

OpenInfobutton Tailoring Guide

Table of Contents

[1 Software Requirements 2](#_Toc354411770)

[2 Target Audience 2](#_Toc354411771)

[3 Background 2](#_Toc354411772)

[3.1 OpenInfobutton processing 3](#_Toc354411773)

[3.1.1 Matching process 3](#_Toc354411774)

[3.2 Knowledge resource response 3](#_Toc354411775)

[4 Creating Knowledge Resource Profiles 3](#_Toc354411776)

[4.1 Opening Authentic and the OpenInfobutton Tailoring Tool 4](#_Toc354411777)

[4.2 Creating a new knowledge resource profile 4](#_Toc354411778)

[4.3 Knowledge resource profile sections 5](#_Toc354411779)

[4.3.1 Knowledge resource profile metadata 5](#_Toc354411780)

[4.3.2 Knowledge resource profile context 7](#_Toc354411781)

[4.3.3 HL7 Compliant Resources 9](#_Toc354411782)

[4.3.4 Subtopic 11](#_Toc354411783)

[4.3.5 Non-HL7 compliant resources 12](#_Toc354411784)

[5 Validating and testing 12](#_Toc354411785)

# Software Requirements

* Windows XP, Windows Vista, or Windows 7
* Altova® Authentic® 2012 Desktop Community Edition (free download)
  + Authentic® is an XML authoring environment that enables users without XML knowledge to create XML documents through Windows-based data entry forms. The OpenInfobutton Tailoring Tool is based on Authentic® forms.
  + Authentic® can be downloaded from <http://www.altova.com/download/authentic/desktop_community.html>
  + Authentic® user guides are available online at <http://manual.altova.com/AuthenticDesktop/authenticdesktopcommunity/>; and in PDF at <http://www.altova.com/documents/AuthenticDesktop.pdf>

# Target Audience

This guide is aimed at those who will be in charge of maintaining OpenInfobutton’s knowledge base. The ideal type of professional to perform the OpenInfobutton knowledge base maintenance is an informaticist, a medical librarian, or an information systems analyst with prior experience in healthcare. Although OpenInfobutton maintenance does not require knowledge of computer programming, some basic knowledge of XML, infobuttons, and health care terminologies is desirable. Before reading this user guide, the reader is advised to go over some of the infobutton material available at <http://www.openinfobutton.org/infobutton-knowledge-base>.

In addition, this guide assumes basic knowledge of the Altova® Authentic® XML editing tool. An online tutorial of Authentic can be found at <http://manual.altova.com/AuthenticDesktop/authenticdesktopcommunity/>.

# Background

OpenInfobutton is a Web service that enables the integration between electronic health record (EHR) systems and online health knowledge resources via tools known as “infobuttons.” Infobuttons use the context of the interaction between a user and an EHR system to 1) automatically determine resources that might be relevant in a particular context; and 2) to dynamically create hyperlinks to relevant content topics within these resources.

OpenInfobutton’s core is a knowledge base composed of a set of XML files called *knowledge* *resource profiles*. A resource profile contains all the information needed for OpenInfobutton to link with a particular knowledge resource, including:

* The healthcare organizations that have access to the knowledge resource.
* Resource authentication for subscription-based resources.
* The contexts in which a particular resource is considered to be relevant. Context is defined in terms the *task* associated with the electronic health record (EHR) module in which an infobutton is located (e.g., order entry, problem list review, lab results review) ; the *user* who clicks on an infobutton (e.g., healthcare provider, patient); *patient* characteristics(e.g., gender, age); *care setting* (e.g., outpatient, inpatient); and the main clinical concept associated with the infobutton (e.g., a medication, a lab test result, a problem)*.*
* Specific information about the resource application program interface (API), such as the base URL (URL of the knowledge resource search engine); the search parameters accepted by the resource API; and the terminologies understood by the resource API.
* The questions or content subtopics (e.g., drug dosing contraindications, side effects) that the resource is able to address.

Each knowledge resource is configured in one distinct XML file. This guide explains how to create and maintain these files using a Windows-based tool called Altova® Authentic®.

## OpenInfobutton processing

Two main steps are processed against the OpenInfobutton knowledge base when OpenInfobutton receives an infobutton request: knowledge resource *matching* and knowledge resourcerequest *creation*.

### Matching process

Knowledge resource matching consists of identifying the resources in the OpenInfobutton knowledge base that are most relevant in the context conveyed by the infobutton request. In other words, the OpenInfobutton tries to match the infobutton request context parameters to those described in each knowledge resource profile and returns the resources that match the request context.

