

Article

[Benjamin De Boe](#) · Jun 14, 2016 5m read

[Open Exchange](#)

iKnow demo apps (part 3) - Dictionary Builder Demo

This is the third article in a series on iKnow demo applications, showcasing how the concepts and context provided through iKnow's [unique bottom-up approach](#) can be used to implement relevant use cases and help users be more productive in their daily tasks. Previous articles discussed the [Knowledge Portal](#), a straightforward tool to browse iKnow indexing results, and the [Set Analysis Demo](#), in which you can use the output of iKnow indexing to organize your texts according to their content, such as in patient cohort selection.

This week, we'll look into another demo application, the Dictionary Builder demo, in which we'll marry iKnow's bottom-up insights with top-down expertise, organizing our domain knowledge into dictionaries that are composed of the actual terms used in the data itself. Sticking to a top-down approach only, you'd risk missing out on some terminology used in the field that a domain expert sitting in his office wouldn't be aware of.

Installation and setup

This demo is available for download from a [GitHub repository](#), where you can either check it out using your preferred Git client, or download it as a zip outright. You should be able to import and compile all classes in any release of Caché starting with 2015.2.

Like the [Set Analysis demo](#), the application consists of a single (CSP) page with [AngularJS](#)-infused HTML and associated JavaScript code that calls a REST interface. To make the application work, you'll need to set up a web application that redirects to the `Demo.DictionaryBuilder.RestHandler` class, either manually or through the `Demo.DictionaryBuilder.Utils.CreateRestWebApp()` method included in the package.

While this demo is meant to work with any domain, that domain needs to be editable in order to save your work. Domains managed by [domain definition classes](#) by default can only be updated through that class and will refuse attempts made by the demo GUI, as described [in this article](#). You can enable editing the domain by setting the top-level `allowCustomUpdates` attribute to true in the XML. For the screenshots below, we used (somewhat unpolished) data from electronic health records, starting from a simple domain definition such as this one:

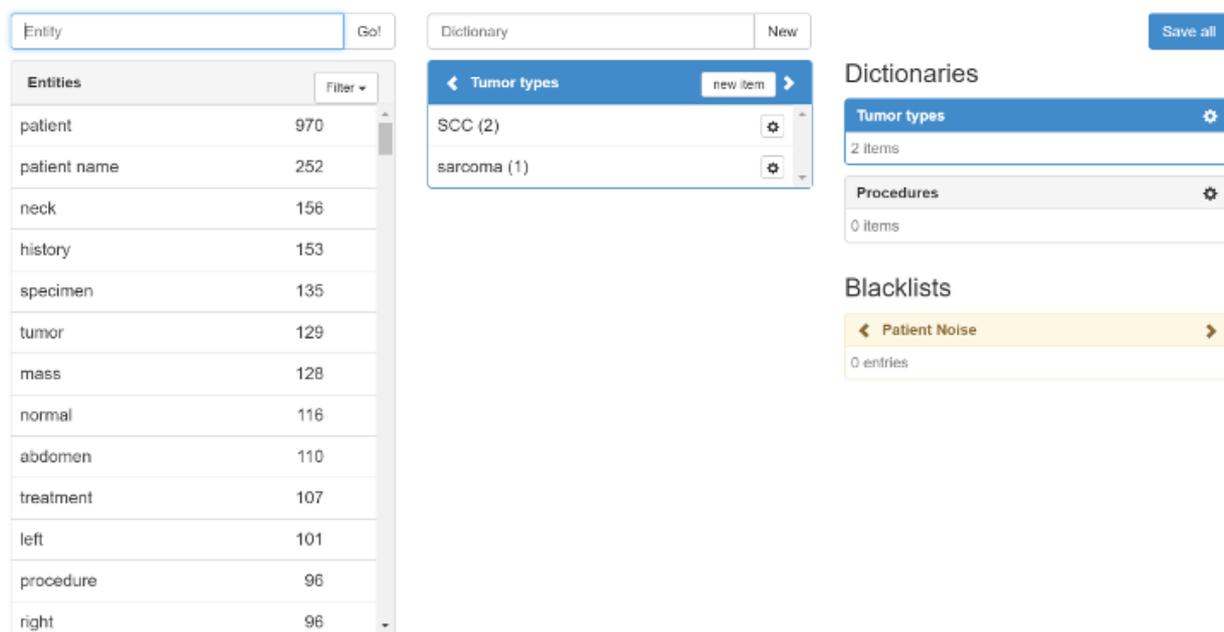
```
<domain name="NETS data" allowCustomUpdates="true">
  <data>
    <table tableName="nets.data" idField="ID" groupField="mrn" dataFields="txt" />
  </data>
  <matching dropBeforeBuild="false">
    <dictionary name="Tumor types">
      <item name="SCC" uri=":tumor:scc">
        <term string="squamous cell carcinoma" />
        <term string="SCC" />
      </item>
      <item name="sarcoma" uri=":tumor:sarcoma">
        <term string="sarcoma" />
      </item>
    </dictionary>
    <dictionary name="Procedures">
    </dictionary>
  </matching>
```

```
<blacklist name="Patient Noise">  
</blacklist>  
</domain>
```

