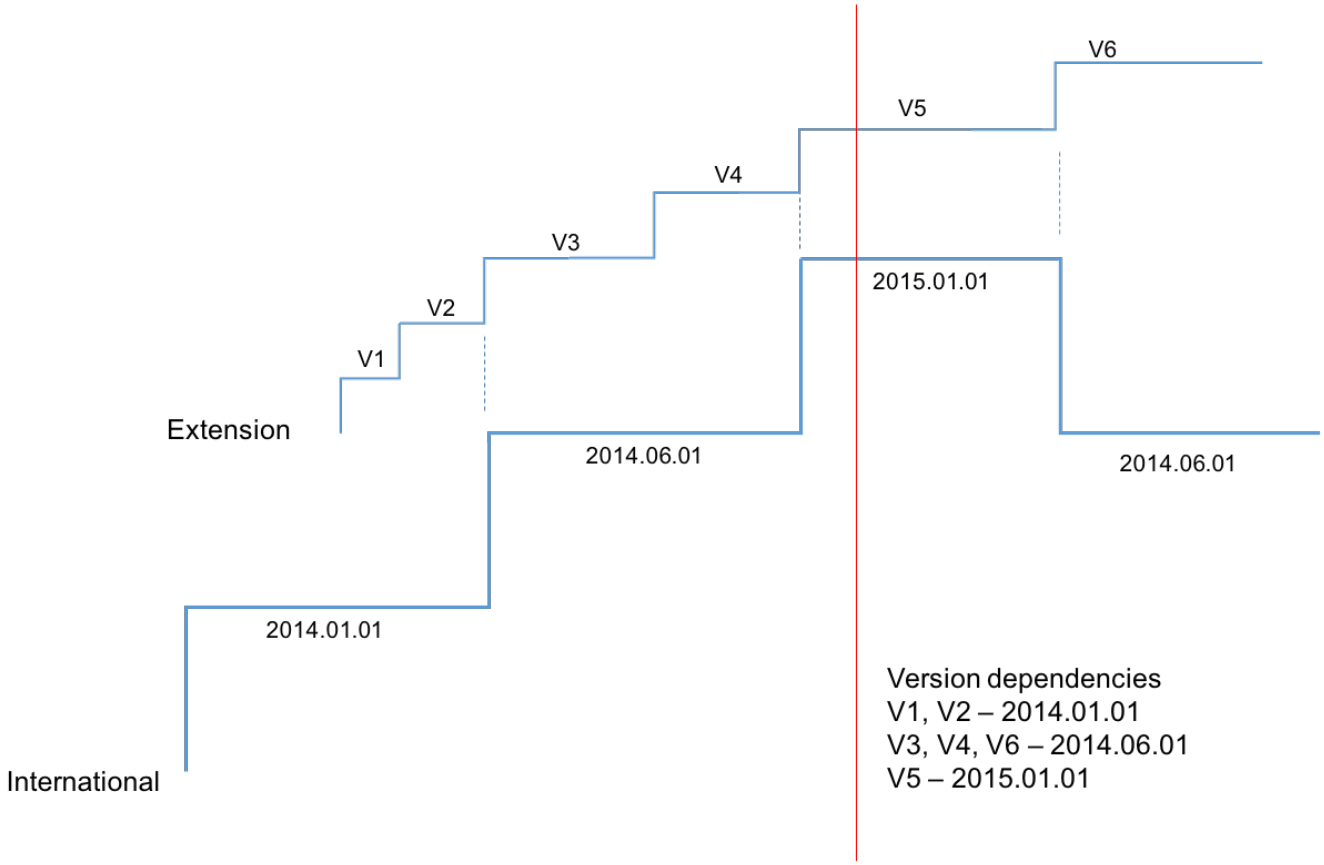
SNOMED CT Extension management

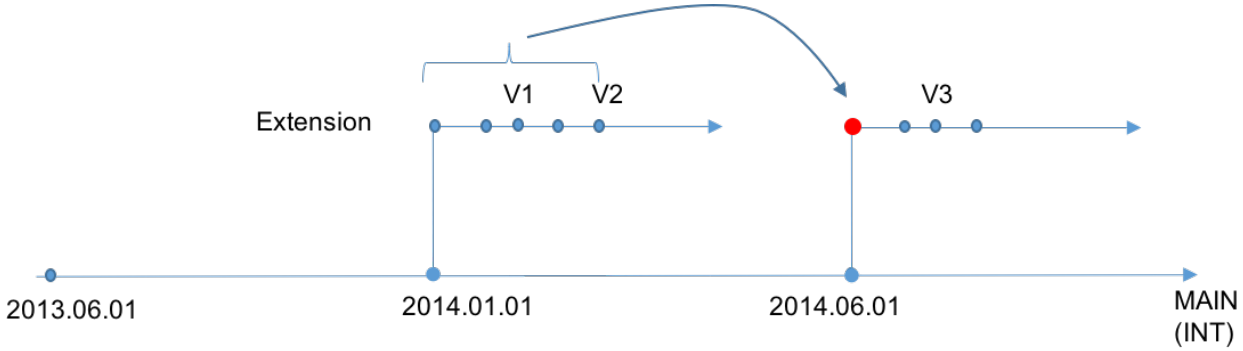
Introduction

Snow Owl is capable of managing multiple SNOMED CT extensions on a single server. This document highlights how each extension is managed in a dedicated area and lists the main authoring scenarios. Similarly to generic SNOMED CT authoring, content is developed via the help of a workflow where the work performed on a task is promoted to the repository after review. Promoting (donating) content from an extension to the international content or between extensions is not supported. Extension authors can follow their own release schedule and extension content can be versioned at any time during the authoring process. Extensions are based on a single international release version. The base international release can be upgraded 'underneath' the extension content. Extensions do not have an additional isolated area for long running development projects (project branch). The import and export of standard RF2 serialized content is supported. The international content is not edited on the server, only updated via delta releases as consecutive releases are available from IHTSDO's own authoring process. Search will be performed for both the native language and English as is currently supported. There will be no special search features to accommodate specific language requirements, such as word decompoundization.



Branching arrangement

The implemented branching arrangement closely follows the current branching preferences where the international release is imported onto the MAIN branch and each Member Country will have a dedicated branch for the extension content. The only commits on the MAIN branch are the contents of the delta release for a particular version. These commit points are versioned during the import process. The dedicated branch will be forked off the MAIN branch at the point representing the base international release as shown in Figure 1.



Versioning

The extension content is marked with a version tag for future reference with a new empty branch. All new components will receive an effective time.

Upgrading/Downgrading

Upgrading means to increase the version of the base international release. During the upgrade process, the current extension branch is forward-rebased via creating a new branch and copying over the entire content as a single commit. As shown in Figure 2, the V1 and V2 versioned content is based on the 2016.01.01, whereas the V3 version is based on the 2016.06.01 version of the international release. The main advantage is that the old branch still maintains the earlier versions of the extension, this way the version dependencies between the extension and international release can be implicitly represented via the branches. If the user decides to switch back to the read-only older V1 version, content is properly displayed with international content from 2016.01.01. To avoid content with incorrect dependencies, only versioned extensions should be allowed to be upgraded. Downgrading means to decrease the version of the base international release. The same process described above for upgrade applies.

