

Versioning & Support Policy

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	-----------------------------------------------	--------------------------------	------------------------------------------	-----------------------------------------	----------------------------------------------

Grouper Versioning and Support Policy

The Grouper project follows these guidelines on support of previous releases.

 Semantic versioning: In 2023 Grouper adopted "semantic versioning". (similar to [semver](#)). See below for details.

- [Versioning](#)
- [Support](#)
- [Semantic versioning \(v4+\)](#)
- [See Also](#)

Versioning

The Grouper development team strives to maintain permanent backward compatibility of the core API / WS across all releases.

Support

Current and the previous releases will receive active support and security updates by the core development team. As of 2024, that means

Overview

- v4 and v5 releases are maintained by the core development team.
- v5 is under development and will contain enhancements, bug fixes, and security fixes
- v4 (v2.6 renamed) receives non-risky and important bug fixes, and security fixes
- Releases in v4 and v5 are linear, once a minor version number exists, no previous minor versions will be released. e.g. when v4.5.0 is released, there will be no more v4.4.* releases
- Releases three months old are labeled "expired" to encourage quarterly upgrades
- Previous releases before v4 are not supported

Advice for Implementers

- Implementers should upgrade to a stable recent build of their supported version and plan to upgrade quarterly.
- Security fixes will prompt a new container build and the builds are linear so it is important to stay current.
- The OS packages are in the container so it either requires a new container from Grouper or a new derived image to patch/upgrade the OS and related files.

More on the release cycle

- Before releasing all containers, security vulnerabilities listed in maven central will be addressed by upgrading those third party libraries.
- Once a newer Long Term Support (LTS) version is announced (e.g. an even major version), then the previous LTS version will be end of life in approximately six months.
- v2.5 was end-of-life on May 1, 2023.
- Supported versions will be released monthly or more frequently.

References used in writing this policy

-Shibboleth Project: <https://wiki.shibboleth.net/confluence/display/SHIB2/ProductVersioning>

-Linux Kernel Project: <https://www.kernel.org/category/releases.html>

-Samba Project: <http://www.samba.org/samba/devel/>

Semantic versioning (v4+)

In 2023, Grouper adopted "semantic versioning". (similar to [semver](#))

The development, upgrades, and releases are not really affected, but the numbering of the versions has changed.

- v2.5.67+ is the legacy versioning strategy
- v2.6.20 is renamed to v4.0.0 - LTS - Long term support
- previously discussed v2.7.0 will be named v5.0.0 for the enhancement version and v6.0.0 for LTS (when it is done)
- previously discussed v3.0.0 will be developed on v7.0.0, stable on v8.0.0 (odd is the enhancement branch, even is the LTS branch)

We stopped using a fourth number. We had previously used the fourth number for a container release where the maven version does not change, but we will just use the next third number, and skip that version in Maven.

Pure semantic versioning only applies to Long Term Support (LTS) releases. The latest Grouper major version has new enhancements until it is complete and stable. The major version will not be incremented until the enhancements are done (e.g. over a year).

For instance if this is an LTS version: v6.1.8

6 = major version (incompatible changes will bump this number)

1 = minor version (backwards compatible changes)

8 = patch (backwards compatible bug/security fixes)

For instance if this is the enhancement version: v5.1.6

5 = major version (will stay at 5)

1 = minor version (incompatible changes or backwards compatible changes)

6 = patch (backwards compatible bug/security fixes)

If there is a security fix in an LTS version that causes incompatible changes, it will be documented. For instance if we need to change a 3rd party library we will do so in a way that is the least risky and disruptive, but it might require some configuration changes or change in functionality. The defaults for all configuration should not change.

- Incrementing a major version will have upgrade instructions.
 - These will be the list of instructions of all the minor versions up to the current version of the next major version.
 - e.g. when upgrading from v6.5.13 v8.8.21
 - Follow instructions for v6.6?, v6.7?, v7.0, v7.1, ..., v7.x, v8.0, v8.1, v8.2, ... , v8.8
- Incrementing a minor version generally means it is a "breaking change" and some default behavior is different.
- Incrementing a patch version of an LTS release generally means it is a non risky upgrade.

The release path for a major version is still linear.

- For instance, if you are on v6.5.13, and there is a security fix
 - If the latest v6 version is v6.8.4, and this is backwards compatible, the fix will be released in v6.8.5
 - If the latest v6 version is v6.8.4, and this is not backwards compatible, the fix will be released in v6.9.0

See Also

[Grouper Security Patches](#)

[Grouper 2.5+ packaging and versioning](#)