

# Asserted Value Set Implementation Assessment

## Contents of this Page

- [Situation](#)
- [Background](#)
- [Assessment](#)
- [Recommendation](#)

## Situation

Currently NCI terminology administrative staff load ~900 value set definitions and the same number of resolved value sets into the LexEVS common terminology service. These value sets are defined in a series of xml files which can be auto generated, but sometimes must be checked individually to insure accuracy or correct errors. Loading time of 18 to 24 hours from the files into LexEVS definition and coding scheme entires is extensive enough to be subject to network and other system performance problems. Even minor updates tend to require that the entire process begin over again. Overall the process is time consuming, painstaking, and at risk to minor hardware issues.

## Background

Originally values sets for LexEVS were expressed in terms similar to the ISO 11179 specification. The specification for Value domains were conceived as a set of rules that could be applied to any version of a given terminology with the expectation that a current version of the value set would be expressed. Value domains, expressed as value set definitions in LexGrid, were expressed as entries in an xml file. Value set resolution was enhanced by adding and implementing a model of a static resolution of a value set into a coding scheme. Value set expression for the NCI Thesaurus (NCIt) has scaled up over the last few years to nearly thousand definitions with their accompanying file sets.

## Assessment

The LexEVS implementation model of loading Value Set Definitions and their Resolved Value Set Coding Scheme counterparts has become unwieldy as the number of value sets has grown. As the value set definition file set grew into the hundreds it became more difficult to keep up to date and more error prone to maintain and load. Value sets asserted by the NCIt terminology source (Source Asserted Value Sets), provide a complete definition of value sets in static terms. It does not require resolving a graph, or a union of a variety of sources to achieve its definition. At the same time the root (or target) entity of the asserted values carries adequate metadata to define what former resolved value set coding scheme represented without the workflow complexity of having to create separate coding scheme database entries. This should result in higher performing value set retrieval and eliminate very lengthy loading times. At the same time this would change the way that value sets are made available to end users meaning query mechanisms and Java and REST api results would retrieve far fewer value sets than before. Implementing an API for a service for this functionality does imply additional APIs and potential changes to the way value set users access value sets. While many resolved value sets would cease to be respresented as stored coding schemes, they would still be accessible as resolved value set coding scheme objects available through the resolved value set API, which would not change in the way it returned value sets. If there were users who took advantage of accessing value sets as if they were coding schemes, they would have to do so using the resolved value set api.

## Recommendation

Load value set definitions from source defined (source asserted) value sets while continuing to allow traditional value set definition xml files loads. Create a source asserted value set API and search mechanism. Integrate traditional resolved value set coding schemes in the search mechanism. Update current Resolved Value Set and CTS2 functions to allow source asserted values to be retrieved along with traditional value sets. Provide remote api support for asserted value set API functions where necessary. Support users who may have been using value sets as coding schemes rather than value sets and resolved value sets.