International version update

When new international release is available, the corresponding delta is imported onto the MAIN branch.

Data provisioning example for a single extension

The following example demonstrates how a blank server is provisioned for extension management. The example is carried out using both the command line OSGi interface as well as the REST API.

OSGi command line interface

First, the full International Release is imported with version tags created (*-v*). The selected Language Refset is English.

```
> sctimport rf2_release -t full -v /path/to/SnomedCT_RF2Release_INT_20160131.zip
/path/to/snomed_ct_international.json
```

Where the `snomed_ct_international.json` release descriptor file is:

```
{
  "name": "Systematized Nomenclature of Medicine Clinical Terms International Version
",
  "shortName": "SNOMEDCT",
  "language": "ENG",
  "codeSystemOID": "2.16.840.1.113883.6.96",
  "maintainingOrganizationLink": "http://www.ihtsdo.org",
  "citation": "SNOMED CT contributes to the improvement of patient care by
underpinning the development of Electronic Health Records that record clinical
information in ways that enable meaning-based retrieval. This provides effective
access to information required for decision support and consistent reporting and
analysis. Patients benefit from the use of SNOMED CT because it improves the recording
of EHR information and facilitates better communication, leading to improvements in
the quality of care."
}
```

After the import process, to check the terminology registry issue the command:

```
> terminologyregistry listall
```

To see all the version branches created execute:

```
> snowowl listbranches snomedStore
```

To import an extension terminology based on the *2016-01-31* International release, issue the command:

```
> sctimport rf2_release -t full -b 2016-01-31 -v
/path/to/SnomedCT_Extension_Release.zip /path/to/snomed_ct_extension.json
```

Where the `snomed_ct_extension.json` release descriptor file is:

```
{
  "name": "B2i Healthcare SNOMED CT extension",
  "shortName": "SNOMEDCT_B2i",
  "language": "ENG",
  "codeSystemOID": "2.16.840.1.113883.6.961",
  "extensionOf": "SNOMEDCT",
  "maintainingOrganizationLink": "http://www.b2i.sg",
  "citation": "This is an example SNOMED CT extension by B2i Healthcare."
}
```

Again, after the import process, to check the terminology registry issue the command:

```
> terminologyregistry listall
```

To see all the version branches created execute:

```
> snowowl listbranches snomedStore
```

There should be a branch named *MAIN/2016-01-31/SNOMEDCT_B2i* dedicated for the initial content of the extension.

