

Publishing the Ontology Policies

1. Slash, not Hash

FIBO uses hash URIs not slash. The only exception is when referring to URIs from other ontologies that use #.

Further reading: [JIRA INFRA-39 Usage of '/' and '#' to indicate properties and classes](#)

2. FIBO will have a 24 hour rule befor acting on a GitHub PR

GitHub Pull requests will have a 24 hour rule for acting on after posted.

3. FBIO will have a single name for each concept as rdfs:label.

4. How to Deprecate

Deprecation in FIBO is typically used when a class or property is promoted from one ontology to another, typically to one that is "higher up" (more foundational) than where it started. There have been cases over the course of FIBO development when a property or class has been introduced by necessity "somewhere", and it turns out to be more generally needed. For example, an inverse property, `isCharacterizedBy`, was needed in BE, and was introduced originally in the `FunctionalEntities` ontology in BE, but the decision was made to move it up to FND to the `Relations` ontology where its inverse was defined. In this case, in order to retain backwards compatibility, the original property was deprecated and made equivalent to the new property in `Relations`.

The meaning of the original class or property should be retained to the degree possible, including all metadata about that property or class, but retention of inheritance, restrictions, and other axioms depends on how the class or property is defined when moved. We retain annotations so that we meet other FIBO hygiene policies and to assist users in understanding the elements in the ontology. As a matter of practice, we have retained inheritance if possible as well, or revised it to match the new concept or property that replaces it, so that the deprecated element shows up in the hierarchy close to, if not in the same general location as the original. Retention of restrictions depends on the changes to the class definition - if reasoning errors are not introduced then they can be retained, but redundancy should generally be limited, and we have been removing restrictions on the deprecated elements and making them equivalent to the new/replacement so that the semantics is preserved as desired. This approach limits maintenance of the redundant/deprecated content.

In summary:

1. For classes, for the deprecated class, retain inheritance and metadata and remove restrictions
2. Add an equivalent class axiom to the deprecated class to indicate that it is equivalent to the class that replaces it
3. For properties, for the deprecated property, retain inheritance and metadata and remove other axioms; add an equivalent property to point to the replacement

Deletion of any released first class ontology element (class, property, or non-generated individual) is not permitted; the released concept must be deprecated following the rules given above.

Deletion of provisional content is allowed if the content of the provisional ontology is provably redundant with something else in FIBO, uses a name that is not preferred (in which case the alternative name should be used in a synonym if appropriate in the concept that replaces it), or is considered to be wrong by subject matter experts. The latter case must be reviewed by the responsible FCT and the removal must be documented, in JIRA at a minimum, in case it turns out that the concept is needed at a later date. Typically non-redundant content is moved to an informative ontology and lingers until the FCT determines that it is truly not needed.