

---

# Visual vocabulary suite

A vocabulary editor and content tagging extension

**This document** provides both a general system guide and a comprehensive technical description of two software tools designed to allow the development of controlled vocabularies and their use in tagging content.

The document is aimed at several audiences. The general overview guide is suitable for management and other general staff who wish to assess the potential for creating and managing controlled vocabularies and for using vocabulary-based categorisation in content-focused websites. The more technical sections are aimed at taxonomists, information architects and developers who want to understand the implications of using the tools within their environments.

---

## Background to the development of the suite

These tools were developed to address two needs:

1. A general purpose, usable, web-based system allowing the development of skos-based controlled vocabularies
2. An extension tool allowing editors of the content management system to easily tag content against a collection of vocabularies

---

## Vocabulary editor, system description

The vocabulary editor is a web-based editor that allows the user to construct a controlled vocabulary as a visually-attractive hierarchical tree structure. It also provides a repository with sophisticated tools for management of the vocabularies, including the capability to import existing structures from Excel (.xls) or SKOS (.xml) files and to export SKOS. A quick search is also provided, allowing quick identification of concepts from vocabularies and allowing the user to locate the concepts visually within the vocabulary hierarchy.

### Features in more depth

The visual editor is accessed by a suitable URL: for example [unilexicon.com](http://unilexicon.com)  
This leads to the home page for the editor (see Figure 1).

## Visual Vocabulary Editor

This is a visual tool for managing controlled vocabularies.

- it allows users to explore and edit vocabularies through an attractive user interface
- it helps users to quickly comprehend relations between concepts in large complex classification systems
- it makes editing SKOS files fast and productive
- simplifies the process of writing contextual queries for navigation vocabularies
- imports and exports vocabularies using a variety of sources and formats including SKOS, Excel and Drupal

<a href="#">Access control</a> (25)	<a href="#">DoCMigration priority</a> (9)	<a href="#">FAQs</a> (19)	<a href="#">Navigation</a> (565)	<a href="#">School phase</a> (28)
<a href="#">Assessment Focus</a> (69)	<a href="#">Dynamic Strand</a> (221)	<a href="#">Glossary</a> (1557)	<a href="#">Ofsted framework</a> (36)	<a href="#">School Year</a> (18)
<a href="#">Behaviour, attendance and SEAL</a> (233)	<a href="#">Early Years</a> (349)	<a href="#">Help Topic</a> (7)	<a href="#">Ofsted report navigation</a> (33)	<a href="#">Secondary mathematics Framework exemplification</a> (93)
<a href="#">Content clump</a> (128)	<a href="#">Educational level</a> (20)	<a href="#">Initiative</a> (148)	<a href="#">Primary intervention resource finder</a> (20)	<a href="#">Street</a> (12)
<a href="#">Content type</a> (94)	<a href="#">Educational topic</a> (256)	<a href="#">Integrated Public Sector Vocabulary</a> (3080)	<a href="#">Product status</a> (5)	<a href="#">Subject Specifiers</a> (3773)
<a href="#">Content workflow state</a> (4)	<a href="#">English region (GOR)</a> (10)	<a href="#">Keystage</a> (6)	<a href="#">Product type</a> (3)	<a href="#">Temporary Education Topic Vocabulary</a> (236)
<a href="#">Creator</a> (15)	<a href="#">EYFS Objectives - Age Group</a> (6)	<a href="#">Language</a> (32)	<a href="#">Pupil Writing Targets</a> (35)	<a href="#">Unit plan block</a> (16)
<a href="#">Cross Curricular Skills</a> (105)	<a href="#">EYFS Objectives - Aspect</a> (34)	<a href="#">Local authority</a> (150)	<a href="#">Resource type</a> (63)	
<a href="#">Curriculum subject</a> (37)	<a href="#">EYFS Objectives - Focus</a> (4)	<a href="#">Location radius</a> (6)	<a href="#">Role</a> (88)	

Vocabularies [add another](#)

[vocabularies](#) [login](#)

How can we make this website better?

