

Ulrich Huber

SLR Toolkit

Informal Documentation

Task: Deduction of Initial Taxonomy

Outline

1. Task
2. Usage
3. Development Guide
4. Possible Improvements

Task

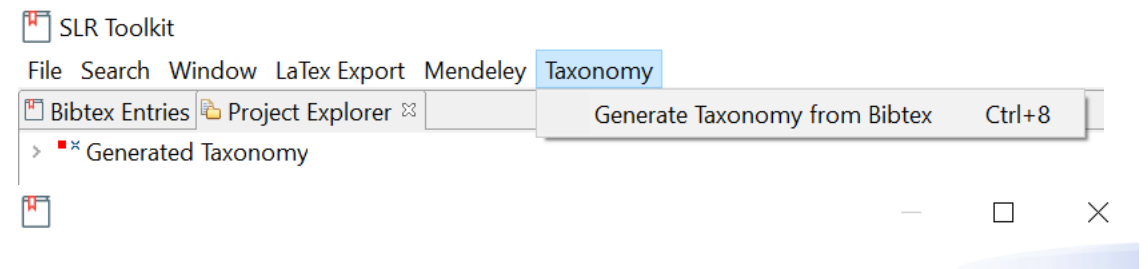
Task

- Create ability to generate a taxonomy based on bibtex metadata such as
 - Publication Type/Document Type – Including more detailed Scopus types
 - Venue – Also resolve cross-references
 - => Implemented in new plugin `de.tudresden.slr.classification`
- Include a New Project Wizard incorporating taxonomy generation
 - => Added to `de.tudresden.slr.wizards`

