UNE Migration Utility

Introduction

The University of New England, Australia, is in a period of transition from WebCT CE 4.1 to Sakai. We needed a solution to migrate the content, quizzes and resources from a WebCT CE 4.1 course to a Sakai site. At present, the export of a content module or an entire course from WebCT CE 4.1 does not transfer across to Sakai becuase its a custom IMS package developed by WebCT (the term 'vendor lock-in' comes to mind!).

After some investigation, we started looking at the Rutgers WebCT Migration utility. We tried it out but found some limitations and dependencies that we could not use. So we decided to re-work the existing tool and add additional functionality.

When we developed this utility, we always kept the Sakai community in mind as we wanted to make this utility generic enough that anyone could download and run it straight away without having to modify source code and fill all the dependencies.

What it is and what it does

This utility takes an exported course from WebCT CE 4.1 and separates out the content into individual packages (content modules, quizzes, resources from My-Files). These packages are then able to be imported into a Sakai site.

- The modules are converted to the IMS CP 1.1.4 format and are able to be imported directly into Melete
- The quizzes are converted to a full UTF-8 IMS-QTI format and are able to be imported directly into Tests & Quizzes (Samigo) as a quiz and also as question banks.
- The files and folders that were in My-Files in a WebCT course, are packaged up and can be uploaded to the Resources tool of a Sakai site, or stored on a user's computer.

We have written command-line scripts to automate the export and transfer of the exported courses and associated files to the machine that will be performing the conversion. This can either be run on a per course basis, or by reading a text file to get a list of courses to export.

We have written a command-line script that accepts arguments and passes them to the conversion utility and exported the converted packages.

We have also written a JSP page that allows a user to browse for the exported courses, select what they want to export (modules, or quizzes or resources, or combinations thereof), and then run the converter. When done, the converted packages are able to be downloaded.
This JSP is not essential, we just made it for convenience. The command-line script does all the work.

Contacts

- Steve Swinsburg
  Web Developer, University of New England, Australia

- Ben Funnell (bfunnel@une.edu.au)
  Web Developer, University of New England, Australia

New Features

- Flexible and highly configurable commandline and JSP wrappers
- Export of quizzes in full UTF-8 format. Previously the XML file encoding was UTF-8 but the character encoding was not, we modified this so the utility exports the file in a true UTF-8 file, so more characters are now retained (ie German characters for language courses).
- Export of all resources that were held in My-Files within WebCT as well, so people have easy access to the raw files as well as the content modules.

Download the utility and associated helper scripts

✓ **Current version**

27 September 2007 - download now

How to export courses from WebCT CE 4.1

Included with WebCt CE 4.1 is a Perl script called cp_api.pl. This script lives in the IMS directory under the admin section of the server (generic/ims/).

This script generates an exported course package at (generic/ims/exported_packages/)

Because we wanted to export hundreds of courses at once, we wrote a short script to read in a list of WebCT courseid's from a file, run the exporter, and then copy them to a different machine. Because it automatically copies, we setup ssh keys to remove the need to login to the other machine all the time.

The script is located in webct_exporter/bulk_export.sh

To use the bulk_export.sh script:

ON THE WEBCT SERVER:

1. Generate an SSH key
ssh-keygen -t rsa

ON THE DESTINATION MACHINE:

2. Add the public key you just generated to the list in ~/.ssh/authorized_keys2

ON THE WEBCT SERVER:

3. Configure bulk_export.sh
There are variables at the top of the script like paths to the user/ directory on the WebCT server and the machine IP and username that the files should be copied to.

ON THE WEBCT SERVER:

4. Run the export script. It accepts one commandline argument - the file containing newline separated courses to convert. An example of this file is given. Failures are collected in a logfile.
./bulk_export.sh courses_to_convert

To export a course manually:

Navigate to the IMS directory on your WebCT server.
Run ./cp_api.pl -action=export -compress=yes -course=$course
where $course is the full WebCT courseid, ie comp100_07s1 (which is what we use)

This will create a zip file in the exported_packages/ dir and will look like this:
comp100_07s1_COURSE_20070906104314.zip

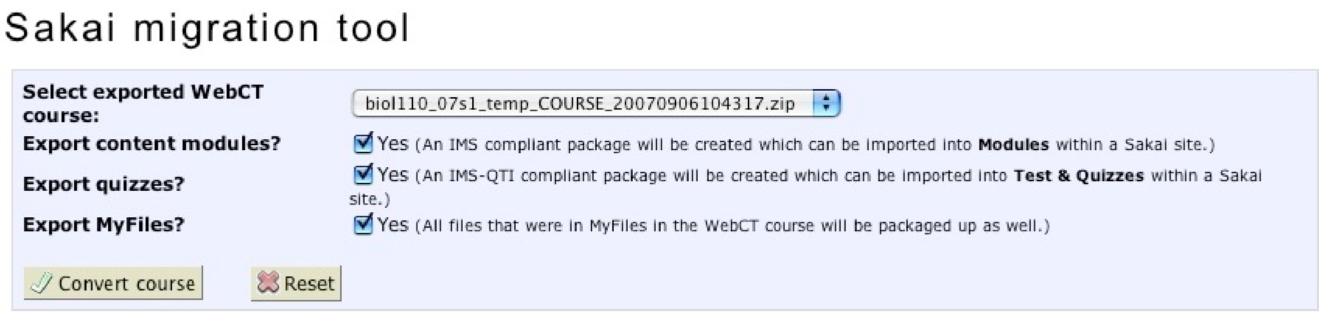
Depending on how you are going to run the course converter, you will need to copy this zip file to your computer as well as the corresponding user/ directory for that course which contains all the uploaded files for that WebCT course.

Then configure the Converter as above and run it on the exported package.