As part of the matching process, OpenInfobutton executes the following steps for every knowledge resource profile stored in the OpenInfobutton knowledge base:

1. Verifies whether the resource is available at the organization that submitted the infobutton request. If not, proceed to the next resource profile.
2. For each *context* parameter in the infobutton request, verify if the parameter value in the infobutton request matches one of the values in the knowledge resource profile matching sets. If one of the *context* parameters does not match, proceed to the next resource.
3. If all context parameters in the infobutton request match the matching sets in the resource profile, the knowledge resource is considered relevant.

## Knowledge resource response

The OpenInfobutton response to an infobutton request includes links to knowledge resources that matched successfully against the context parameters represented in the infobutton request. For each matched resource, a set of links is created. Each link is associated with a content subtopic that is addressed in the knowledge resource.

# Creating Knowledge Resource Profiles

This section provides a step-by-step set of instructions on how to create OpenInfobutton knowledge resource profiles. To get started, you will need to open the Authentic® tool.

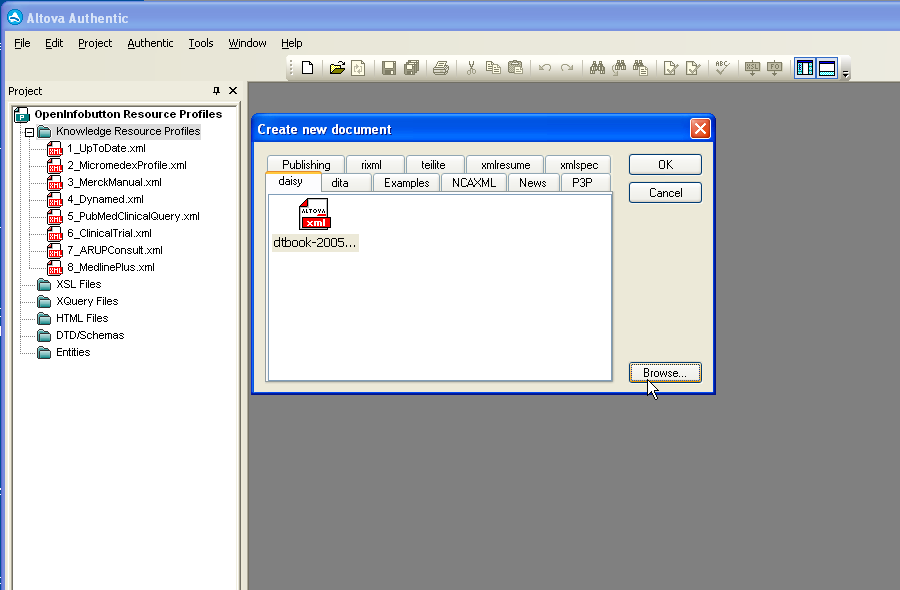
## Opening Authentic and the OpenInfobutton Tailoring Tool

The OpenInfobutton Tailoring Tool can be opened in two ways:

* Option 1: Open windows explorer, go to C:\Program Files\OpenInfobutton Tailoring Tool\ and double-click the file “OpenInfobutton Resource Profiles.spp”. This file is an Authentic® project file that contains a started set of resource profiles as well as any new profile you create.
* Option 2: Open the Authentic® program. Select File->Open Project. Go to C:\Program Files\OpenInfobutton Tailoring Tool\ and choose “OpenInfobutton Resource Profiles.spp”.