Send us your suggestions, love letters. Tell us about issues...

Figure 1. The visual vocabulary editor home page

This shows a list of all of the vocabularies currently loaded into the repository. It also allows the user to create or upload another vocabulary or to use the search box to search concepts.

## Viewing and editing vocabularies

To view or edit a vocabulary, the user clicks on a vocabulary name. This will give a new display showing that vocabulary with a subset of its hierarchical structure. For example, here is the Educational topic vocabulary:

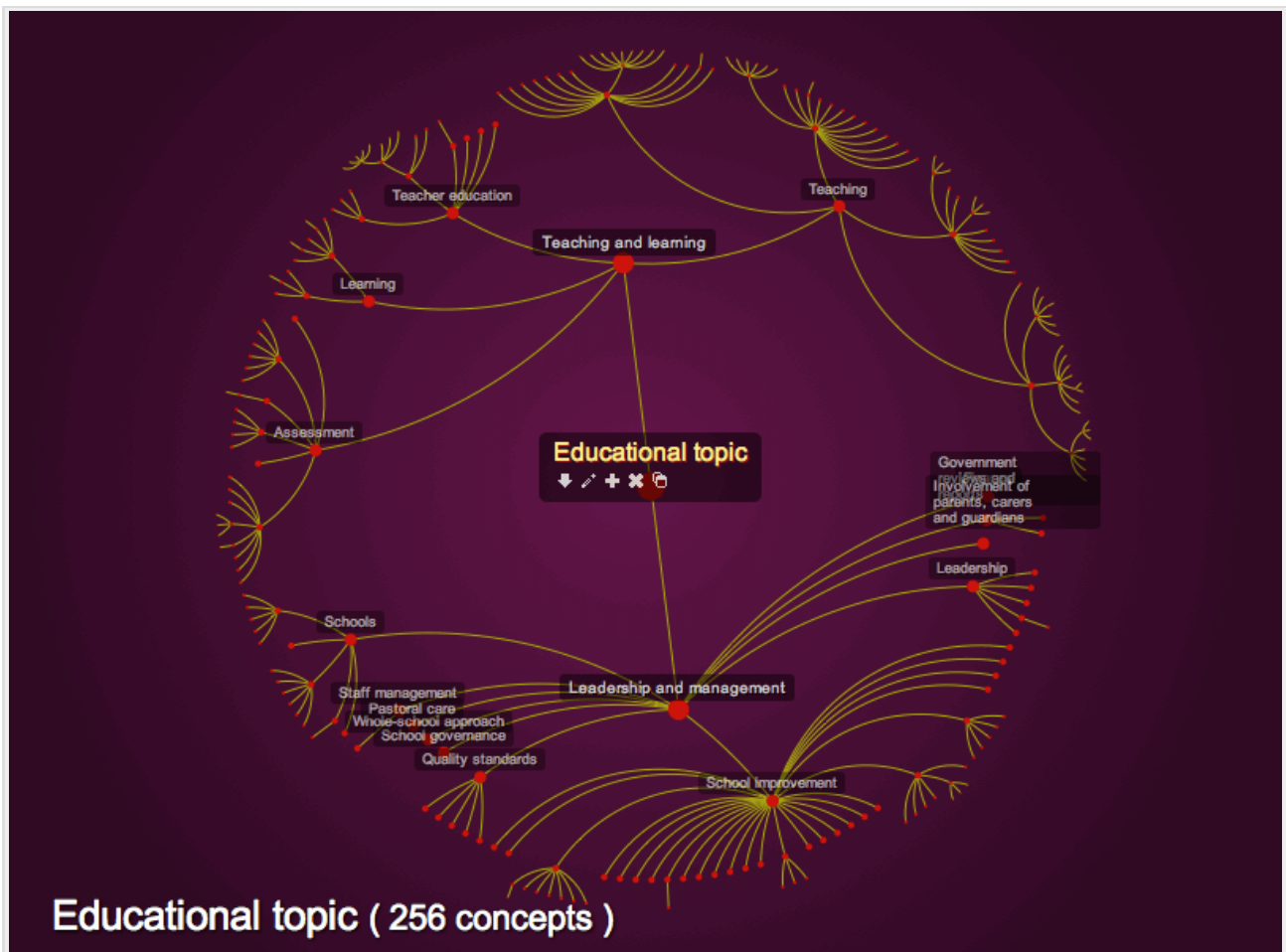


Figure 2. The visual vocabulary view of the Educational topic vocabulary

The Vocabulary title is at the centre of the view, with a set of branches and nodes representing the child and grand-child concepts. In the interests of clarity only a selection of descendant concepts are shown, and only the first two levels of these are labelled. Selecting a node will cause that node to move to the centre and show its child concepts:



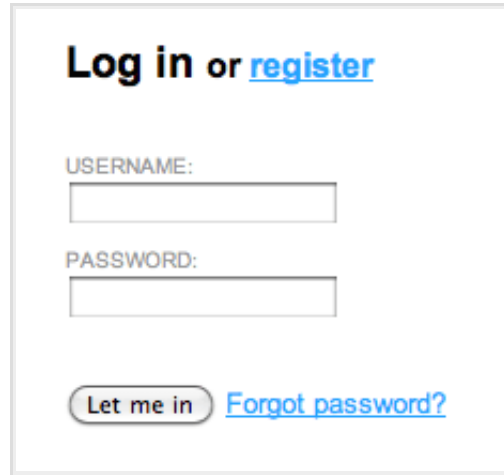
Figure 3. Selecting a child term from the Educational topic vocabulary

In Figure 3, the user has selected Teaching and learning. This concept is now centered in the display and its child, grand-child (and so on) concepts are shown. The route back to its parent is also

shown, together with the other child concepts of the parent. This overall tree structure is called a **hypertree**. Note also that the hierarchical structure is shown in the panel to the right. The user can choose concepts from here to focus upon as an alternative to the tree structure.

It is worth noting that each vocabulary and each node has its own permalink. So it is possible for a user to bookmark, say, Educational topic > Teaching and learning > Teaching as <https://unilexicon.com/vocabularies/teaching/#node10584>

The user can also log in to make changes to the vocabulary.



**Log in or [register](#)**

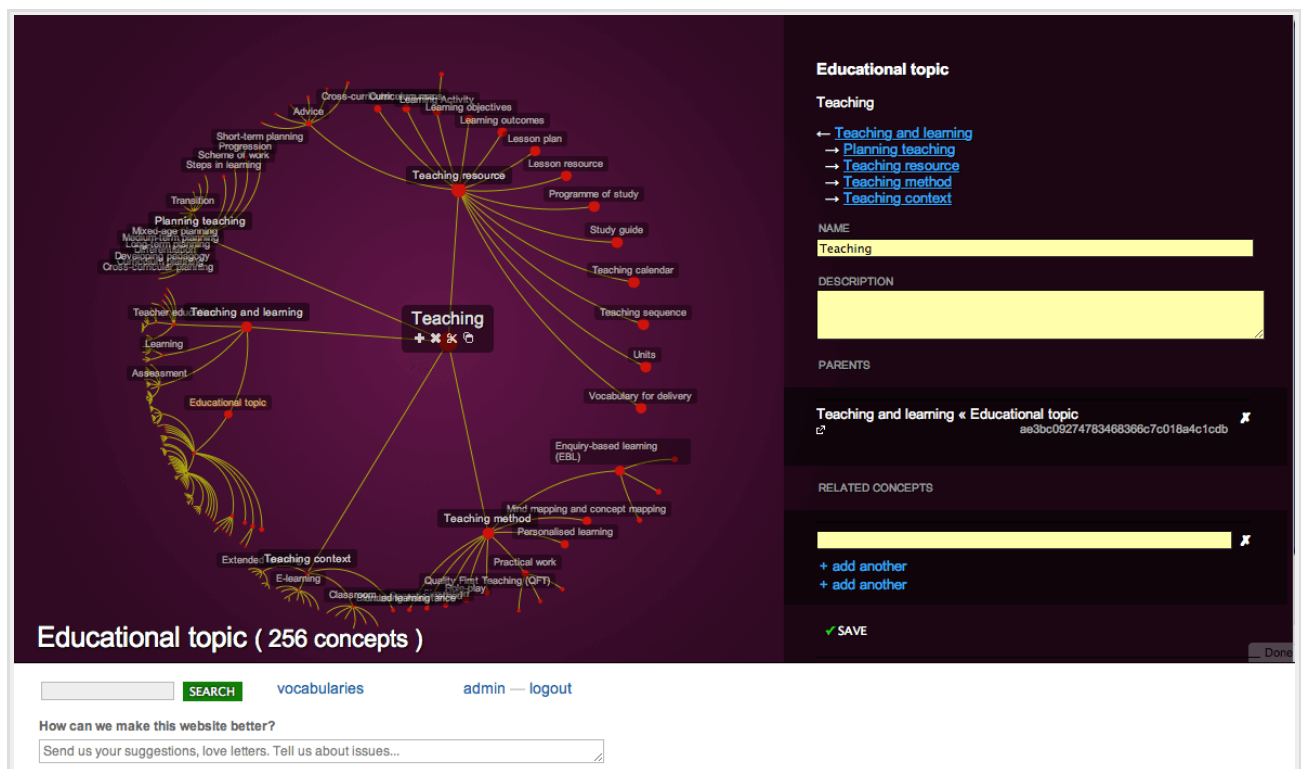
USERNAME:

PASSWORD:

[Forgot password?](#)

Figure 4. The login / register screen

Once logged in, the view changes to show additional information about the current concept. Figure 5 shows the Teaching concept (Educational topic > Teaching and learning > Teaching):



**Educational topic ( 256 concepts )**

**Educational topic**

**Teaching**

- ← Teaching and learning
- Planning teaching
- Teaching resource
- Teaching method
- Teaching context

NAME  
Teaching

DESCRIPTION

PARENTS

Teaching and learning ← Educational topic  
aa3bc09274783468366c7c018a4c1cdb

RELATED CONCEPTS

+ add another  
+ add another

✓ SAVE

SEARCH vocabularies admin — logout

How can we make this website better?  
Send us your suggestions, love letters. Tell us about issues...

Figure 5. Concept editing display once logged in

This shows a set of additional fields allowing the authenticated user to add a description and to add relationships (e.g. RT, UF and USE relationships) with terms elsewhere in this vocabulary and in other vocabularies.

Editor also implements [assistance with navigational queries](#) for navigation specific vocabularies.

## Content tagging plugin, system description

Having a repository of well-designed controlled vocabularies is important, but equally important is providing a content management system with an effective mechanism for making use of those vocabularies.

The web-based tagging editor is designed as a small efficient extension to Chrome web browser. The sequence of screenshots below outlines the process (*note: this is in active development and the appearance will change substantially, but this is a reasonable guide to the way the tool will work*).

The screenshot displays a web-based tagging editor interface. It features a 'TITLE' field containing 'AFs 3 and 5: Project on diversity and adaptation'. Below the title is an 'EXCERPT' field with a text area containing a paragraph about 'Assessing Pupils' Progress (APP) standards'. The bottom section is titled 'TAGS' and lists several tags with their corresponding IDs and a blue 'X' icon for each. The tags are: '1b « DoCMigration priority' (ID: 3da95520e8a62a9ac69b6a3973403bdb), 'Adaptation « Environment, energy and nutrient transfer « Science « Subject Specifiers' (ID: Sc-0100), 'Article « Content type' (ID: c587bbc8739dbf2f6e46eea38414ad88), 'Assessing Pupils' Progress (APP) « Personalisation and Progression « Initiative' (ID: 372bb0cff6454733d7ec7f0e8f8655ee), 'Assessment Focuses (AFs) « Assessing Pupils' Progress (APP) « Assessment « Teaching and learning « Educational topic' (ID: f78b8a255e2c429505b9d928e1e682bb), and 'Biology « Science « Subject Specifiers' (ID: 4b3a2582c1db93a6a9d1fba8a412c949). The bottom-most tag, 'Education and skills « Integrated Public Sector Vocabulary', is partially visible.

