# Taxonomyserver documentation

Reinout van Rees
Stabu foundation, Delft university of technology
$Id: documentation.xml,v 1.4 2001/12/30 01:09:47 reinout Exp $

## Gentle introduction

The EU-funded eConstruct project [econstruct] has the aim of allowing the building and construction (BC) industry to communicate using the internet. In this modern day and age you need an XML data format. EConstruct has two, the taxonomy format XTD (XML Taxonomy Definition) for holding information about objects, properties, units and their definitions, translations and relations. Using the data in that taxonomy format, eConstruct generates a second format, used solely for communication about the actual doors, windows and heights.

Besides the XML data format(s) you need a small, central core of services to be able to do something with it. At a meeting between the project partners CSTB, Stabu and Delft university of technology it was agreed to aim for a two-part *eConstruct infrastructure*. On one hand a *taxonomy server* (TS) and on the other hand a *supplier catalogue server* (SCS). The former stores the taxonomy information (aforementioned objects, properties, units and their definitions, translations and relations) and the latter stores the catalogue data. Both can be queried for their information using a standardised interface.

To allow this work to gain some ground, you need open source [opensource] programs to demonstrate the usage and to allow easy adoption of the technology. It is conventional wisdom that you need accompanying open source software to "nail down the standard". That's why the eConstruct core is also called *eConstruct reference architecture*. So both the taxonomy server and the supplier catalogue server are released under an open source license (the Library GNU Public Licence [lgpl]).

Of course there is a visualisation layer on top of both servers, so the story is less easy than told here, but for the basic understanding it's more than adequate.

Note: this document only introduces to the taxonomy server, not to the supplier catalogue server!

## Technical architecture of the taxonomy server

### The taxonomy

In the eConstruct project, the taxonomy is made by Stabu [stabu] using the LexiCon program (which is an implementation of the ISO PAS 12006-2 [isopas]). The resulting taxonomy is converted using the lexiconconvertor program [lexiconconvertor] to the bcxml XTD format.

Currently the resulting taxonomy is copied into the source of the taxonomy server, followed by a recompile and re-package of all files. A future extention is to allow for external storage of the taxonomy, allowing for a plug-and-play exchange of an old taxonomy for a new one.

### SOAP interface

SOAP [soap] is the standard for exchanging XML files according to a standard interface. EConstruct defined such a standard interface for both the taxonomy server and the supplier catalogue server. This definitions can be found in the eConstruct document D201 [econstruct].

This document defines three functions to be fulfilled by the taxonomy server. The results are all in the XTD taxonomy format.

| | |
|---|---|
| searchNameInTaxonomy | search for a string that matches a name of an object in the language used for searching. |
| getTaxonomyItemHierarchy | Return a hierarchy of objects, both from the root to the currently selected object and branching out from the selected object to the tips of the hierarchy. |
| searchItemInTaxonomy | Returns all possible information about a single object. All properties, translations, etcetera. |

For each function there exists an XSLT stylesheet that selects the correct information from the XTD taxonomy file. A java class has been made that functions as a wrapper that calls the XSLT files. The java class contains functions that can be connected to the desired SOAP functions.

### HTML interface

To provide an HTML user interface on top of aforementioned functions, all that was needed were XSLT stylesheets that formatted the output of the three previous stylesheets to HTML. Combining these three HTML-outputs proved enough for providing an attractive user interface to the taxonomy server. A java class was written to tie all the stylesheets together.

The end result is a java servlet, plug-and-play-packaged as a java .war file that can be used in a java servlet web server such as Tomcat or Webspere.

### Summary

In conclusion it can be said that a simple combination of two times three XSLT stylesheets and two java classes provide a solid, well-functioning taxonomy server that is easily adaptable and extendable.

## Using the taxonomy server

### Obtaining the binary distribution

The latest binary distribution can be found at [sourceforge]. There you need to download the `taxonomyserver.war` and the `taxonomyprocessor.jar`.

### Installing

#### Installing tomcat

The instructions here might seem a bit long, but they are included in almost every serious XML application nowadays. It's a result of the rapid evolution of XML software compared to Tomcat. The upcoming tomcat 4.0 will probably reduce this list to one item.

