



SCORM Learning Management System Authoring Template  
v1.01

## User Guide

Copyright 2016 CoursePlay Elearning Solutions Ltd  
Published & Distributed by Digital Workshop



## Introduction

CoursePlay is designed to provide a simple, but fully customisable, learning management system. Centred around HTML5 launch pads which can be redesigned and extended in the WYSIWYG editor in Opus Pro, it allows developers and instructional designers to create their own very flexible, modular elearning management system to suit their precise requirements.

Although designed for use with Opus Pro, the LMS portion of CoursePlay is able to use SCORM v1.2-conformant material from any authoring tool.

Unlike many LMS systems it provides a graphical interface to make the learner's experience much more engaging and can be redesigned easily to suit any house style

## Key Features

### Design Your Course Interface Easily

CoursePlay provides the course LaunchPads as Opus Pro templates which means you can design the course interface to look the way you want it easily, add content and organise the material that you provide to the customer in a visual WYSIWYG manner.

### Upload Any SCORM Content

The CoursePlay Uploader lets you upload any SCORM v1.2 material quickly and easily and that includes any SCORM material produced in Opus.

### Reports Standard SCORM Data

The LMS portion of CoursePlay can store any core SCORM v1.2 data models you require with the basic information already built in.

### Two Step User Registration

CoursePlay will handle new user registration and will email the registered user with a username and password.

### Different User Levels

CoursePlay allows different user levels with different access rights including Standard and Administrator by default.

### Includes Database and PHP

In addition to the Opus Pro files for your Courses and your LMS interface, the CoursePlay package includes everything you need to install and run the server portion of your LMS including the database and all the PHP code to run the LMS. You need a little technical knowledge to get your LMS up and running.

### Design Interface and Content in Opus Pro

All the content creation and administration elements, colour schemes, page layouts, thumbnails and button styles are customisable in Opus Pro just like any other object. You can add content before, after or on the CoursePlay Launchpads and copy them any number of times.

### **Fully Extendable To Do Whatever You Need**

The CoursePlay LMS uses PHP and MySQL and can be easily extended to do whatever you need. If you are not a developer, you can always ask one of our team to extend the functionality for you at very cost-effective rates under Digital Workshop's [ProSolve](#) service.

## Overview

There are a number of components provided with the CoursePlay package which are divided into elements for *content*, *administration* and *core functionality*.

The core content files are provided as Opus Pro .imp files in the `_templates` directory and can be edited as required before being published to HTML5, renamed as `index.htm` and uploaded to your CoursePlay installation.

You will probably have two installations of CoursePlay – one on your local machine which is for development purposes which will include all the content of the distributed .zip file including documentation and templates and the server-side installation which will be the edited CoursePlay files and your course content, which you will make available to your learners and which is registered using the process described later in this manual.

## CoursePlay Content Elements

### Courses menu

This is provided as *course-menu.imp* in the *pads-courseinterfaces* subdirectory of *\_templates*. You will need to open this in Opus Pro and redesign it as required. Changing the Course titles and thumbnails to reflect your content.

You will also need to edit the actions for each of the course launch frames on the menu page once you have uploaded courses to the online LMS (see below). Once edited you will need to publish, rename the resulting *.htm* to *index.htm* and copy the files and all their resources to the *pads/courseinterfaces* subdirectory of the CoursePlay directory.

### SCORM Courses

We have provided sample templates for creating the content of your courses as Opus Pro .imp files. There are two templates one for standard content provided as *scormbasic.imp* in the *packages-scormbasics* subdirectory of the *\_templates* directory; and the other for video content provided as *scormvideo.imp* in the *packages-scormvideo* subdirectory of the *\_templates* directory.

You can edit these in Opus Pro as with any other publication and then publish as a SCORM *package* ready to upload to your LMS. You must use a published version of the Package Uploader tool to upload your course packages, or to register a course package uploaded separately via FTP. The Package Uploader is provided as an option on the Administrator's menu.

We have provided a guide to packaging your SCORM course in the Opus SCORM packager. This is available as *CoursePlay SCORM Content Package Guide.pdf* in the *\_documentation* subdirectory of your developer installation of CoursePlay.

**Note that all course content goes into the *packages* subdirectory of your CoursePlay installation.**

Once your copy of CoursePlay has been installed to the web server you can then use the Uploader interface to upload these SCORM packages and allocate them to one of the items on your courses menu, as explained in the *Administrator Guide*

## CoursePlay Administration Elements

### User Login

This is the dialog used by Learners to access your LMS and will be seen by your users so should be consistent with the course menu design.

This publication is provided as *loginpad.imp* in the *pads-default* subdirectory of the *\_templates* directory. Open this in Opus Pro and edit to suit your house design but do not change any actions or scripts. Publish as a stand-alone webpage, rename the resulting *loginpad.htm* to *index.htm* and copy all the files into the *pads-defaults* subdirectory of the CoursePlay directory.

