

Note: This document has now moved to a PR on [GitHub](#)

Versioning Guidelines

To ensure a clear noun for referring to tools, APIs, models, data, and specification we will be using the word 'Product' as a substitute. A product is any entity that is built in Translator that will evolve across time and any changes to the entity will have direct or indirect consequences on its downstream application.

Given the rapid development of products across Translator, there is a high probability of encountering problems with compatibility as we build more complex systems on top of existing ones. To address this compounding problem one practical solution is to start versioning the products. This document provides a set of recommendations on how to go about versioning products in Translator.

This document contains 4 parts, each addressing the versioning policy for a particular product type:

- Tools & APIs
- Data Models
- Specification
- Datasets

When it comes to versioning, the default recommendation is to follow the [Semantic Versioning Guidelines](#).

Following is an excerpt from [Semantic Versioning](#) (SemVer) specification v2.0:

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards compatible manner, and
3. PATCH version when you make backwards compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

Ensuring that the products follow the above specification is the first step at communicating the evolution of a product in a structured manner. But there are product specific exceptions when the SemVer guidelines are not a perfect fit. Such scenarios and exceptions are detailed in the sections below.

Tools and APIs

Tools and APIs refer to software built by Translator teams to be used widely across the consortium.

The semantic versioning guidelines apply in its entirety to Tools and APIs.

In addition to the above, following are a few recommendations for releasing Tools and APIs

- Have periodic release cycles where new changes to the products are made available as stable release artifacts
- In the presence of period release cycles, discourage use of main (master) branch in production environments

Data Models

Data Models currently refer to just one cross-cutting data model across Translator - the Biolink Model.

While we follow semantic versioning guidelines, certain exceptions are permitted:

- It is entirely possible for a minor release to be backwards incompatible with previous versions.

Specifications

Specifications defined by committees and working groups that are used by tools and/or APIs that are built by Translator teams.

While we follow semantic versioning guidelines, certain exceptions are permitted:

- It is entirely possible for a minor release to be backwards incompatible with previous versions.

Data Products

Data Products are,

- Products generated from an ETL step by KPs/ARAs
- Ontologies that are developed by KPs/ARAs

- Graph serialization in KGX format generated by KPs/ARAs

While it is difficult to apply semantic versioning guidelines directly to data products, the following holds true:

- A major version release constitutes adding or removing data, or modifying the structure of the data as a result of a change in modeling decision
- A minor version release constitutes small changes in the data but the overall structure of the data remains the same
- No maintenance release for data products

General Best Practices

When building a new product that uses one or more products as its dependency:

- Always use a stable release of a product as a dependency
- Do not pin dependencies of a product to a fixed version. This will lead to dependency resolution errors where another product might be expecting the same dependency at a different version