Versioning

Overview

In order to keep a product ecosystem healthy, reliable, and secure, a versioning system is required to indicate which version of the system is being used. There are several different systems to version programs and products which are to be released, with the aim of avoiding dependency hell.

Semantic Versioning - 2.0.0

The semantic versioning spec defines that a software package is given the version number:

MAJOR.MINOR.PATCH

Where the numbers are incremented when:

* MAJOR - incompatible API changes are made
* MINOR - functionality is added in a backwards compatible manner
* PATCH - backwards compatible bug fixes are added

To create a package with semantic versioning, first a precise and comprehensive public API must be identified. This can be by documentation in the readme, in the code itself, or any other means which is consistent and readily available along with the package.

Version numbers have the following restrictions:

* No leading zeros
* No negative numbers
* When a version number is incremented, any numbers beneath it are reset to 0
* The major version 0.minor.patch is used for development of the first release (public API not stable)
* Version 1.0.0 is the first stable public API
* Build metadata may be denoted by appending a plus sign then a series of dot separated ASCII alphanumeric identifiers