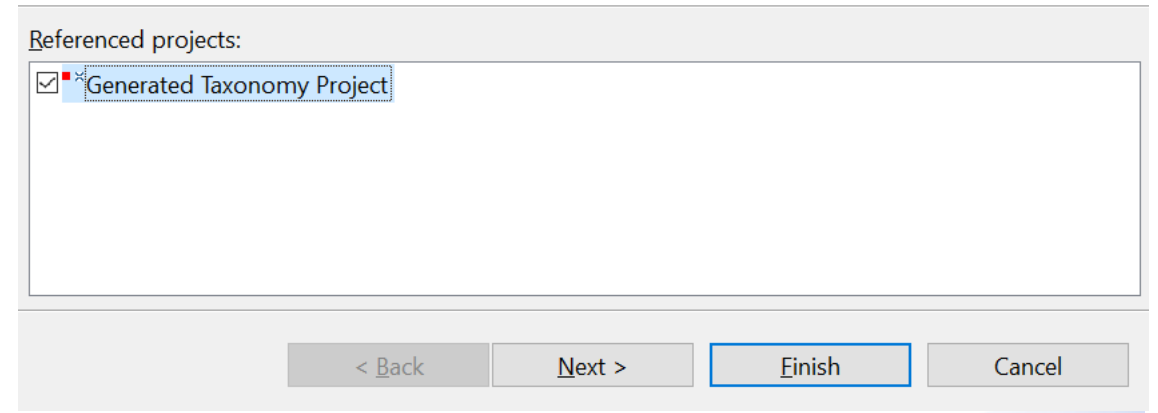
Usage

Usage

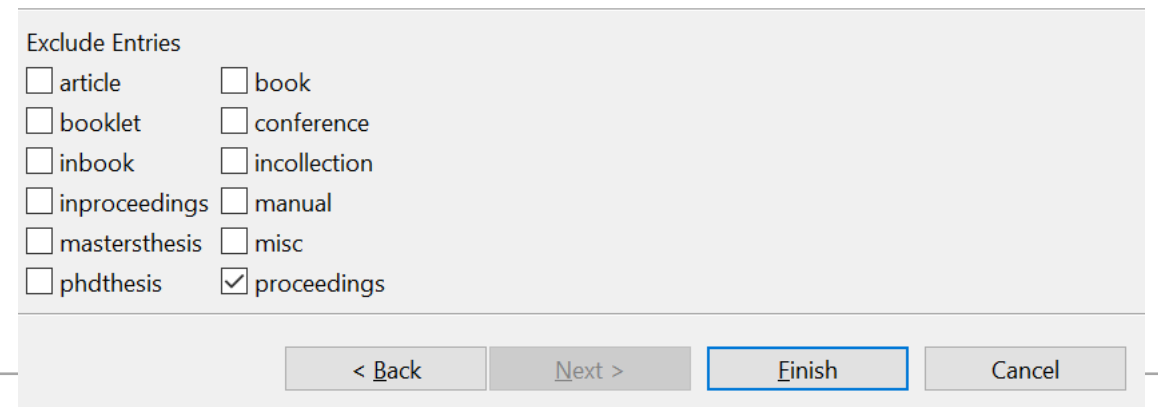
- From Menu
 - From tab „Taxonomy“ select „Generate Taxonomy from Bibtex“
 - Select projects to process
 - Select types to exclude



