



OpenInfobutton Tailoring Guide

Table of Contents

1	Software Requirements	2
2	Target Audience	2
3	Background	2
3.1	OpenInfobutton processing	3
3.1.1	Matching process	3
3.2	Knowledge resource response	3
4	Creating Knowledge Resource Profiles	4
4.1	Opening Authentic®	4
4.2	Creating a new knowledge resource profile	4
4.3	Knowledge resource profile sections	6
4.3.1	Knowledge resource profile metadata	6
4.3.2	Knowledge resource profile context	8
4.3.3	Context parameters	10
4.3.4	Subtopic	13
4.3.5	Non-HL7 compliant resources	13

1 Software Requirements

- Windows XP, Windows Vista, or Windows 7
- Altova® Authentic® 2011 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>

2 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 <https://sites.google.com/site/infobuttonsforcprs/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/>.

3 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.

A resource profile also contains metadata that helps with the documentation of the resource profile, including authors, versions, modifications, and descriptions of each component of the profile.

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®.

3.1 OpenInfobutton processing

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

3.1.1 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.

3.2 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.

4 Creating Knowledge Resource Profiles

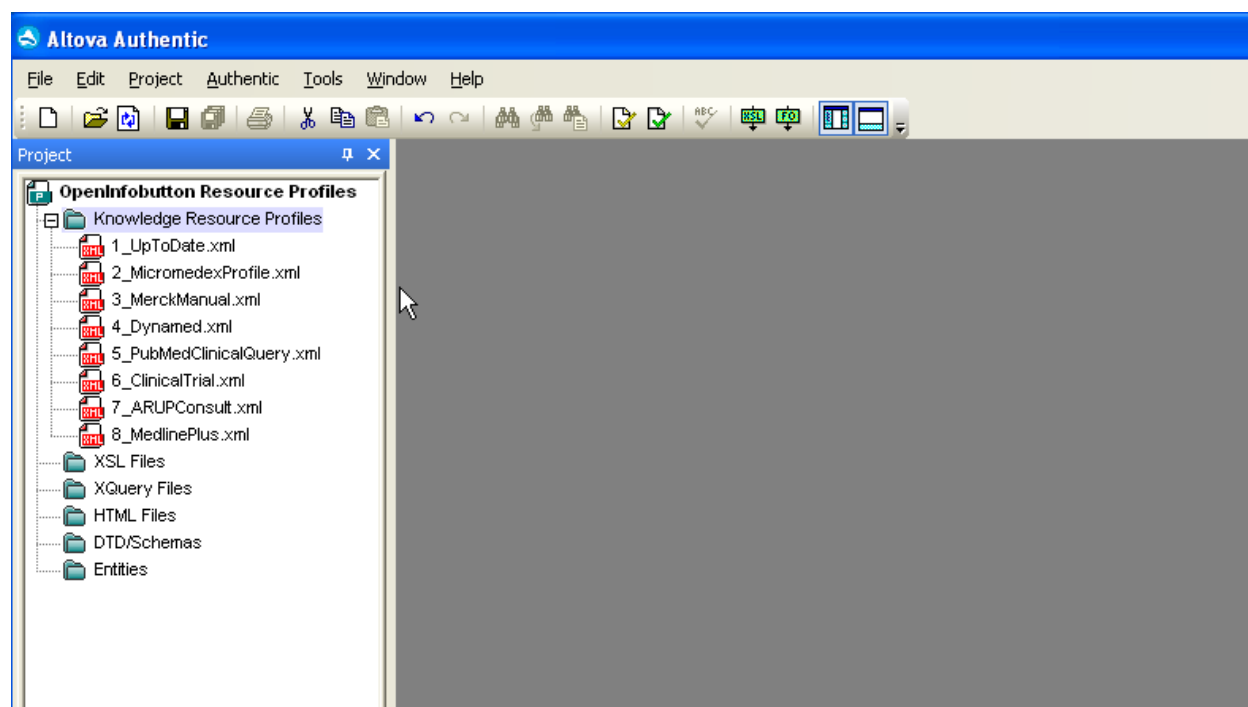
This section provides a step-by-step set of instructions on how to create OpenInfobutton knowledge resource profiles. To get started, you will need to open the Authentic® tool.