REST API

Importing the terminology and creating an entry in the terminology registry require two separate REST endpoints to be invoked.

Upload the import configuration:

```
POST /snowowl/snomed-ct/v2/imports
{
  "createVersions": true,
  "type": "FULL",
  "branchPath": "MAIN",
  "codeSystemShortName": ""
}
```

Note: For backward compatibility, missing or empty *codeSystemShortName* result in a SNOMED CT International Release entry in the terminology registry, no explicit operation is required.

Import the International archive:

```
POST /snowowl/snomed-ct/v2/imports/{importId}/archive
```

After the import process, check the terminology registry for the entry representing the

International release:

```
GET /snowowl/admin/codesystems
```

Create new branch for extension:

```
POST /snowowl/snomed-ct/v2/branches
{
  "metadata": {},
  "parent": "MAIN/2016-01-31",
  "name": "Extension_branch"
}
```

To check the branch created (*MAIN/2016-01-31/Extension_branch*):

```
GET /snowowl/snomed-ct/v2/branches
```

Create new Code System for the extension. The extension will initially be based on the *2016-01-31* International release:

```
POST /snowowl/admin/codesystems
{
    "oid": "2.16.840.1.113883.6.961",
    "name": "B2i Healthcare SNOMED CT extension",
    "shortName": "SNOMEDCT_B2i",
    "organizationLink": "http://www.b2i.sg",
    "primaryLanguage": "ENG",
    "citation": "This is an example SNOMED CT extension by B2i Healthcare.",
    "iconPath": "icons/b2i_extension_icon.png",
    "terminologyId": "com.b2international.snowowl.terminology.snomed",
    "repositoryUuid": "snomedStore",
    "branchPath": "MAIN/2016-01-31/Extension_branch",
    "extensionOf": "SNOMEDCT"
}
```

Check the terminology registry for the extension entry:

```
GET /snowowl/admin/codesystems
```

Upload import configuration for extension import:

```
POST /snowowl/snomed-ct/v2/imports
{
  "createVersions": true,
  "type": "DELTA",
  "branchPath": "MAIN/2016-01-31/Extension_branch",
  "codeSystemShortName": "SNOMEDCT_B2i"
}
```

Import the extension:

```
POST /snowowl/snomed-ct/v2/imports/{importId}/archive
```

Versioning extension content

There is no command-line command available for versioning, the functionality is only accessible via the REST interface:

```
POST /snowowl/admin/codesystems/SNOMEDCT_B2i/versions
{
  "version": "Extension:2016-06-01",
  "description": "Extension test version",
  "effectiveDate": "20160601"
}
```

Upgrading extension content

There is no command-line command available for upgrading, the functionality is only accessible via the REST interface.

Import the new International DELTA archive to the MAIN branch in two steps:

First create the import configuration.

```
POST /snowowl/snomed-ct/v2/imports
{
  "createVersions": true,
  "type": "DELTA",
  "branchPath": "MAIN",
  "codeSystemShortName": "SNOMEDCT"
}
```

Then upload the archive to start the import.

```
POST /snowowl/snomed-ct/v2/imports/{importId}/archive
```

After the import process, create a new branch for extension:

```
POST /snowowl/snomed-ct/v2/branches
{
  "metadata": {},
  "parent": "MAIN/2016-07-31",
  "name": "Extension_branch"
}
```

Then copy over the entire content onto the target extension branch. If the merge fails because of conflicts, fix the errors and then try again.

```
POST /snowowl/snomed-ct/v2/merges
{
  "source": "MAIN/2016-01-31/Extension_branch",
  "target": "MAIN/2016-07-31/Extension_branch",
  "commitComment": "Upgrade extension content."
}
```

The above call will return the unique ID of the merge process which can be used to poll its status (e.g. *IN_PROGRESS*, *COMPLETED*, *CONFLICTS*, etc).

```
GET /snowowl/snomed-ct/v2/merges/{mergeId}
```

Finally update the branch path of the extension Code System to point to the target extension branch.

```
PUT /snowowl/admin/codesystems/SNOMEDCT_B2i
{
  "repositoryUuid": "snomedStore",
  "branchPath": "MAIN/2016-07-31/Extension_branch"
}
```

Downgrading extension content

There is no command line command available for downgrading, the functionality is only accessible via the REST interface.

Copy over the current extension content onto the target extension branch. If the merge fails because of conflicts, fix the errors and then try again.

