

# Common Type Library

- [CT scopes](#)
- [CT schema versioning and dependencies](#)
- [CT schema export support](#)

The Common Type Library (CTL) is a repository of reusable data type schemas that could be referenced and reused as the user creates specific schema definitions for Kaa modules.

This capability unifies management of all schemas in a Kaa server instance.

## CT scopes

Common Types (CT) can be defined within **scopes**: SYSTEM, TENANT, and APPLICATION. Scopes impact the visibility of CTs: for example, a CT defined with the application scope in App A is not visible for App B.

Each CT is identified by its unique Fully Qualified Name (FQN) and is subject to versioning. FQNs cannot conflict in any given scope. An attempt to create a new CT with a conflicting FQN will result in an error. Creating different CTs with matching FQNs within different applications of the same tenant is permitted, even though not advised. Prior to creating such a CT, a warning message will be displayed to the user in the admin UI.

The expected outcomes of an attempt to create a CT with a non-unique FQN are summarized in the following table.

CT scope	System	Tenant	Application
Matching FQN at System level	error unless the version is unique	error	error
Matching FQN at Tenant level	error	error unless the version is unique	error
Matching FQN in a different Tenant	N/A	OK	OK
Matching FQN at Application level	error	error	error unless the version is unique
Matching FQN in a different Application of the same Tenant	N/A	N/A	warning in web UI; OK at services level

Users are able to promote the CT's scope from APPLICATION to TENANT, provided that there is no other CT with the identical FQN within the given tenant, and that they have a permission to do so. Otherwise an appropriate error will be displayed.

## CT schema versioning and dependencies

The CT schema version must be explicitly defined in the type schema as shown below. An attempt to load a CT with no schema version will result in an error. Similarly, an attempt to load a schema with the already used version will result in an error. Deleting a schema version is only permitted if the FQN-version combination is not used in any schemas.

```
{
  "type" : "record",
  "name" : "SampleCT",
  "namespace" : "org.kaaproject.sample",
  "version" : 1,
  "dependencies" : [{ "fqcn": "org.kaaproject.sample.ReferencedCT", "version" : 2}]
  "fields" : [
    ...
  ]
}
```

The CTL user interface automatically suggests the next available version (max loaded + 1) for a given FQN when creating a new schema.

When referencing CTs, the user must specify both the FQN and the version. CTs may reference each other (see org.kaaproject.sample.ReferencedCT in the example above). Cyclic dependencies are not permitted. (Thus, CTs are nodes in a directed acyclic graph of dependencies.)

## CT schema export support

There are four ways of CT schema export:

1. **"shallow"** export simply retrieves the given schema as a file.
2. **"deep"** export retrieves the given schema as one file, and all of the referenced CTs as a separate file, recursively.

3. **"flat"** export will retrieve the given schema as one file with all referenced CTs inline.
4. **"library"** export will retrieve the given schema and all referenced CTs as compiled java classes compressed to a java archive file.