Tag	ID	Action
1b « DoCMigration priority	3da95520e8a62a9ac69b6a3973403bdb	X
Adaptation « Environment, energy and nutrient transfer « Science « Subject Specifiers	Sc-0100	X
Article « Content type	c587bbc8739dbf2f6e46eea38414ad88	X
Assessing Pupils' Progress (APP) « Personalisation and Progression « Initiative	372bb0cff6454733d7ec7f0e8f8655ee	X
Assessment Focuses (AFs) « Assessing Pupils' Progress (APP) « Assessment « Teaching and learning « Educational topic	f78b8a255e2c429505b9d928e1e682bb	X
Biology « Science « Subject Specifiers	4b3a2582c1db93a6a9d1fba8a412c949	X
Education and skills « Integrated Public Sector Vocabulary		X

Figure 10. New tagging editor interface (mock-up)

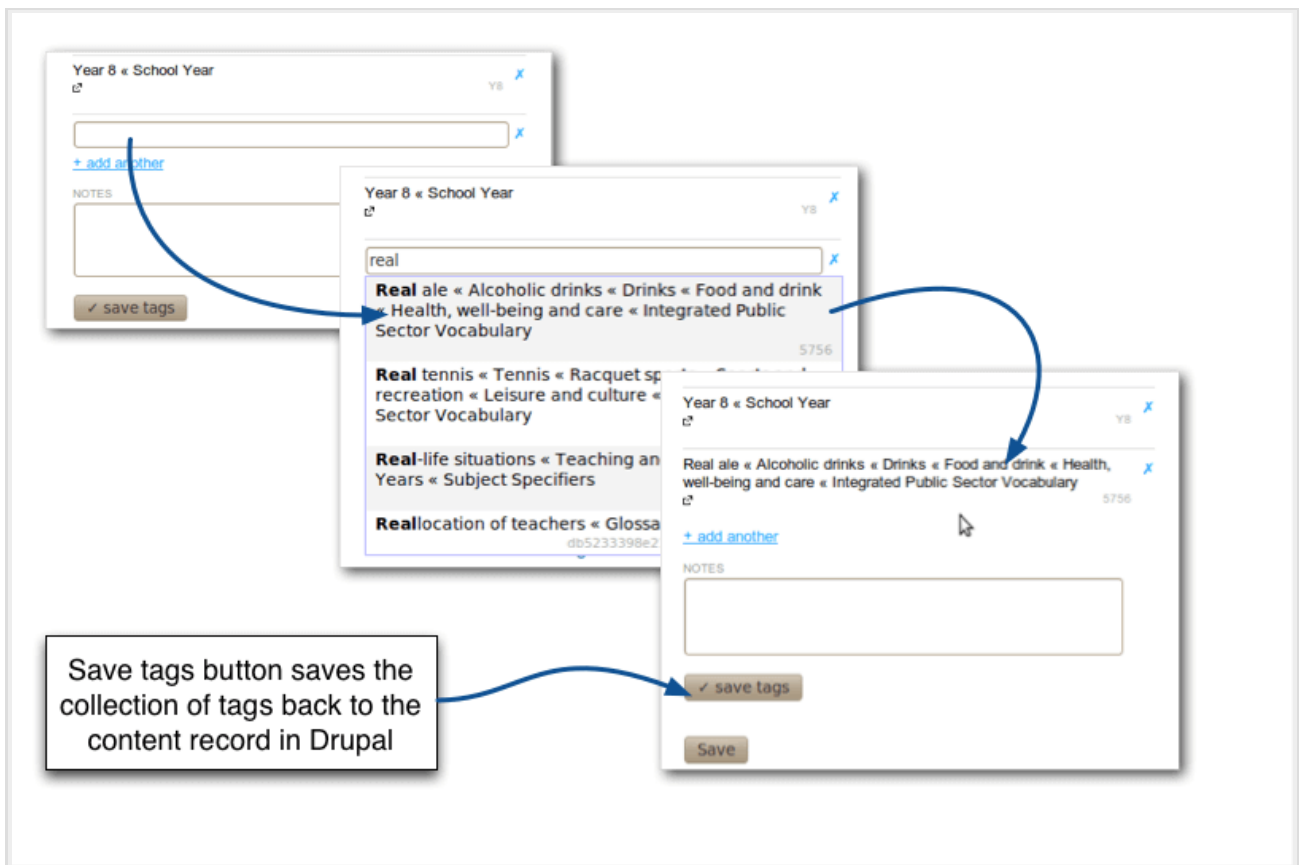


Figure 11. Finding and adding tags to content

## Technical components of the visual vocabulary editor

### Hosting and system requirements

Ubuntu Linux server, 256MB memory. Debian also works. All code is generally platform independent.

### Installation

To instal the dependencies on a fresh Ubuntu server:

```
sudo apt-get install gunicorn mysql-server python-django python-mysqldb python-xlrd python-docutils postfix nginx
```

Server configuration files are included.

Our server runs on [Linode](#).

### Technical description

Web based SKOS editor.

### List of code modules

- [The Django framework](#)

- [LXML](#) for parsing incoming SRU and SKOS RDF import. LXML uses [libxml](#) and [XPath](#), more about [parsing SRU With LXML & XPath](#)
  - [XLRD library](#) by John Machin, used by Excel import. [More information about the import algorithm](#)
  - [jQuery](#) and jQuery based:
    - [Django dynamic formset](#)
    - [Autocomplete](#)
    - Icons from [jQuery UI project](#)
  - [JavaScript InfoVis Toolkit](#) to create interactive data visualizations
  - [Ubuntu server](#)
  - [Nginx](#) front-end webserver [our configuration](#)
  - Gunicorn server or [Apache2](#) with [Mod\\_wsgi](#), more in [Django deployment docs](#)