- First install tomcat (3.2.3 is what I used)
- Download xalan 2 [xalan] (it is now also included in the lib-directory of the source code, so you can get it from there). The xalan distribution includes a xerces version.
- remove lib/jaxp (jaxp is included in the xalan/xerces combo mentioned above, the jaxp included in tomcat isn't good enough)
- copy the downloaded xalan.jar and xerces.jar to tomcat's lib directory. Then rename parser.jar to zparser.jar. Tomcat normally automatically loads all jars in the lib directory in an alphabetical order and we want xalan/xerces to be loaded first. (parser.jar is an older xml parser and doesn't play nice with the current standards, so the "z" in front of it puts it last in the classpath). Normally it'll take some battling with classpaths etcetera. I personally prefer *not* to set a system-wide classpath but to set it in batchfiles that start programs, that way you don't need one classpath that suites every program...
- on unix/linux, make the tomcat *.sh executable
- add TOMCAT_HOME and JAVA_HOME variables to the bin/tomcat.sh (or .bat)
- set the port (8080 is the default) you want tomcat to operate on in conf/server.xml

### Installation of the taxonomyserver

Once tomcat is prepared correctly, the installation of the taxonomyserver is quite easy.

- put the taxonomyprocessor.jar in the tomcat/lib directory (it contains the functions needed by soap)
- enter the correct soap parameters in your local soap-admin (following the values on the econstruction.citg.tudelft.nl:8080 site) [these two steps should ensure a 100% functional soap taxonomy server!]
- to get the html-interface running, also install taxonomyserver.war in your tomcat deploy directory and restart tomcat. /taxonomyserver should now give you a reasonably usable taxonomy interface.

## Using the web interface

Now either your local computer or the demo site at http://econstruction.citg.tudelft.nl/taxonomyserver [http://econstruction.citg.tudelft.nl/taxonomyserver] should give you a html interface to the taxonomy contents. The opening screen allows you to search for a term in a certain language (like "door" in English or "porta" in Greek). Try one of these and press **enter**.

The second screen shows a list of matches, select one and continue.

The thirth (and most important) screen shows a hierarchy on the left hand side and the properties of the currently selected object on the right hand side. The hierarchy can be used to select either more generic or more specialised objects, according to preference. Once the best fit has been found, the appropriate values to search for can be filled in at the right hand side of he screen.

Pressing **enter** generates the bcxml query and shows a list of catalogues. Select one to search in that catalogue. (Which passes you into the realm of the supplier catalogue server).

## Obtaining the source code

The source code can be obtained using CVS from sourceforge [http://sourceforge.net/projects/bcxml], module name "bcxml".

# Bibliography

[econstruct] Michel Bohms. *econstruct web-site* . On-line at econstruct.org [http://econstruct.org].

[isopas] *ISO PAS 12006-3* . On-line at icis.org [http://www.icis.org/tc59sc13wg6/] .

[lexiconconvertor] *lexiconconvertor program*. Reinout van Rees. On-line at bcxml.sourceforge.net [http://bcxml.sourceforge.net/progs/lexiconconvertor/] .

[lgpl] *Library GNU Public License*. Richard Stallman. On-line at gnu.org [http://www.gnu.org/copyleft/lesser.html] .

[opensource] *open source web-site* . On-line at opensource.org [http://www.opensource.org].

[sourceforge] *sourceforge download site*. On-line at sourceforge.net [http://sourceforge.net/project/showfiles.php?group_id=20109].

[soap] *SOAP simple object access protocol specification*. On-line at w3.org [http://www.w3.org/TR/SOAP/].

[stabu] *Stabu foundation web-site* . On-line at stabu.nl [http://www.stabu.nl].

[unittest] *unit tests*. J. Donovan Wells. On-line at extremeprogramming.org [http://www.extremeprogramming.org/rules/testfirst.html] .

[xalan] *Apache xalan xslt processor*. On-line at xml.apache.org [http://xml.apache.org/xalan] .