### Package Uploader

This is the dialog used by Administrators to upload courses to your CoursePlay installation. If you are reselling CoursePlay you may wish to edit this to match the customer's design but if you are simply using this to maintain your own LMS there is no need as the end users should never see it.

This publication is provided as *uploader.imp* in the *pads-default* subdirectory of the *\_templates* directory. Open this in Opus Pro and edit to suit your house design but do not change any actions or scripts. Publish as a stand-alone webpage, rename the *uploader.htm* to *index.htm* and copy all the files to the *uploader* subdirectory of the CoursePlay directory.

### User Manager

This is the dialog used by Administrators to manage the users of your CoursePlay installation. If you are reselling CoursePlay you may wish to edit this to match the customer's design but if you are simply using this to maintain your own LMS there is no need as the end users should never see it.

This publication is provided as *usermanager.imp* in the *pads-usermanager* subdirectory of the *\_templates* directory. Open this in Opus Pro and edit to suit your house design but do not change any actions or scripts. Publish as a stand-alone webpage, rename the *usermanager.htm* to *index.htm* and copy all the files to the *pads/usermanager* subdirectory of the CoursePlay directory.

## Core Functionality

These are the files required to build, run and manage the LMS database and handle the communication between all the various elements. These files are run on your web server and should not be edited.

For CoursePlay to work you will need access to a webserver which is running an up-to-date version of PHP and MySQL. Both of these usually come provided with, or can be installed on, any web-hosting solution.

### CoursePlay

This directory includes all the content of your CoursePlay some of which you will populate with your own versions of files as outlined above.

Once you have edited all the files you need you can simply upload the whole CoursePlay directory to the required location on your webserver. You must ensure the directory structure is maintained.

## Database

CoursePlay will record the information to a MySQL database which should be set up on your webserver. You will need to know the location and admin password for your database to configure the login process for CoursePlay. Again these are provided by your webhosting provider via their User Control Panel.

There are also a number of different front ends you can use to access your MySQL installation, including MySQL Workbench

We have provided an *.sql* file which you can import into your MySQL installation and will automatically create all the tables you need.

This file is provided as *courseplay.sql* in the *\_database* directory and can be copied to an appropriate location on your webserver where you can access it from the MySQL interface.

## Getting Started

*You can also follow the process of installing and setting up CoursePlay is detailed in Introductory video tutorials provided via the CoursePlay page on the Digital Workshop website.*

## Overview

To run courses in CoursePlay you will need to connect the course menu buttons to a course installed in the database using the CoursePlay Uploader. This will provide an identification reference called the SCOID to which the course button can be connected.

Therefore an initial installation of CoursePlay must be set up before you can connect the courses to run.

## Initial Installation

Simply upload the CoursePlay directory and all its subdirectories to the chosen location on your webserver. You may find it is easiest to zip CoursePlay into an archive and unpack this on your server so that all directories and subdirectories are correctly created and populated with the relevant files.

Make sure the URL you are using for your LMS is set up to allow php scripts to run and that the default document is *index.php*, as CoursePlay needs to run *index.php* by default. There should be no other index files (e.g: *index.html*) in the root directory for your CoursePlay domain.

### Create Database

Before setting up CoursePlay itself you need to create and setup the database where you will store the information. We recommend using industry-standard MySQL available free of charge from Oracle from [www.mysql.com](http://www.mysql.com).

If your server is not already supplied with MySQL you will need to install it and you will need to install an Administration program such as phpMyAdmin or MySQL Workbench.

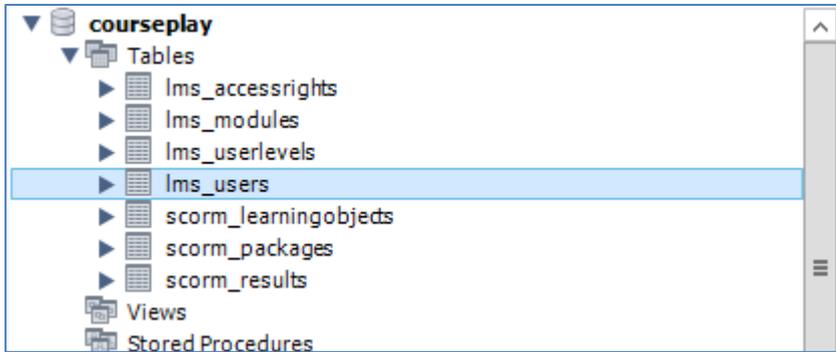
The MySQL installation will set up a localhost connection and you will need to know the details of this connection – including the database name, username and password for the connection. You will need these in order to add them to your *login.php* as we will see later.

In your MySQL administration program use the Import facility to import *CoursePlay.sql* from wherever you have uploaded it. This will create an empty database to hold your CoursePlay information (the default database name we use is *CoursePlay* but you can use another if you prefer but will need to change the *CoursePlay.sql* file as explained below).

