependent on libraries in the new major version to be recompiled. Generally, such requirements are limited to code that enhances **XRootD** functionality (e.g. plug-ins). User application code that only uses public API's should continue to work unchanged. Consequently, major versions are infrequent and are introduced approximately every 5 years.
- The second number increments within the major version and designates a minor version. Minor versions introduce new features within a major version. They are binary compatible with all versions within the major version and occur as often as needed to address community needs. On average, there are about four minor versions per year.
- The last digit increments whenever changes are applied to a minor version to fix problems. These occur at random frequency, as often as needed to fix problems. Since patch versions represent the minimum change necessary to fix a problem they provide the forward path for problem resolution.
- When the first number increments, the second and third numbers are reset to zero and when the second number increments the third number is reset to zero.

## Release support policy

**XRootD** supports at most two concurrent major versions, **current** and **previous**, as follows:

- When a major version is introduced it becomes the **current** version. The **previous** version is the highest numbered version in the immediate prior major version. The previous version is fully supported for six months and certain changes in the current version may be back ported to the previous version as long as the back port does not involve extensive changes or introduce incompatibilities in the previous version. After six months the previous version is supported only to the extent to fix critical software and security bugs. Thus the highest numbered previous version is considered supported for a full year after a new major version is released.
- When a minor version is introduced it becomes the platform to fix any problems in any prior version within that major version. However, all prior versions within the major series are still supported for problem resolution for a full year after the new minor version is released.
- When a patch version is introduced it does not change the support level for any previous patch version. That means the support policy is only based on major and minor versions as described above.
- Sites are expected to migrate to a supported version within the major version that fixes a reported problem; which may require migrating to a new minor version. This means that problems reported in any supported prior version are always fixed in the most recent minor version (i.e. highest numbered version within a supported major version).

## Support policy for non-public API's

When code uses a non-public API (i.e. any definition in `xrootd/private`) it is considered not supported. Non-public API's are made available for historical reasons. These API's may change at any time and may require code dependent on such private API's to be recompiled or, at worst, rewritten whenever a new release of any kind becomes available. Consequently, they are only installed when specifically requested. The use of private API's is highly discouraged.

## End of life policy

- End of life for a major version is announced on the **XRootD** web site at least 4 months in advance.
- There is no official announcement when a minor version reaches end of life because migrating to a new minor version should be no more difficult than migrating to a new patch version.
- When a version reaches end of life it means that the **XRootD** team no longer updates the software, fixes issues, or troubleshoots installations with such versions.

## Backward compatibility

- **XRootD** strives to keep configuration files backward compatible. However, incompatibilities may occur when a site mixes old directives with newly introduced directives. In the absence of conflicting directives, configuration files should be backward compatible for two major versions (typically 10 years). Incompatibilities in pre-existing configuration files are considered bugs and should be immediately reported so that they can be fixed.
- **XRootD** always keeps backward compatibility between clients and servers for at least 3 major versions (typically 15 years). However, older clients are restricted to the features available in their respective version and newer clients are restricted to the features provided by older servers. Consequently, they only interoperate within their intersecting feature sets subject to any further restrictions that may be enforced by a site for security purposes.

## Software availability

The **XRootD** web site provides source and RPM level access to at least two major versions. Access to versions from other distributors (e.g. EPEL and OSG) is dictated by the policies of their respective organizations.