



Project Information			
Project Acronym	DEC		
Project Title	Distributing Eportfolio Content		
Start Date	01.02.2010	End Date	02.07.2010
Lead Institution	Pebble Learning Ltd.		
Project Director	Shane Sutherland		
Project Manager & contact details	Matthew Wheeler, ePortfolio Consultant, Pebble Learning Ltd. University of Wolverhampton, Shifnal Road, Telford, TF2 9FT.		
Partner Institutions	n/a		
Project Web URL	http://wiki.cetis.ac.uk/PIOP_PebblePad		
Programme Name (and number)	PIOP3		
Programme Manager	Lisa Gray		

Document Name			
Document Title	Final Project Report		
Reporting Period			
Author(s) & project role	Matthew Wheeler, Project Manager		
Date		Filename	DEC_Final_Report.docx
URL	<i>if document is posted on project web site</i>		
Access	√ General dissemination		

Document History		
Version	Date	Comments
V1.0	28.07.10	

Distributing Eportfolio Content (DEC)

Final Project Report

1. Project Description

PebblePad have continued to extend the use of the Leap2a specification through the further development of an open source Moodle Block, which facilitates the export of Leap2a zip-packages from Moodle v1.9. This enables any Moodle users to export Moodle items as Leap2a assets and import them into other compatible programmes or tools, for example PebblePad or Mahara.

The enhanced Moodle block enables the following items to be exported in the Leap2 format as a downloadable file:

- Any file previously uploaded into Moodle
- Moodle blog contributions
- Moodle forums
- Resources (for example course materials uploaded by tutors)
- Assignments (including the task, submission and feedback)

2. Organisational and technical issues encountered during the project

Implementing OAuth

The project team at Pebble Learning Ltd. needed to implement a new version of Open Authorisation (OAuth) across the whole PebblePad system as part of the DEC (Distributing Eportfolio Content) project. The project team had previously implemented a 3-legged OAuth variant in previous versions of PebblePad.

Implementing 2-legged OAuth in a portfolio system like PebblePad requires more planning of the access controls than is required for the regular 3-legged variant. In 3-legged OAuth the user is involved in the authorisation process and grants a single application access to parts of their portfolio which they decide, in the 2-legged version the consumer (or receiving system) is trusted to access the data of any user in the portfolio system without explicitly requesting permission from the individual users.

For the PebblePad implementation of 2-legged OAuth we chose to restrict full read access to the portfolio to 3-legged OAuth only. Trusted consumers can access basic overview information and create new assets within a user's store but it cannot read existing assets without the user granting access through the regular 3-legged OAuth process. This allows for common uses in external tools, such as the Moodle block, for displaying notifications and sending reports to the portfolio without any external setup by the user. However, it does not cover cases where an external system needs to aggregate assets in users' portfolios with minimal user interaction.

We choose to address this problem by implementing a new option that allows a trusted consumer to read a limited set of assets chosen by users. This appears to the user as a send to <consumer> option, which in turn grants the consumer permission to read the asset through PebblePad's Leap2a feeds. This then in turn allows the <consumer> to read from any user's portfolio without any additional setup, whilst protecting the private data stored in the users portfolio.

There are limited training courses available on such technologies, and the project budget did not enable the project team to attend a course, instead, the team researched at depth OAuth and its variants by engaging in forums, literature and through basic trial and error. It is the PebblePad way to involve all the company's developers in the process of knowledge transfer, we achieve this through project work and internal training to share any new knowledge gained.

Exporting Assets from Moodle

There were some technical challenges incurred when trying to map contextualised Moodle information to their entries within the Leap2a structure. For example, Scorm quizzes were especially difficult and in most cases all tests were only able to generate a summary of the quiz taken. This is due to the method of storing Quizzes within Moodle would differ depending on the originating author of the quiz. The project team tested numerous different examples, only being successful with SCORM 1.3. The other SCORM versions always provided a summary, however, they did not always contain the data within the quiz.