4.1 Opening Authentic and the OpenInfobutton Tailoring Tool

The OpenInfobutton Tailoring Tool can be opened in two ways:

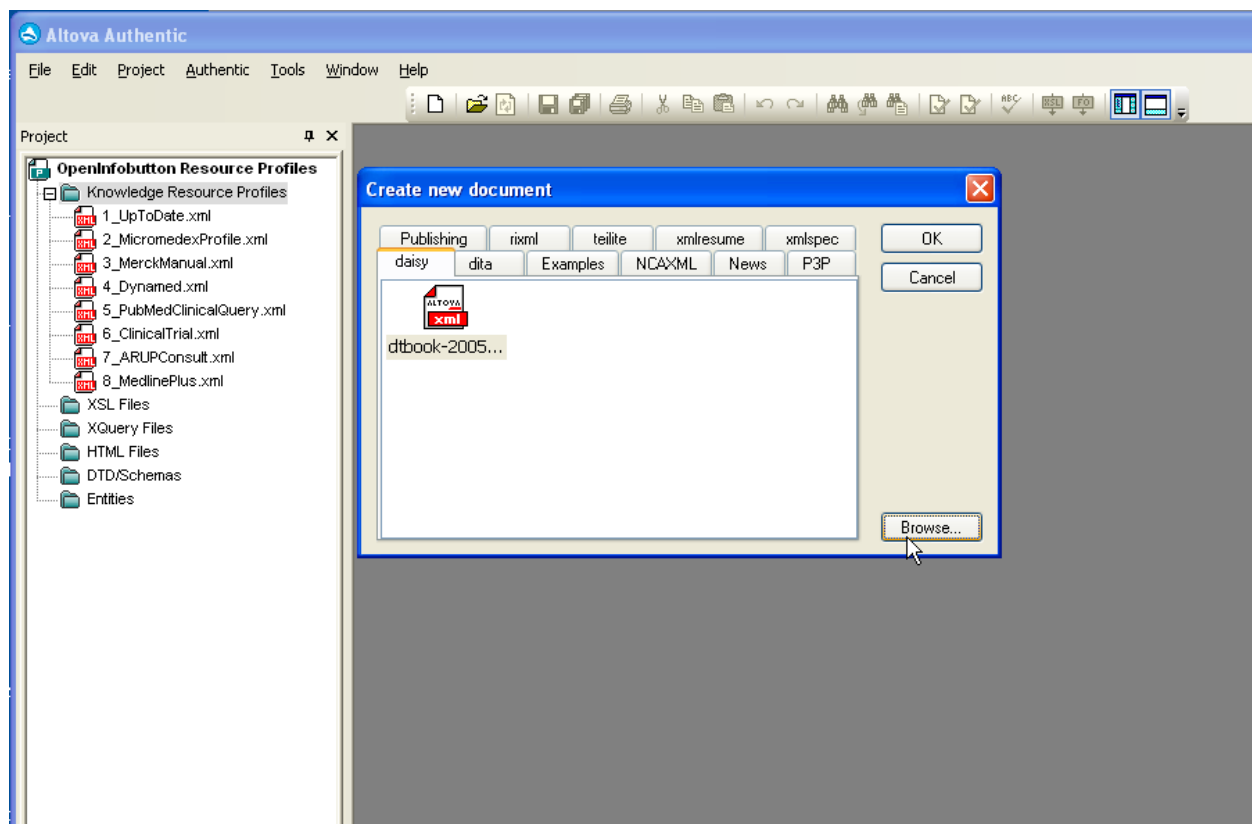
- Option 1: Open windows explorer, go to C:\Program Files\OpenInfobutton Tailoring Tool\ and double-click the file “OpenInfobutton Resource Profiles.spp”. This file is an Authentic® project file that contains a started set of resources profiles as well as any new profile you create.
- Option 2: Open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

Once the project is opened, you should see the following screen with a list of existing resource profiles on the left side. You can modify any of the existing resources by double-clicking the resource profile. The next sections will show how to create a new profile.