If no new content was added to the current branch since the merge, you can skip this step.

```
POST /snowowl/snomed-ct/v2/merges
{
  "source": "MAIN/2016-07-31/Extension_branch",
  "target": "MAIN/2016-01-31/Extension_branch",
  "commitComment": "Downgrade extension content."
}
```

Then update the branch path of the extension Code System to point to the target extension branch.

```
PUT /snowowl/admin/codesystems/SNOMEDCT_B2i
{
  "repositoryUuid": "snomedStore",
  "branchPath": "MAIN/2016-01-31/Extension_branch"
}
```

SNOMED CT Swedish Extension management example

The following example demonstrates how a blank server is provisioned for the Swedish extension management through the REST API.

First create the import configuration for the International release.

```
POST /snowowl/snomed-ct/v2/imports
{
  "createVersions": true,
  "type": "FULL",
  "branchPath": "MAIN",
  "codeSystemShortName": "SNOMEDCT"
}
```

Then upload the RF2 archive file to start the import.

```
POST /snowowl/snomed-ct/v2/imports/{importId}/archive
```

After the import process, create a new branch for the Swedish extension which is based on version 2015-01-31.

```
POST /snowowl/snomed-ct/v2/branches
{
  "metadata": {},
  "parent": "MAIN/2015-01-31",
  "name": "SNOMEDCT-SE"
}
```

Create a new Code System for the Swedish extension. The branch path points to the previously created branch.

```
POST /snowowl/admin/codesystems
{
    "oid": "2.16.840.1.113883.6.962",
    "name": "SNOMED CT Swedish Extension",
    "shortName": "SNOMEDCT-SE",
    "organizationLink": "http://www.socialstyrelsen.se/",
    "primaryLanguage": "SWE",
    "citation": "SNOMED CT contributes to the improvement of patient care by
underpinning the development of Electronic Health Records that record clinical
information in ways that enable meaning-based retrieval. This provides effective
access to information required for decision support and consistent reporting and
analysis. Patients benefit from the use of SNOMED CT because it improves the recording
of EHR information and facilitates better communication, leading to improvements in
the quality of care.",
    "iconPath": "icons/snomed.png",
    "terminologyId": "com.b2international.snowowl.terminology.snomed",
    "repositoryUuid": "snomedStore",
    "branchPath": "MAIN/2015-01-31/SNOMEDCT-SE",
    "extensionOf": "SNOMEDCT"
}
```

Create the import configuration for the Swedish import.

```
POST /snowowl/snomed-ct/v2/imports
{
  "createVersions": true,
  "type": "FULL",
  "branchPath": "MAIN/2015-01-31/SNOMEDCT-SE",
  "codeSystemShortName": "SNOMEDCT-SE"
}
```

Upload the Swedish RF2 archive file to start the extension import.

```
POST /snowowl/snomed-ct/v2/imports/{importId}/archive
```

Create a new branch based on version 2016-01-31. The Swedish extension will be merged onto this

branch.

```
POST /snowowl/snomed-ct/v2/branches
{
  "metadata": {},
  "parent": "MAIN/2016-01-31",
  "name": "SNOMEDCT-SE"
}
```

Start a merge to upgrade the extension content onto the previously created branch.

```
POST /snowowl/snomed-ct/v2/merges
{
  "source": "MAIN/2015-01-31/SNOMEDCT-SE",
  "target": "MAIN/2016-01-31/SNOMEDCT-SE",
  "commitComment": "Upgrade Swedish extension to international version 2016-01-31"
}
```

The POST operation will return a *mergeId* that can be used to monitor the progress of the merge via:

```
GET /snowowl/snomed-ct/v2/merges/{mergeId}
```

Which operation will return with failed status and the following conflicts:

```
Relationship with ID '6271000052128' has a conflict of type 'HAS_INACTIVE_REFERENCE'
on target branch, conflicting attributes are: [destinationId -> 373245004].
Relationship with ID '16741000052126' has a conflict of type 'HAS_INACTIVE_REFERENCE'
on target branch, conflicting attributes are: [destinationId -> 373245004].
Relationship with ID '44001000052128' has a conflict of type 'HAS_INACTIVE_REFERENCE'
on target branch, conflicting attributes are: [destinationId -> 237968007].
Relationship with ID '1929491000052120' has a conflict of type
'HAS_INACTIVE_REFERENCE' on target branch, conflicting attributes are: [destinationId
-> 237968007].
```

After fixing the conflicts, start another merge (same as before) and then update the branch path of the Swedish Code System.

```
PUT /snowowl/admin/codesystems/SNOMEDCT-SE
{
  "repositoryUuid": "snomedStore",
  "branchPath": "MAIN/2016-01-31/SNOMEDCT-SE"
}
```

Developer notes

During the upgrade/downgrade process, conflicts are identified between the terminology to be upgraded/downgraded and the international release. Using the **mergeConflictRuleProvider** extension point, rule providers can be registered the provide conflict rules. The currently registered rule provider is **com.b2international.snowowl.datastore.server.snomed.merge.SnomedMergeConflictRuleProvider.java**.