Blogs within Moodle mapped especially well to the Leap2a specification with each blog post becoming a separate entry. An option was added in the Moodle block to link all blog entries to a new blog, thus enabling the relationships to remain intact. Forums are also exported as an entry with any comments attached.

All assignments and resources are reproduced as inline HTML and wrapped with the Leap2a inline entry; this has produced good results throughout testing when presenting the asset inside PebblePad.

3. Deliverables

	<i>Status</i>	<i>Comment</i>
Project website live (http://wiki.cetis.ac.uk/PIOP_PebblePad)	√	
Initial project meeting (23.02.2010 Derby)	√	
PIOP technical meeting (09.03.2010 Nottingham)	√	
OAuth Authorisation Layer developed	√	See section 4
Moodle/PP Interface designed	√	
LEAP2A export function developed	√	
OAuth Authorisation Guidance Notes/Report completed	√	See section 4
Local PebblePad testing	√	
Community testing	√	

4. OAuth Authorisation Libraries

DotNetOpenAuth Library

DotNetOpenAuth Library provides a complete solution for basic consumer and provider OAuth implementations, by allowing OpenId support. The DotNetOpenAuth Library abstracts most of the details from the application at the expense of flexibility. In particular this library does not currently support 2-legged OAuth.

Besides the lack of flexibility, the primary problems with this library are the relative complexity of the code-base, which makes it difficult to modify and debug, together with a lack of implementation documentation. Excellent logging support means that it is rarely necessary to investigate the code when troubleshooting, but lack of documentation is an issue with only the samples serving as a guide.

Adding OAuth support using the library requires implementations of four interfaces, which are primarily related to persistent storage of tokens by the application.

- `IServiceProviderTokenManager`
 - This is called on by the library to store and retrieve tokens, this is normally a wrapper around the storage mechanism used by the application to persist and retrieve data from persistent storage. The example code provides an example, which leverages Linq2Sql for object persistence.
 - Earlier versions of the library relied on use of an ORM (Object Relational Mapper) to automatically persist some changes to token details, such as the verification code. This appears to have been rectified in the current version with the inclusion of an update method in the token manager interface to handle information that is assigned after the initial creation of the token.
- `IConsumerDescription`
 - Describes an OAuth consumer, this class can mostly be lifted from the example code in the library and tweaked to suite the application data access requirements.
- `IServiceProviderRequestToken` and `IServiceProviderAccessToken`
 - These represent OAuth tokens for unauthorised (request) and authorised (access) tokens, a single class can be used for both since they share most of their properties.

Primary API access is through the `ServiceProvider` class, which must be instantiated with a concrete instance implementing `IServiceProviderTokenManager`.

The OAuth authorisation flow is achieved by calling on the service provider to read the OAuth parameters from the request, which are returned as a different type of request token depending on the user's stage in the process. This token is generally passed back to a method in the service provider to move onto the next step in the flow, the `OAuth.ashx` file in the provider, as an example this can be used unaltered to handle the flow.

The final requirement is a page to handle authorisation of any OAuth requests. This shows details of the requesting application and option to allow the user to allow or deny the request. The `Authorize.aspx` from the provider can be used as the basis for this, but will require modification to protect access so that the user is forced to login when accessing the page.

DevDefined OAuth Library

The DevDefined OAuth library provides a more focused and flexible implementation, which is aimed only at OAuth and abstracts less of the details away from the originating application.

The code is smaller and easier to modify than DotNetOpenAuth but logging is non-existent and left entirely to the application. Documentation is sufficient in combination with the samples and provides enough guidance to start implementation.

Flexibility on the consumer side in particular is much better than DotNetOpenAuth library, it provides a fluent interface through `OAuthSession` to construct and make requests. It also provides methods to allow access to the OAuth parameters by calling `GetRequestDescription` on a request, this allows usage of proxy classes such as the ASP.NET webservice interfaces by injecting the parameters.

Creating a provider using the library only requires implementations of three interfaces, which are used to store and retrieve tokens, consumers and nonces, which are used to prevent replay attacks.