  - [MySQL database](#), any [database supported by Django ORM](#) will work, including PostgreSQL & SQLite
- 

## Technical components of the content tagging plugin

[Chrome extension](#) is written in JavaScript ( plain native code, [DOM localStorage](#) ), CSS and HTML5. Tagging interfaces are loaded for each content node using AJAX. Tagging interface itself uses jQuery. Frontend uses AJAX to talk to backend WSGI application, which communicates to underlying CMS. SRU is used to retrieve data from the CMS. SWORD is used to post amended data back.

### System requirements

[Chrome](#) or [Chromium](#) web browser.

### Technical description

Tagging plug-in is implemented as a browser extension for Google Chrome and Chromium browsers.

It is CMS independent interface that uses integration with the content management system to be tagged.

This can be achieved by hooking into existing APIs of the client as in the following example.

### Tagging integration dependencies for Drupal CMS, legacy of The National Strategies

- **SRU** (Search/Retrieval via URL), a standard XML-focused search protocol for Internet search queries, utilizing CQL (Contextual Query Language), a standard syntax for representing queries.
- **SWORD** (Simple Web service Offering Repository Deposit), XML protocol for depositing

content from one location to another. It is a profile of the Atom Publishing Protocol (known as APP or ATOMPUB).

- **CQL** (Contextual Query Language), a formal language for representing queries to information retrieval systems such as search engines, bibliographic catalogs and museum collection information. It's design objective is that queries be human readable and writable, and that the language be intuitive while maintaining the expressiveness of more complex query languages. It is being developed and maintained by the Z39.50 Maintenance Agency, part of the Library of Congress.

Modules: [SRU](#), [SWORD](#), [CQL](#), W3C standard [SKOS export/import for Drupal](#).