Mt Wilson

Versioning

# Background

Administrators and users of Mt Wilson need to be informed of the software version in order to facilitate their understanding of changes in the software.

# Architecture

There are separate versioning schemes for the software, the network APIs, and the overall solution.

Each software component must have a version number. Software components must use a semantic versioning scheme. In summary, this scheme uses a three-part version number comprised of major version, minor version, and patch number. Initial prototypes have a major version of zero. The first public release has a major version of one. After that, the major version is incremented when major backward-incompatible changes are made to the component. Initial prototypes have a minor version of one. After that, the minor version is incremented whenever a new feature is added. The minor version is reset to zero every time the major version is incremented. The patch number starts at zero. The patch number is incremented whenever a bug is fixed. The patch number is reset to zero whenever the minor version changes. When the patch number is zero, it does not need to be included in the version number.

A network API must have a single version number. The version number is incremented when backward-incompatible changes are made to the API. Any change which is backward compatible is by definition within the same API version number. The range of possible changes that are backward compatible depends on the specific protocols in use. An HTTP server is able to support multiple API versions concurrently by making each one available at a different URL. When this is done, URL path should include the version number prefixed with “v” like “v1”, “v2”, “v3”, etc.

The overall solution has a version number for marketing purposes. This version number is associated with a product guide that indicates the specific software component versions included and network API versions supported by the overall solution. The versioned set of software components included in the solution is tested more thoroughly in order to provide a great customer experience.

# Change Management

A guideline to use when making changes to the software is that adding new features is generally backward-compatible, but removing existing features is always backward-incompatible. In many cases it’s possible to make backward-compatible behavioral changes to existing features. Any change to an existing data structure or format should be regarded as backward-incompatible unless the data structure, format, or associated specified behavior specifically allows for the inclusion of new fields or methods. Renaming fields or methods is the same as adding a new method of the same name and deleting the original method - for this reason renaming things is always backward-incompatible.

Example: Mt Wilson network API v1 did not specify any behavior regarding addition of new fields to data structures. Consequently, both the client and server code were not tolerant of new fields because no work was done to support that. Adding new fields to Mt Wilson v1 is backward-incompatible. Mt Wilson network API v2 specifically states that clients and servers should ignore any fields they do not recognize in order to allow adding new fields while preserving backward-compatibility. Consequently, a lot of features can be added to Mt Wilson v2 without breaking compatibility for compliant clients and servers.

# Version Query

Software components that include a command-line interface should support a “version” command that displays the component’s version number.

Network APIs should include a “version” API that returns the server’s software component version number.

However, in both settings the version number alone may not be enough information to convey to the user the capabilities of the server. This is especially true for extensible software to which new features may be added after installation and especially when those features are not necessarily published by the original vendor. Therefore, both a command and network API to display the software version should also display the set of installed extensions and their version numbers.