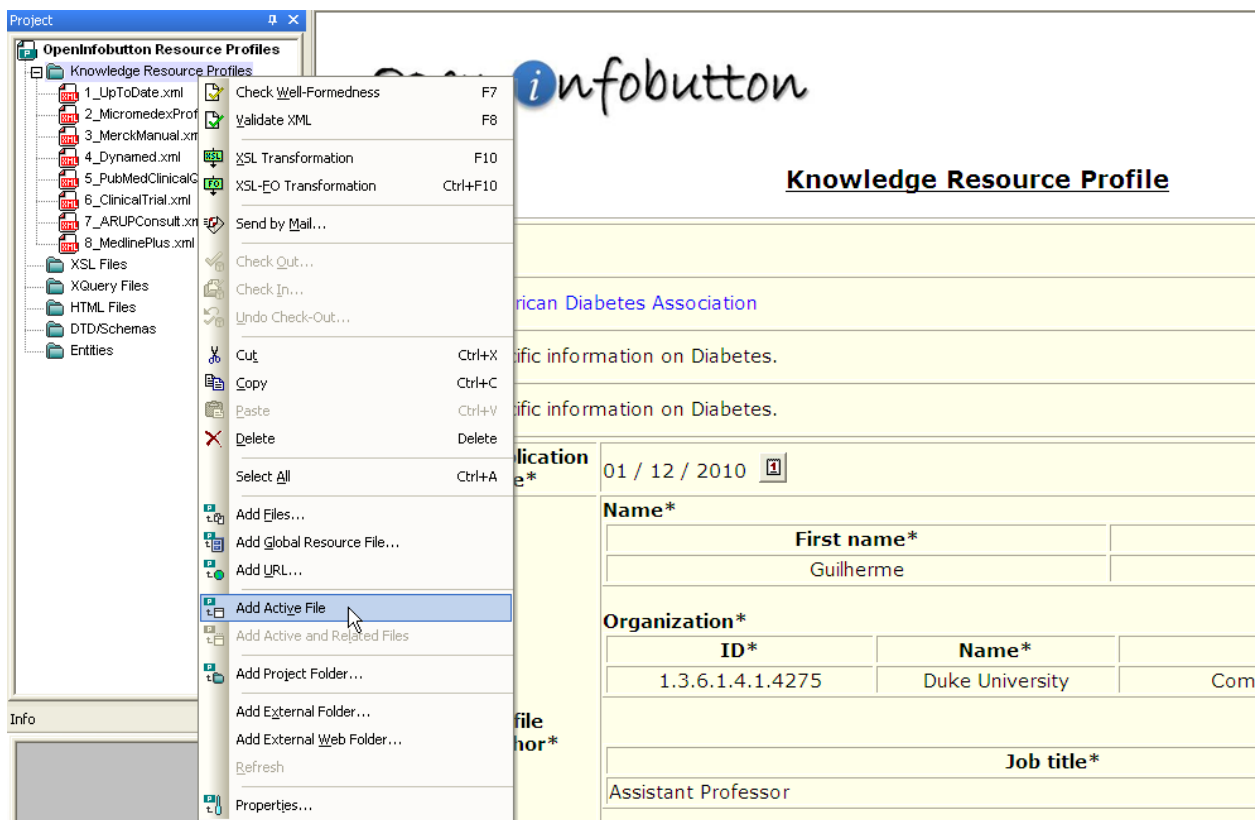
4.2 Creating a new knowledge resource profile

- 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.




4.3 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.

4.3.1 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.

Resource ID*	1				
Resource name*	MedlinePlus				
Resource purpose	Provides patient education information.				
Profile description*	A freely available patient education resource provided by the National Library of Medicine. Contains content on diseases, medications, and other health topics. MedlinePlus API is compliant with the HL7 Infobutton Standard.				
Version control*	Publication date*	08 / 01 / 2011 			
	Profile author*	Name*			
		First name*	Last name*		
		John	Doe		
		Organization*			
		ID*	Name*	Department*	
	1.3.6.1.4.1.3768	Veterans Health Organization	Portland VA Medical Center		
	Job title*				
	Systems Analyst				
	Contact information*				
e-mail*	phone				
john.doe@va.gov	999-888-1234				
Resource contact information	Sales	First name	Last name	e-mail*	Phone
		add sales	add sales	add sales	add sales
	Technical contact	First name	Last name	e-mail*	Phone
		add technicalContact	add technicalContact	add technicalContact	add technicalContact
	Technical support	First name	Last name	e-mail*	Phone
		add technicalSupport	add technicalSupport	add technicalSupport	add technicalSupport
Resource URL*	http://www.medlineplus.gov				
Organizations authorized to use this resource*					
Organization ID*		Organization name*			
1.3.6.1.4.1.3768		Veterans Health Organization			