How to use the converter

Please see the included README for information about configuring the utility. We moved all of the configuration options into a Java class called Config.java which is read by default, but overridden if options are specified.

from a JSP page:



Sakai migration tool

Select exported WebCT course: biol110_07s1_temp_COURSE_20070906104317.zip

Export content modules? ☑ Yes (An IMS compliant package will be created which can be imported into **Modules** within a Sakai site.)

Export quizzes? ☑ Yes (An IMS-QTI compliant package will be created which can be imported into **Test & Quizzes** within a Sakai site.)

Export MyFiles? ☑ Yes (All files that were in MyFiles in the WebCT course will be packaged up as well.)

how to use the utility from a JSP page

```
<!-- A web form like the one in the screen grab above would capture what course
 you wanted to convert and also any associated options. These would then be processed
 like below and then the converter is called. -->

<!--setup flags for the conversion command -->
<c:if test="${export_modules == 1 && export_quizzes == 0 && export_files == 0}">
        <c:set var="export_options" value="--melete" scope="application" />
</c:if>

...

<c:if test="${export_modules == 1 && export_quizzes == 1 && export_files == 0}">
        <c:set var="export_options" value="--melete --samigo" scope="application" />
</c:if>

...

<c:if test="${export_modules == 1 && export_quizzes == 1 && export_files == 1}">
        <c:set var="export_options" value="--all" scope="application" />
</c:if>

<%

String command = "java -cp " + application.getAttribute("converter_location").toString() +
                " Converter" +
                " --export=" + application.getAttribute("export_path").toString() +
                " --destination=" + application.getAttribute("conversion_output_dir").toString() +
                " " + application.getAttribute("export_options").toString() +
                " --classpath=" + application.getAttribute("converter_location").toString() +
                " --webcthost=" + application.getAttribute("webct_host").toString() +
                " --webctdir=" + application.getAttribute("webct_user_dir").toString() +
                " --tmp=" + application.getAttribute("webct_tmp_dir").toString();

out.print("command:" + command + "<br/>");
Process p =        Runtime.getRuntime().exec(command);
int ret=p.waitFor();

if(ret != 0) {
        pageContext.setAttribute("proceed_ok", false);
        pageContext.setAttribute("ret", ret);
}
out.print("ret:" + ret + "<br/>");

%>

<c:if test="${proceed_ok == false}">
        <div class="portlet-msg-error">
                <h3>Conversion process failed</h3>
                <p>The converter returned status code <strong><c:out value="${ret}" /></strong> and could not
complete successfully. Please contact the system administrator.</p>
                <p><input type="button" value="Back" onclick="javascript:history.back(); return false;" /></p>
        </div>
</c:if>
```

from the commandline

how to use the utility from the commandline

```
$ java -cp bin Converter

The following options are supported:
         --export=/full/path/to/export.zip
    [ --destination=/full/path/to/destination/ ]
    [ --classpath=/full/path/to/classes/and/scripts/ ]
    [ --webcthost=webct host with NO PROTOCOL eg online.une.eu.au ]
    [ --webctdir=/full/path/to/webct/user ]
    [ --tmp=/full/path/to/temp/working/directory/ ]
    [ --modules ] (convert only modules)
    [ --quizzes ] (convert only quizzes)
    [ --userfiles ] (collect webct user files)
    [ --all ]  (convert both melete and samigo if available (default))

Trailing slashes are required.
```

from another Java class:

how to use the utility from another Java class

```
String exportZip="/Users/bfunnel/dev/converter/exports/educ293_07s2_COURSE_20070829131046.zip";
String destination="/Users/bfunnel/dev/converter/complete/";

Converter c = new Converter();
c.setExportZip(exportZip);
c.setDestination(destination);
c.unZip();

if(c.convertAll()==0) {
        c.moveMeleteZip();
        c.moveSamigoZip();
        System.out.println("Completed successfully");
} else
        System.out.println("Error");

c.deleteTmp();
```

What do do after the conversion process is complete

Modules.zip

The modules.zip file should import directly into Melete lesson builder. We did experience the occasional error and are not sure why. Usually we just reconverted the package using the webct2melete script instead and it went ok.

Quizzes.zip

The quizzes.zip file that the xporter produces contains a bunch of XML files in IMS-QTI compliant format. If there are any associated resources with a quiz, they will be in a folder called 'protected' within this archive.
Generally there is 1 XML file per quiz from WebCT. A legacy from the Rutgers code is that there is a 100 question limit, so once this limit is reached
a new file is created. Not many quizzes have more than 100 questions though, right!

You can import these XML files directly into Samigo to recreate the quizzes. You can also create question pools out of the same files. For any related resources like images, we uploaded these to the Resources tool in a Sakai site (the folder could be marked as private so students cannot directly access these resources). The questions then need to be looked at to fix up the image paths.

Userfiles.zip

There is also a package created that contains all of the files that were in My-Files in the WebCT course. These may be useful to an Instructor so we just uploaded the zip file to the Resources tool for the site and left it there for the instructor. If the instructor wants to use it, they will need to download the file and unzip it (there is no unzip utility in Resources).

Open Issues and Enhancement Requests

- Modify the tool so that it also converts Course Genie IMS packages. We use Course Genie for some courses but they cannot be imported into Melete. If we could get the utility to convert these packages instead of having to put them into WeBCT first, then export, then convert, that would save a lot of time (as well as removing WebCT as a dependency in future) 13/09/07 Steve Swinsburg
- Write web services to allow the automatic upload and import of the converted packages directly into Melete and Samigo. 05/09/07 Steve Swinsburg
- Someone write code to re-link the resources within a quiz to a path in the resources tool in a Sakai site. This link currently still points to a path in WebCT and needs to be done manually 27/09/07 Steve Swinsburg