The idea of the app is to build a dictionary of real-world terminology that helps us organize and highlight the EHRs according to the properties of the tumors the patients were suffering from. A few terms are already in the dictionary, but we'll add more, based on the real data rather than starting from textbook knowledge only.

Exploring the data

So let's open our demo interface by accessing the following URL, where the number at the end should correspond to the domain ID of the domain you wish to work with: <http://localhost:57772/csp/user/DictionaryBuilder.csp?1>. On the left, you'll see a list of the top concepts for this domain, similar to what we saw in the [Knowledge Portal](#) earlier on. On the right, you'll see the list of [dictionaries](#) currently defined for your domain, one of which is expanded to the item level in the central column.



Now, again similar to how the earlier demos worked, let's start by typing something in the top left box and pressing the "Go!" button. That'll show you the similar entities for the fragment you just entered. For example, if you type "squa" and press the button, you'll see all the similar concepts iKnow identified, which includes many more subtle variations of squamous cell carcinoma (SCC), that would have been hard to enumerate upfront, or recombine with those subtle qualifiers ("moderately differentiated scc") that were added by the original text author.

Creating dictionaries

So we have our real-world bottom-up data on the left, and our top-down dictionary-organized knowledge on the right. Now you can enrich your dictionaries by clicking an entity on the left and dragging it over to the dictionary item for which you want to add the entity as a term.

The screenshot shows the Dictionary Builder interface. On the left, there is a search bar with 'squa' and a 'Go!' button. Below it is a list of entities with columns for the entity name, a count, and a star icon. The entities listed are: squamous cell carcinoma (51), metastatic squamous cell carcinoma (18), squamous (8), m0 squamous cell carcinoma (3), recently diagnosed squamous cell carcinoma (3), focal squamous cell carcinoma (2), invasive squamous cell carcinoma (2), metastatic keratinizing squamous cell carcinoma (2), moderately differentiated squamous cell carcinoma (2), primary squamous cell carcinoma (2), and squamous cell carcinoma patient name (2). In the center, a dictionary titled 'Tumor types' is expanded, showing a 'new item' button and a list of terms: SCC (2), metastatic squamous cell carcinoma (18), and sarcoma (1). On the right, there is a 'Save all' button and a sidebar with 'Dictionaries' (showing 'Tumor types' with 2 items and 'Procedures' with 0 items) and 'Blacklists' (showing 'Patient Noise' with 0 entries).

If you drag an entity to the "new item" button at the top of the expanded dictionary, or straight onto a collapsed one in the list on the right, it will be added as a new term. If you drop it on the cog icon, you'll see a dialog allowing you to edit the terms for that item.

The screenshot shows a dialog box titled 'Dictionary Item: chemotherapy'. It has a close button (X) in the top right corner. The dialog contains the following fields and sections: 'Item name' with a text input field containing 'chemotherapy'; 'Item URI' with a text input field containing ':Procedures:chemotherapy'; and 'Terms' with a plus sign icon and a list of three terms: 'chemotherapy', 'chemo', and 'chemotherapies'. Each term has a close button (X) to its right. At the bottom of the dialog, there are two buttons: 'Remove' and 'Apply'.