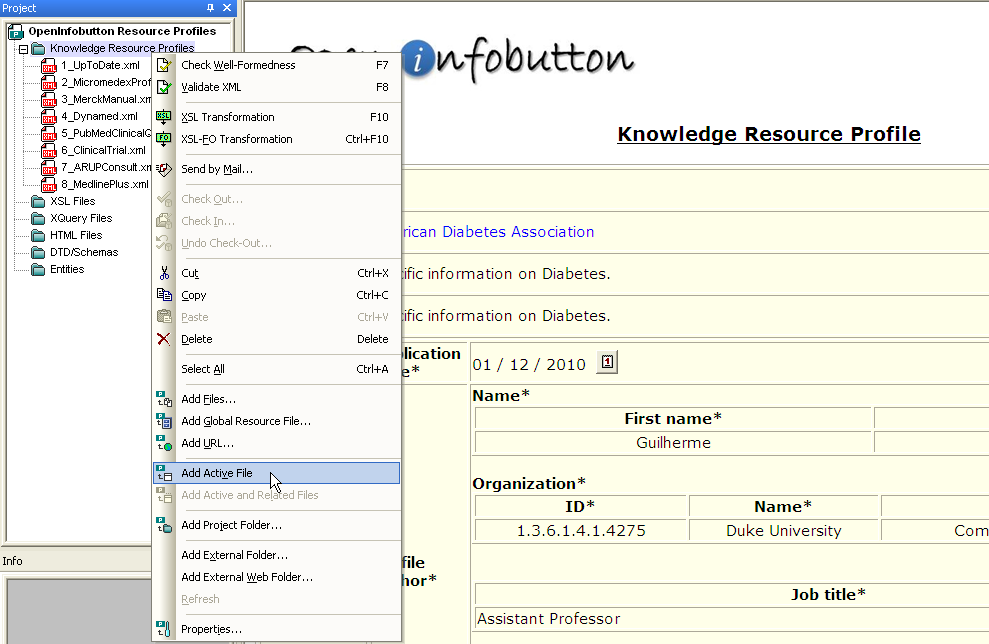
## Creating a new knowledge resource profile

* Click File->New and then click the “Browse” button on the pop-up screen. Select the c:\Program Files\OpenInfobutton Tailoring Tool\ResourceAuthoring.sps file. The knowledge resource tailoring form will open.
* Select File->Save and save the file to the same folder where the ResourceAuthoring.sps is located (c:\Program Files\OpenInfobutton Tailoring Tool\).



You should also add the new file to the OpenInfobutton knowledge base project.

* Right click the Knowledge Resource Profiles folder and choose “Add Active File” from the drop-down menu.

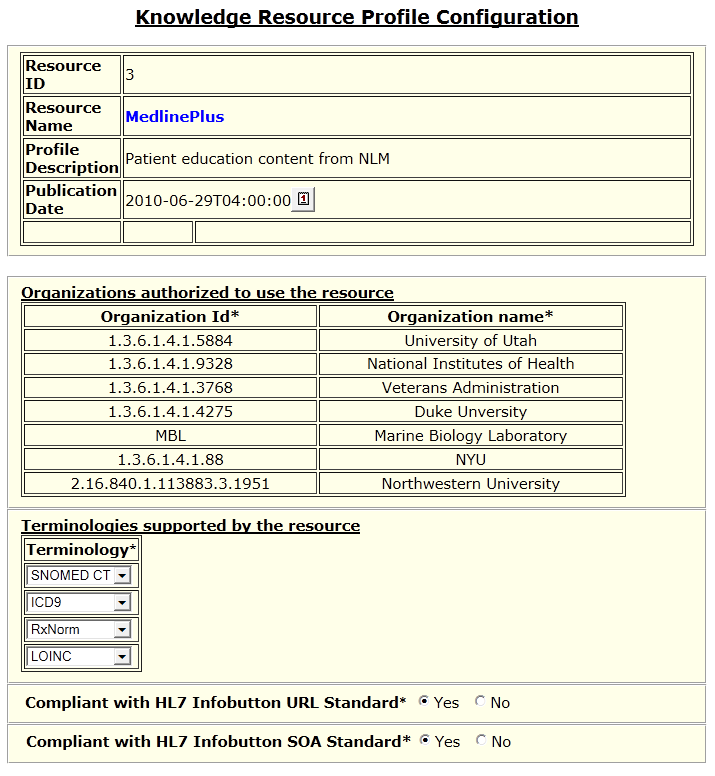


## Knowledge resource profile sections

A knowledge resource profile is composed of the following sections: *metadata* and *context*. The metadata section contains fields to help you document the profile and changes to it over time. The context section contains knowledge resource processing information, including the contexts in which the resource is relevant, information about the resource API, and subtopics covered by the resource. A knowledge resource must have one or more contexts. Each context includes a set of context parameter settings and a list of subtopics that are relevant in this particular context. The next sections of this document will go over the metadata and context sections of the resource profile.

### Knowledge resource profile metadata

The following screen shows a metadata example of the MedlinePlus profile. Descriptions of the metadata fields are provided below. Required fields are displayed in bold and with an asterisk next to the field label.

