

# Semantic Versioning

The CManage Project adopted [Semantic Versioning](#) with Registry v0.9 (so really v1.0.0). Since then, there has been some confusion as to when to rev the major version number, since "when you make incompatible API changes" (the criteria for a new MAJOR version) and "when you add functionality in a backwards compatible manner" (the criteria for a new MINOR version) are not exactly the same standard.

To improve clarity, as of Registry v4.0.0 and Match v1.0.0 the following (stricter) definitions will be used:

- The MAJOR version number is incremented when the release includes a change that is not *drop in place*.
- The MINOR version number is incremented when there are no changes included that are not *drop in place*.
- The PATCH version number is incremented when the only changes are backwards compatible bug fixes.

The term *drop in place* refers to the concept that the deployer should be able to drop the new code in place and not encounter surprises. (This is conceptually similar to "adding functionality in a backwards compatible manner".) Changes are not *drop in place* when they affect the behavior of the system in a way that is unexpected by an administrator, user, or automated (API) client, and that require actions to be taken on behalf of the affected party. Examples of such changes include:

- Backwards incompatible API changes, including both the REST API and Plugin interfaces.
- Backwards incompatible changes to [Registry Application Rules](#).
- Behavioral changes that impact business processes.
- Changes where the affected party must make a decision as to how the functionality behaves going forward.

Examples of changes that are considered *drop in place*:

- User interface changes are generally considered drop in place, unless otherwise warranted.
- Changes that are automatically handled by `upgradeVersionShell`, even if such changes do not allow rolling back to an earlier version of the code without also restoring a database backup.