You can also create new dictionaries using the text box in the top center of the screen, for example to create an additional dictionary for topologies, another aspect we wish to know more about and would like to learn the real-

world terminology for. After adding more dictionaries, items and terms, your screen may look like this:

The screenshot shows the Dictionary Builder interface. On the left, there is a table of entities with a search bar and a 'Go!' button. The table lists various laparotomy-related terms and their counts. In the center, there is a 'Dictionary' section with a 'New' button and a list of procedures with their counts and a 'new item' button. On the right, there is a 'Dictionaries' section with a 'Save all' button and a list of dictionaries with their item counts. Below that is a 'Blacklists' section with a 'Patient Noise' blacklist containing 0 entries.

Entities	Filter	
exploratory laparotomy		10
status post exploratory laparotomy		3
laparoscopic		2
laparoscopic diverting colostomy		2
mini laparotomy		2
exploratory laparotomy 2		1
exploratory laparotomy 3		1
exploratory laparotomy gross description		1
laparoscopic bilateral lymph node dissection		1
laparoscopic cholecystectomy		1
laparoscopic diverting colonoscopy		1
laparoscopic robotic prostatectomy		1

Dictionary	New
Procedures	new item
treatment (2)	
biopsy (1)	
chemotherapy (3)	
surgery (1)	
ct scan (1)	
radiation therapy (1)	

Dictionaries

- Tumor types (4 items)
- Procedures (6 items)
- Topology (10 items)

Blacklists

- Patient Noise (0 entries)

Clearing things up

While it's clear this approach is helpful in validating whether your top-down insights correspond to the terminology used in the field and use that to enrich your own knowledge, it also brings up a certain amount of noise that's obstructing the view somewhat. For this, you can use [blacklists](#), which are simple lists of terms you wish to exclude from query results (as we saw in the [knowledge portal demo](#)). You can populate these (orange bar in the lower right corner) in the same drag-and-drop fashion as you populated your dictionaries, and they'll be removed from the list of top/similar terms on the left straight away.

The screenshot shows the Dictionary Builder interface. On the left, there is a table of entities with a search bar and a 'Go!' button. The table lists various terms and their counts. In the center, there is a 'Dictionary' section with a 'New' button and a list of procedures with their counts and a 'new item' button. On the right, there is a 'Dictionaries' section with a 'Save all' button and a list of dictionaries with their item counts. Below that is a 'Blacklists' section with a 'Patient Noise' blacklist containing 5 entries.

Entity	Go!
procedure	96
right	96
negative	95
no evidence	85
time	83
clear	82
biopsy	81
chest	80
bladder	79
skin	78
soft	74
chemotherapy	70
surgery	67

Dictionary	New
Procedures	new item
treatment (2)	
biopsy (1)	
chemotherapy (3)	
surgery (1)	
ct scan (1)	
radiation therapy (1)	
exploratory laparotomy (1)	

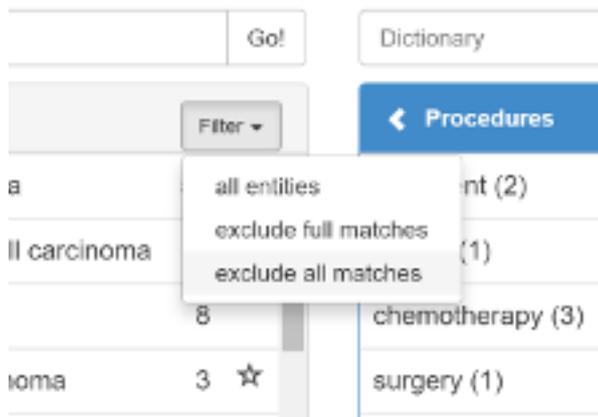
Dictionaries

- Tumor types (4 items)
- Procedures (7 items)
- Topology (10 items)

Blacklists

- Patient Noise (5 entries)
time 83

Separately, to filter the list of entities on the left to exclude the ones you have already mapped to a dictionary, use the "filter" dropdown in the entities' header bar.



Note: Remember to press the "Save all" button when you have finished modifying your dictionaries and blacklists, as all changes are only retained on the client side until you click that button. That will also fully refresh the star icons identifying matched terms.

Similar to the [Set Analysis demo](#) last week, this generic demo shows you how you can work with iKnow output to explore and organize your data. Gradually moving to more high-level use cases, we'll build upon this demo in next week's article, where we'll combine the dictionaries created here into rules we'd like to match against our data.

[#InterSystems Natural Language Processing \(NLP, iKnow\)](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/iknow-demo-apps-part-3-dictionary-builder-demo>