- `ITokenStore`
- `IConsumerStore`
- `INonceStore`

Access to provider functions is available through the `OAuthProvider` class, this is used with an OAuth context created by `OAuthContextBuilder` to grant and authorise tokens. The example provider has samples for the authorisation and request pages, which can be used as a basis for a production provider.

5. Scenarios of practice

There is a range of contexts where a Moodle based Leap2a export has proved to be beneficial to users, for example:

- A Learner transferring assets from an FE institution (where Moodle is a widely used system) into a HE context where they may first encounter an ePortfolio;
- Teachers transferring assets from a college CPD course in Moodle to REfLECT (the bespoke version of PebblePad used by the Institute for Learning) or their installation of PebblePad for internal appraisal purposes;

As part of the DEC project we worked closely with some of our customers to fully test the utility of the Leap2a export in the following scenarios:

1. A graduating student was leaving their Further Education institution for higher studies at University, where the new institution is currently using PebblePad. The student was able to manage their work and evidence in Moodle and transfer these items out of Moodle using the Moodle block's Leap2a export and successfully import these items into their new PebblePad account.

2. A teacher working in the Learning and Skills Sector has been using Moodle and REfLECT contemporaneously. Evidence gained of their CPD on a Moodle course needed to be presented on REfLECT for their official end of year CPD return to the IfL. The individual was easily able to export their evidence and quickly upload it into REfLECT, and their organisations installation of PebblePad, as the Moodle block can be directed to numerous installations at once. This means that evidence can be gathered once and distributed to multiple locations preventing multiple data-entry.

6. Additional deliverables as agreed with individual mini-projects

Some of the more experienced projects in the PIOP community met in Nottingham to discuss shared objectives across the various projects. Although this cannot be truly classed as training it did provide the opportunity to discuss and agree on shared methods, which included an agreed method of web-service discovery. The outcome was a suggestion to use a centralised database that stores the information about each application and service. For the timeline of this project Newcastle have agreed to host the discovery service.

7. Specification Considerations

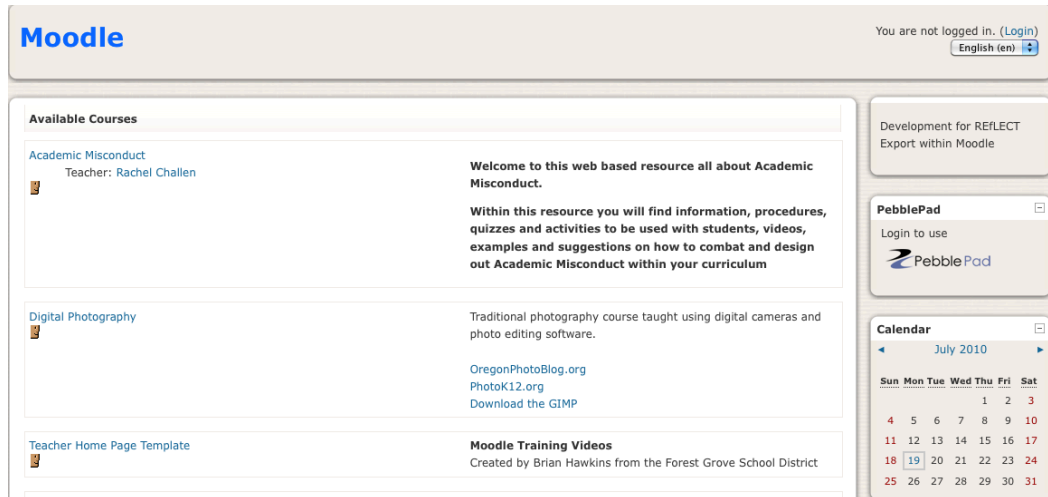
Moving to the 2010-07 version of the Leap2a specification has introduced a significant backwards compatibility issue with the initial 2009-03 version. As part of the move to the leapspecs.org site all the namespaces used to identify the elements used by the specification have now changed. This means that although the specification is largely unchanged, older importers will be unable to recognise any of the Leap2a elements, even those that have not changed.