Fill out the fields as follows:

- Resource ID: a unique identifier for the resource profile. The identifier needs to be an integer and it must be globally unique across all resource profiles in a given OpenInfobutton instance.
- Resource name: a human-readable title for the resource. This is the resource name that will be displayed to OpenInfobutton users.
- Resource purpose: description of the overall purpose of the resource. For example, this field can be used to describe the main purpose for tailoring this resource within OpenInfobutton.
- Profile description: a description about the resource content, its API, availability, etc.
- Version Control: publication date, the person responsible for the profile authoring, and the organization that the profile author works for.

- Contact information: resource sales contact, technical contact, and technical support contact.
- Resource URL: a link to the resource main Web site. Can also be used as a reference to the resource API online documentation if available.
- 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.

4.3.2 Knowledge resource profile context

The following screen shows the fields of a blank profile context definition.

Context definition (1 of 1):

HL7 compliant	<input type="checkbox"/> Yes
ID	1
Name	
URL style	add @uriStyle
Description	add contextDescription
Resource base URL	
URL static parameters	add attributes

Gender: [add patientGender](#)

Age group: [add patientAgeGroup](#)

Task: [add task](#)

Service delivery location: [add serviceDeliveryLocation](#)

Encounter type: [add encounterType](#)

User language: [add performerLanguage](#)

User discipline: [add performerDiscipline](#)

User type: [add performerKnowledgeUserType](#)

Information recipient language: [add informationRecipientLanguage](#)

Information recipient discipline: [add informationRecipientDiscipline](#)

Information recipient type: [add informationRecipientUserType](#)

Concept of interest: [add subjectCoverage](#)

Subtopic:
☐ Use for searching ☐ Use for matching
[add matchingDomain](#)

The yellow section has the following fields:

- HL7 compliant: indicates whether the knowledge resource API is HL7 compliant or not. HL7 compliant resources are easier to configure since it is natively supported by OpenInfobutton .
- ID: a unique identifier for the context. Any integer can be used, as long as it is unique within the resource profile.
- Name: a human-readable name for the profile.
- 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/].

- Description: a narrative description that explains the context parameter settings.
- 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. For example, an HL7-compliant patient education resource may have a static parameter name “informationRecipient” set with a fixed value of “PAT”.

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

HL7 compliant*	<input type="checkbox"/> Yes		
ID*	1		
Name*	Medication-related contexts.		
URL style	add @urlStyle		
Description	Medication knowledge from order entry (OE) and medication list (MLREV). Patient is the knowledge consumer.		
Resource base URL*	http://apps.nlm.nih.gov/medlineplus/services/2mplus.cfm?		
URL static parameters*	name	value	
	informationRecipient	PAT	

4.3.3 Context parameters

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 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: *search* and *match*.

Gender				
<input checked="" type="checkbox"/> Search* <input type="checkbox"/> Match*				
Age group add patientAgeGroup				
Task* <input type="checkbox"/> Search* <input checked="" type="checkbox"/> Match* Matching domain enumeration [Code / Code System] - Press TAB to add more <table border="1"> <tr> <td>Order entry</td> <td>HL7</td> </tr> <tr> <td>Medication list review</td> <td>HL7</td> </tr> </table>	Order entry	HL7	Medication list review	HL7
Order entry	HL7			
Medication list review	HL7			
Service delivery location add serviceDeliveryLocation				
Encounter type add encounterType				
User language add performerLanguage				
User discipline add performerDiscipline				
User type add performerKnowledgeUserType				
Information recipient language				
<input checked="" type="checkbox"/> Search* <input checked="" type="checkbox"/> Match*				
Information recipient discipline add informationRecipientDiscipline				
Information recipient type add informationRecipientUserType				
Concept of interest* <input checked="" type="checkbox"/> Search* <input type="checkbox"/> Match*				