Generate Taxonomy from Bibtex

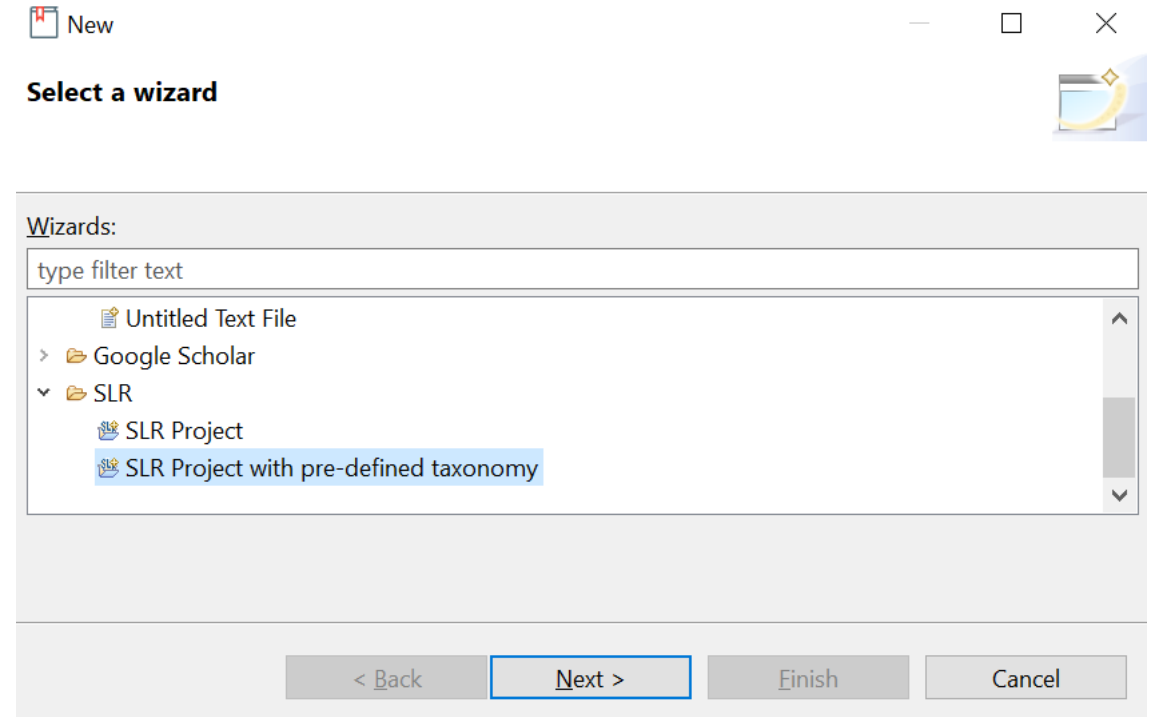


Exclude Bibtex Entry Types



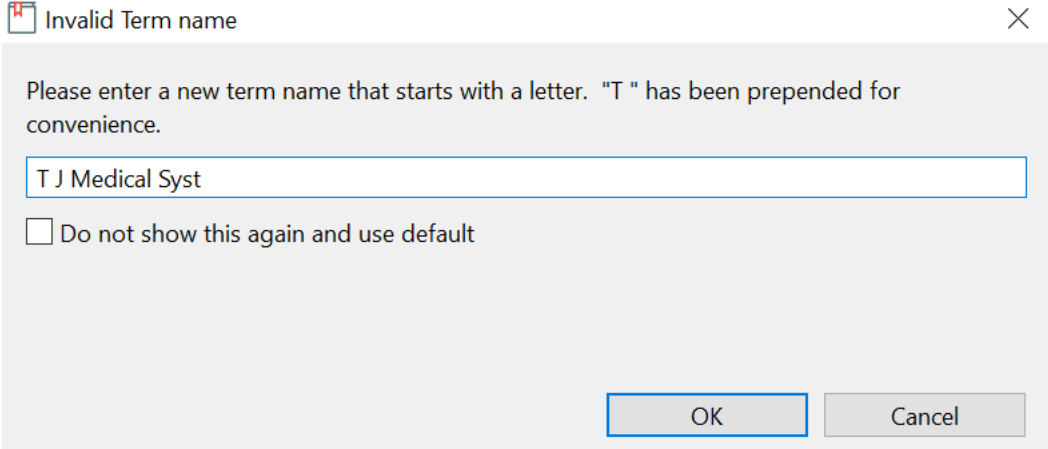
Usage

- From New Project Wizard
 - In the new projects menu select „New SLR-Project with pre-defined taxonomy“
 - Select new project options as usual
 - Select types to exclude



Usage

- Resolve malformed term names (invalid characters, first character is not alphanumeric, etc.)
- Option to use default to avoid the popup: Use check box



Invalid Term name

Please enter a new term name that starts with a letter. "T " has been prepended for convenience.

T J Medical Syst

☐ Do not show this again and use default

OK Cancel

Development Guide

Development Guide – Generate Taxonomies with StandardTaxonomyClassifier

- Instantiation
 - Requires a „term name validator“ to check for invalid term names
 - Requires a „malformed term name handler“ to replace invalid term names if necessary
 - Both can be instantiated automatically with the default constructor or supplied via dependency injection
 - Default Constructor instantiates a MalformedTermNameHandlerImpl that does not pop up the malformed term name dialog
 - To enable this behaviour, instantiate using StandardTaxonomyClassifier(false)

Development Guide – Generate Taxonomies with StandardTaxonomyClassifier

- Taxonomy Generation
 - Generate taxonomies with
 - createStandardTaxonomy(Document doc) for single documents with no cross reference
 - createStandardTaxonomy(Document doc, Document crossRefDoc) for documents with a cross reference
 - classifyDocumentsInProject(IProject project, List<String> excludeList) for entire projects
 - excludeList can be omitted if no type should be excluded
 - While using wizards and user input, exclusion lists can be generated via EntryExcludeSelectionPage.getExclusionList()

Development Guide – Check for invalid term names

ITermNameValidator

- Functional Interface returning a boolean whether the given String is a valid term name
- Implemented in TermNameValidatorImpl
 - Also implements Eclipse IInputValidator to be used in eclipse text fields
 - Checks if a given name starts with a letter and only contains alphanumerical characters

Development Guide – How to fix invalid input

IMalformedTermNameHandler

- Functional Interface returning a valid term name variant of a supplied string
- Requires a term name validator
- Implemented in MalformedTermNameHandlerImpl
 - Strips supplied string of non-alphanumeric characters
 - Prepends a standard prefix
 - If useDefault is set to false, opens a dialog with a pre-formatted term name string according to the rules above that can be changed via user input (See Usage slides)

Problems and Possible Improvements

Problems and possible improvements

- There are several problematic areas that can be improved
 - Consider Scopus Types in exclude list, maybe precompute all occurring types and exclude based on that
 - Improving performance by saving document changes asynchronously, multi-threading in general
- Some Nice-To-Haves
 - A progress bar
 - Proper Command Undo functionality
 - Different ways to deal with invalid term names (custom prefix/handling options)