This should be a one-off process with relatively short-term impact this time, but as the adoption of the Leap2a standard increases, even minor changes could introduce significant issues. Backwards compatibility will have to be considered more carefully in future revisions to minimise issues between portfolios using different versions of the specification.

Appendix

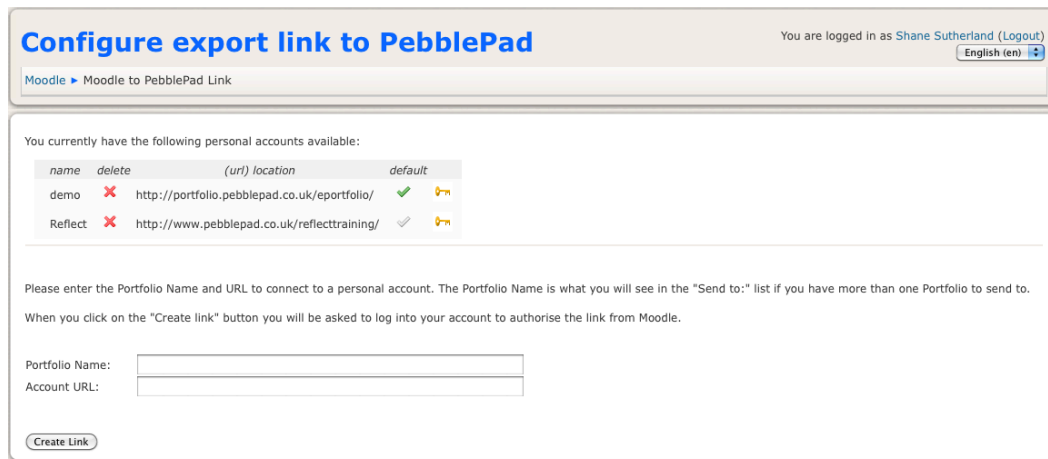
PebblePad Moodle Block is available to download from the following URL:
http://www.pebblepad.co.uk/cs_documentation/PebblePad-Moodle_Block-V1.1.zip

Moodle Homepage with PebblePad Moodle Block



The screenshot shows the Moodle homepage. At the top left is the Moodle logo. At the top right, it says "You are not logged in. (Login)" and "English (en)". The main content area is divided into two columns. The left column has a section titled "Available Courses" with three course listings: "Academic Misconduct" by Rachel Challen, "Digital Photography", and "Teacher Home Page Template". The right column has a "Development for REFLECT" block, a "PebblePad" block with a login button and the PebblePad logo, and a "Calendar" block showing July 2010.

Moodle Block Settings

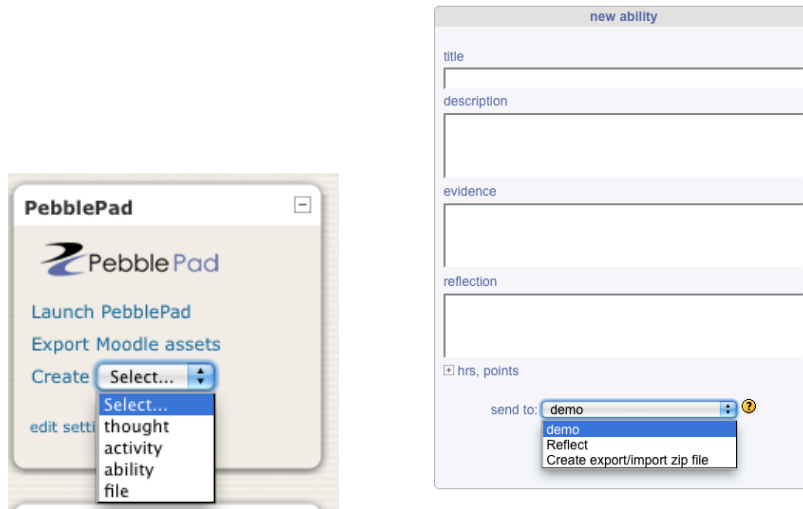


The screenshot shows the "Configure export link to PebblePad" settings page. At the top, it says "You are logged in as Shane Sutherland (Logout)" and "English (en)". Below the title, there is a breadcrumb "Moodle > Moodle to PebblePad Link". The main content area starts with the text "You currently have the following personal accounts available:" followed by a table:

name	delete	(url) location	default
demo	X	http://portfolio.pebblepad.co.uk/eportfolio/	✓
Reflect	X	http://www.pebblepad.co.uk/reflecttraining/	✓