- Search: if checked, 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 links to the resource.
 - In the example below, *gender*, *subtopic*, *language*, and *concept of interest* will be used when searching MedlinePlus. Nothing else needs to be done if the resource is HL7 compliant.
- Match: if checked, the parameter will be used to determine whether the resource is relevant in a particular EHR context. If *match* is checked, an option to add a *matching domain enumeration* is displayed. The matching domain enumeration is a set of codes for which a resource is

relevant. To add new enumeration items, position the cursor at one of the existing enumeration 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 indicate the code system that the code belongs in. For example, EHR task codes belong in the HL7 code system.

Task*

☐ Search* ☒ Match*

Matching domain enumeration [Code / Code System] - Press TAB to add more

Order entry	HL7
Medication list review	HL7

4.3.4 Subtopic

The *Subtopic* parameter has a slightly different role than the previous context parameters. Its enumeration 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 in the *Subtopic* enumeration will be offered to the OpenInfobutton user.

In addition to *code* and *code system*, items in the Subtopic enumeration also need a *display name*. Any text can be typed as part of the display name, but you should choose a short and meaningful term for the user. The example below contains a Subtopic parameter with three enumeration items.

Subtopic*

☒ Search* ☐ Match*

Subtopic enumeration [Code System / Code / Display name] - Press TAB to add more

Code System*	Code*	Display Name*
MeSH	Administration & dosage	Dose
MeSH	Adverse effects	Side effects
MeSH	Contraindications	Contraindications

4.3.5 Non-HL7 compliant resources

When a resource is not compliant with HL7, a new set of fields is displayed once *search* is checked: *search using* and *syntax on resource*.

Gender

☒ **Search*** ☐ **Match*** Search using*: code

Syntax on resource:

Static value prefix	Parameter name*	Static value suffix
add @valuePrefix	gndr	add @valueSuffix

Age group

☒ **Search*** ☐ **Match*** Search using*: code

Syntax on resource:

Static value prefix	Parameter name*	Static value suffix
add @valuePrefix	age	add @valueSuffix

- Search using: indicates whether OpenInfobutton should use the parameter *code* or *display name* (text) when searching against the resource API. In the above example, both *gender* and *age group* should be used for searching using coded values.
- Syntax on resource: name of the parameter in the resource API. In the above example, the resource API gender parameter is called “gndr” and the age parameter is called “age”.
 - If needed, a static 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”.

5 Validating and testing

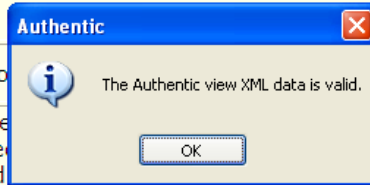
Once your resource profile is ready, you can validate it against its XML schema. This will verify whether all the profile required elements are filled out. To validate a profile, click on the XML->Validate menu or press the F8 key. If your profile is valid you should see a message like the one in the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



Open **i**nfobutton

Knowledge Resource Profile

Resource ID*	1		
Resource Name*	MedlinePlus		
Resource Purpose	Provides patient education		
Profile Description*	A freely available patient education content on diseases, medications, and medical procedures. The National Library of Medicine. Contains MedlinePLUS API is compliant with the HL7 Infobutton Standard		
Version Control*	Publication date*	08 / 01 / 2011	
	Profile author*	Name*	
		First name*	Last name*
		John	Doe
		Organization*	
		ID*	Name*
		1.3.6.1.4.1.3768	Veterans Health Organization
		Department*	
		Portland VA Medical Center	
		Job title*	
		Systems Analyst	
		Contact information*	



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

- Copy the profile XML file from c:\Program Files\OpenInfobutton Tailoring Tool\ to the c:\profiles folder. That's where OpenInfobutton reads the resource profiles from. If you are replacing an existing version of your profile with an updated one, make sure you back up the existing version first.
- 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 at [http://\[OpenInfobutton server name\]/infobutton-service/InfobuttonQA.htm](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](http://[OpenInfobutton server name]/infobutton-service/OpenInfobuttonDemo.html)