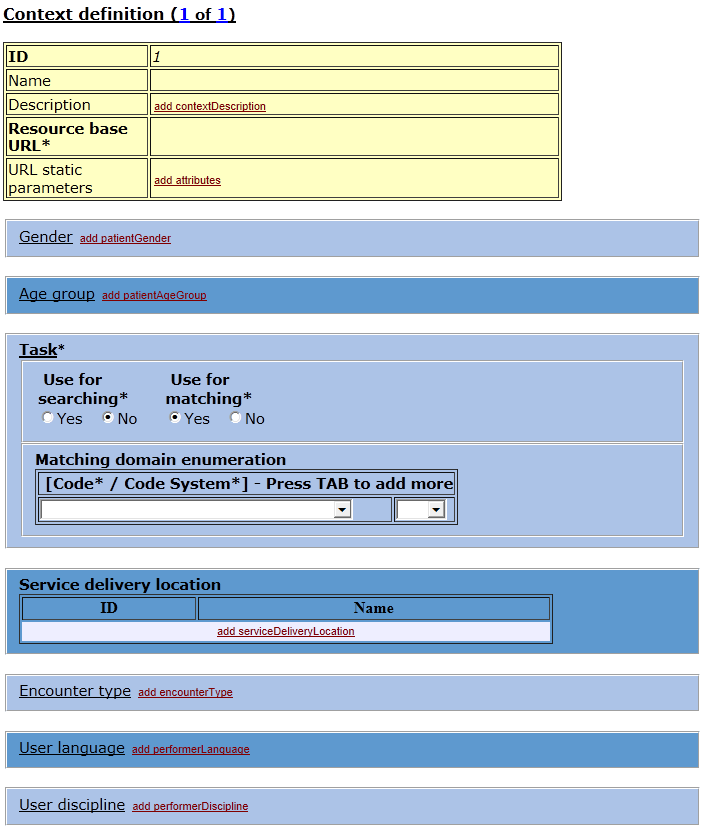


Fill out the fields as follows:

* *Resource ID*: a unique identifier for the resource profile. Currently this identifier is not used by OpenInfobutton’s logic, but an integer must be entered for the profile to validate.
* *Resource name*: a human-readable title for the resource. This is the resource name that will be displayed to OpenInfobutton users.
* *Profile description*: any kind of documentation about the resource content and its API.
* *Publication date*: Last date the resource was changed.
* *Organizations authorized to use this resource*: indicates the organizations that have a subscription to the resource. For freely available resources, indicates the organizations that would like to have access to this resource via OpenInfobutton.
  + *Organization ID*: the globally unique identifier used by the organization’s EHR to communicate with OpenInfobutton. Any globally unique identifier can be used, but OIDs (Object Identifier) are strongly recommended. Most organizations in the US have an OID. Organization OIDs can be searched at the ISO OID registry at <http://www.oid-info.com> . The OID of the Veterans Administration is 1.3.6.1.4.1.3768.
  + *Organization name*: a human-readable name of the organization.
* *Terminologies supported by the resource*: indicates the standard terminologies that are supported by the knowledge resource.
* *Compliant with HL7 URL Standard*: This specification has a standard format for infobutton requests using URLs. See [OpenInfobutton.org](http://www.openinfobutton.org/hl7-enabled-knowledge-resources) for a list of HL7-compliant resources.
* *Compliant with HL7 Infobutton SOA Standard*: This specification includes a standard format for infobutton responses and a RESTful implementation. See [OpenInfobutton.org](http://www.openinfobutton.org/hl7-enabled-knowledge-resources) for a list of HL7-compliant resources.
* *URL style*: used only for resources that are not HL7 compliant. Options are “CLEAN” and “DIRTY”. Dirty URLs follow a pattern consisted of a list of parameter names and values separated by “&”. For example, [http://website.com/searchengine?search=diabetes&gender=F]. Most knowledge resources use dirty URLs. Clean URLs follow a pattern consisted of parameter where names and values are somewhat disguised in the URL. Without knowledge of the resource API, it is difficult to figure out the parameter names used by the API. For example, the example above in a clean URL would look as follows: [http://website.com/searchengine/search/diabetes/gender/F/].

### Knowledge resource profile context

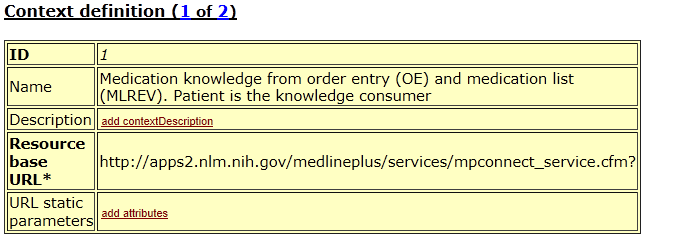
A resource profile must contain one or more context. Each context definition has a metadata section (yellow box) and a set of context parameters (blue boxes). The following screen shows the fields of a blank profile context definition.