Below the table, there is a paragraph of instructions: "Please enter the Portfolio Name and URL to connect to a personal account. The Portfolio Name is what you will see in the 'Send to:' list if you have more than one Portfolio to send to. When you click on the 'Create link' button you will be asked to log into your account to authorise the link from Moodle." Below this are two input fields: "Portfolio Name:" and "Account URL:". At the bottom left is a "Create Link" button.

Creating PebblePad Assets directly from the Moodle Block



Exporting Moodle Assets from the Moodle Block



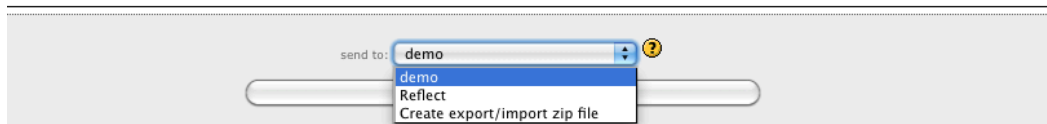
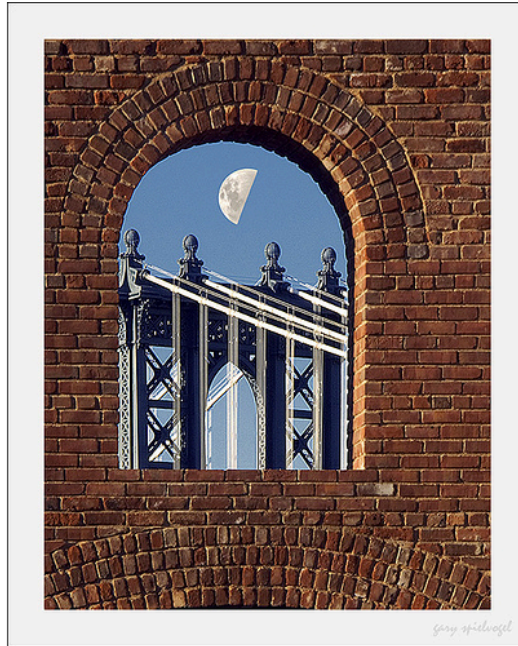
Here's just one more example from [Gary's Flickr stream](#). When you see a window, be sure to look through it with a photographer's eye

towards composition.

Can you tell in this photo that the moon was added? You could put almost anything behind this wall and get an interesting photograph.

Explore on your own:

Flickr [Framing Tags](#)



Other Useful Links

OAuth Specification Website (<http://tools.ietf.org/html/rfc5849>)

Overview and beginners guide (<http://oauth.net/documentation/getting-started/>)

Information on Google's implementation
(<http://code.google.com/apis/accounts/docs/OAuth.html>)

Documentation for the DevDefined OAuth library
(<http://code.google.com/p/devdefined-tools/wiki/OAuth>)

Home page of DotNetOpenAuth library (<http://www.dotnetopenauth.net/>)

Details about OpenSocial signed requests (2-legged OAuth)
<http://sites.google.com/site/oauthgoog/2leggedoauth/2opensocialrestapi>
http://wiki.opensocial.org/index.php?title=Introduction_To_Signed_Requests
http://wiki.opensocial.org/index.php?title=Validating_Signed_Requests

Moodle Beginners Guide
http://docs.moodle.org/en/Development:Finding_your_way_into_the_Moodle_code

Moodle Developer Course
<http://dev.moodle.org/>

Moodle Development Block
<http://docs.moodle.org/en/Development:Blocks>

General Moodle Integration
http://docs.moodle.org/en/Integration_FAQ