The yellow section has the following fields:

* *ID*: a unique identifier for the context. OpenInfobutton does not use this field in its logic. Any integer can be used.
* *Name*: a human-readable name for the profile for documentation purposes.
* *Resource base URL*: the URL of the resource search API. Typically the base URL starts with http or https and goes until the question mark “?”.
* *URL static parameters*: a list of parameters whose values are immutable and will always be included in URLs that OpenInfobutton generates for the resource.

The following screen shot illustrates an example of the yellow section of MedlinePlus:

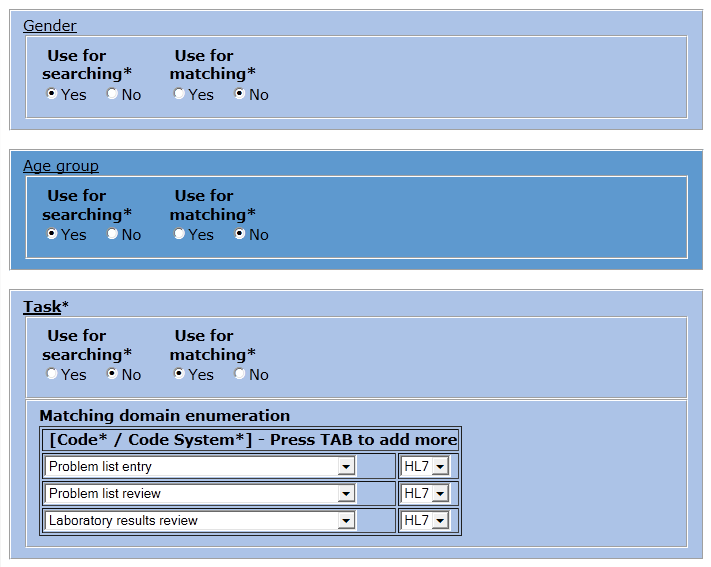


### HL7 Compliant Resources

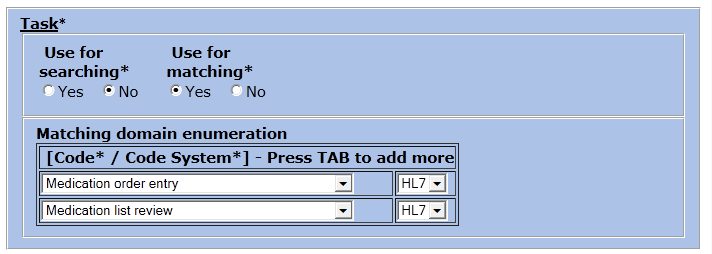
The blue section has a list of available context parameters. When creating a new profile, context parameters are displayed as collapsed elements with an “add...” link on the right side of each context label. Only the context parameters that are used for matching the resource and/or searching against the resource API need to be configured in the profile, except for *task* and *concept of interest,* which are required context parameters. Each context parameter is described below:

* *Gender*: gender of the patient represented in an infobutton request.
* *Age group*: a code that represents the patient's age group.
* *Task*: a code that represents the task that is being performed in an EHR/PHR system.
* *Service delivery location*: clinical facility where the patient care is taking place at the time the infobutton request is initiated. May be used to identify potential local knowledge resources (e.g., local protocols) that are specific to the facility where the patient care is being delivered
* *Encounter type*: a code that represents the type of encounter (e.g., inpatient, outpatient).
* *User language*: the preferred language of the EHR user.
* *User discipline*: the discipline of the EHR user (e.g., Physician, Registered Nurse).
* *User type*: indicates whether the user is a provider or a patient.
* Information recipient language: the preferred language of the person who will consume the content retrieved by OpenInfobutton.
* *Information recipient discipline*: the discipline of the person who will consume the content retrieved by OpenInfobutton (e.g., Physician, Registered Nurse).
* *Information recipient type*: indicates whether the person who will consume the content retrieved by OpenInfobutton is a provider or a patient.
* *Concept of interest*: represents the main clinical data of interest in an infobutton request (e.g., a medication, a lab test result, a disease in the patient's problem list).
* *Subtopic*: indicates one or more specific topics of interest associated with the concept of interest (e.g., indications, contra-indications, dose).

Whenever a context parameter definition is added, the following check box fields are displayed: *Use for* *searching* and *Use for* *matching.*



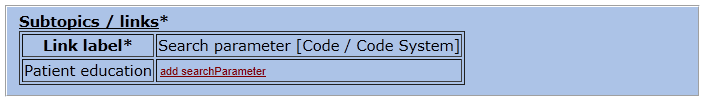
* *Use for searching*: if yes, indicates that the resource API is able to process the parameter for searching content. When this option is checked, OpenInfobutton will add this parameter to the links it generates for the resource.
  + In the example above, *gender*, *age group*, *and task* will be used when searching MedlinePlus. Nothing else needs to be done if the resource is HL7 compliant.
* *Use for matching*: if yes, the parameter will be used to determine whether the resource is relevant in a particular EHR context; an option to add a *matching domain enumeration* is displayed. The matching domain enumeration is a set of codes in which a resource is relevant. To add new items, position the cursor at one of the existing items and press the TAB key.
  + In the example below, MedlinePlus will be considered a good “match” only for requests that originate from *order entry* or *medication list review* infobuttons. The pick lists on the right side indicates the code’s code system. For example, EHR task codes are present in the HL7 code system.

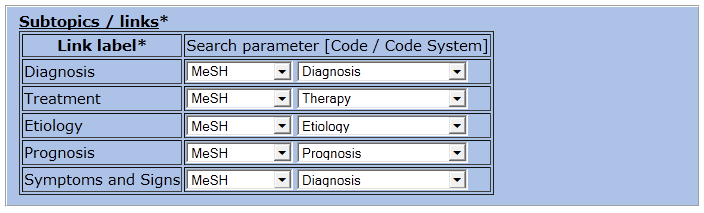


### Subtopic

The *Subtopic* parameter defines the set of content subtopics that are considered to be relevant in a particular context. If a particular context matches an infobutton request, the items listed under *Subtopic* will be offered to the OpenInfobutton user.

* *Link label*: a human-readable label for the subtopic. This label may be used to display as a link in the OpenInfobutton user interface.
* *Search parameter*: the code and code system associated with the subtopic. These fields are optional and used only if the resource supports the subtopic parameter for searching.

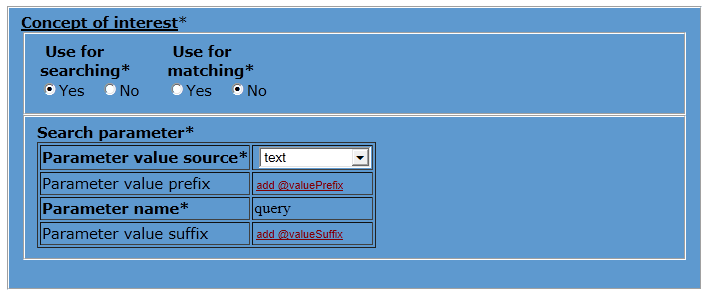
The example below shows a simple configuration for a subtopic. Since the entire resource is dedicated to patient education, the subtopic only provides a label for the user interface.

In the example below, 5 subtopics will be generated. Each subtopic is associated with a code that will be used to refine the search results.

### Non-HL7 compliant resources

When a resource is not compliant with HL7, a different set of fields is displayed if *Use for searching = “yes”* tospecify a custom parameter name used by the resource API as well as the source of value for this parameter.

* *Parameter value source*: indicates whether OpenInfobutton should use the parameter *code* or *text* (most common for non-HL7 resources) when searching the resource API.
* *Parameter name*: name of the parameter in the resource API. In the example below, the resource API concept of interest is called “query”.
  + If needed, a *prefix* and/or *suffix* may be added to parameter values as well. This is not common and typically not necessary in most resources. For example, if the parameter name is “searchCode” and the static suffix is “|ICD9”, OpenInfobutton would render this parameter in a URL searching for “210.00” as “searchCode=210.00|ICD9.



# Validating and testing

Once your resource profile is ready, you can validate it against its XML schema. This will verify whether all the profile required elements are filled out. To validate a profile, click on the XML->Validate menu or press the F8 key. If your profile is valid you shoud see a message indicating so.

After validating a profile, you can deploy it to be tested with OpenInfobutton.

* Use the Drupal admin tool that comes with OpenInfobutton’s installation package to load the profile to the OpenInfobutton database.
* After the file is copied you can test the profile using the Web-based OpenInfobutton Q&A tool. This tool can be launched from a Web browser and it is typically located at http://[OpenInfobutton server name]/infobutton-service/InfobuttonQA.htm. You can also test the profile using the OpenInfobutton demo at http://[OpenInfobutton server name]/ infobutton-service/OpenInfobuttonDemo.html.
* Ask your system administrator if you are not sure where to find these tools.