Then upload the *CoursePlay.sql* to an appropriate location on your webserver and import that file into your blank database. If you get an error message saying that no database has been selected then you should open the *.sql* file in a text editor such as Notepad++ and modify the line `USE 'CoursePlay'` and replace *CoursePlay* with the name you gave your

database. In some MySQL admin software such as phpMyAdmin you may need to remove this line altogether as you select the database prior to importing the sql file.

This will populate the database with all the tables you need and populate some of those tables with key entries which the CoursePlay setup needs. If you refresh the view of your database you should find a list of tables looking like the image below.



The instructions as to how to do all of the above will be provided with whichever front-end you use to administer your MySQL service. View the online tutorials if you require a step by step guide or contact [support@digitalworkshop.com](mailto:support@digitalworkshop.com) to order our remote setup service. Note that the columns of the table may have a Not Null setting by default and you should deselect all but those shown in the diagram as only these need to be Not Null.

This will create all the database tables and queries required for CoursePlay to work.

Then register a user for that database which CoursePlay can use as its database admin user. Give this user full privileges (SELECT, AMEND, DROP etc) for the CoursePlay database and make sure you take a note of the name and password for this user as you will need them to continue with CoursePlay setup.

Note this is distinct from the Admin User who is set up to administer CoursePlay itself, which may be a customer contact.

## Database Login Configuration

Finally, you will need to edit the *login.php* file found in the main directory of your CoursePlay installation in order to set the connection to the database you have created for CoursePlay to use. This information is created as part of the MySQL database user setup used to allow access to your MySQL database.

Using a standard text editor you will need to edit the entries as follows:

```
$DBhost = "";
```

put the database network location or hostname between the quotes on this line – usually this will be *localhost*

```
$DBuser = "";
```

put the database admin user created above, between the quotes on this line

```
$DBpass = "";
```

put the admin password you created when creating the database admin user between the quotes on this line

```
$DBName = "";
```

put the name of the database between the quotes on this line. The default is *courseplay*.

You will end up with a *login.php* which looks something like the following:

```
$DBhost = "localhost";  
$DBuser = "myAdminUser";  
$DBpass = "myPassword";  
$DBName = "CoursePlay";
```

With some web-hosting arrangements the *DBUser* and the *DBName* will be the same and may be fixed by the service provider.

If you get an error message

```
ERROR MSG: Cannot connect.
```

This will probably indicate a problem with these settings.

If you try to register an Admin User for your database as detailed in the next section without these login details it will fail.

## Registration

Before you can start using CoursePlay you will need to register your copy and set up an Admin user for the LMS who will add and delete users, upload SCORM packages and so forth.

Using a browser navigate to the URL you are using for CoursePlay and the *index.php* file should automatically direct you to the preset form which allows you to register the current installation of CoursePlay and then set the details for the Admin User who will manage the CoursePlay LMS – add users, courses etc.

Note that, once you have configured your CoursePlay installation the *index.php* will automatically direct you and your learners to the standard Login page (you can edit the layout and design by editing *login.imp* in Opus Pro).



## CoursePLAY Registration

STEP 1 - Enter the relevant details for your site and then click register to create the Registered User for CoursePlay at this URL.  
The Registered User details can be different to the Admin User details provided during the next step.

First Name:  Family Name:

E-mail:

Confirm E-mail:

Registration Code:   
(see your order details)

Default Pad Location:

URL to Register:

CoursePlay Serial No:

**Register CoursePLAY**

On the registration form, the Default Pad details will be automatically filled from the URL of the current CoursePlay installation but you can edit it if it is incorrect. This address will be used to lock registration to the URL. You can have as many courses and as many users on this single URL as you like but if you want a separate installation of CoursePlay you will need to purchase and register a separate licence.

The email address entered here will be used to send any licence issues or technical problems, update notifications and so on, and may be different to the Admin User for the LMS itself

When you click **Register CoursePlay** your details, including the URL will be sent to the DW server and you will be emailed a serial number from Digital Workshop which is locked to the URL of your CoursePlay installation. . If this URL has already been registered you will receive an error message and you will need to buy an additional licence or

This serial number will be used to save the CoursePlay configuration in the file *config.php* and appends an extra section of the registration code so that the configuration is secure to your site and even DW cannot subsequently access it. **You should therefore take a backup copy of *config.php* at this point to ensure you keep your serial number safe.**

You will then be taken to the **Create Admin Account** process detailed in the next section.

## Set Up Administration Account

CoursePlay provides a number of administration functions which will allow you to create your site and populate it with courses. You cannot use CoursePlay without an Administrator's account and this needs to be initialized via our servers.

Complete the form by providing the name and email address of the Admin User.



## CoursePLAY - Create Admin Account

STEP 2 - complete the fields and click create.

The Admin User details can be different to the Registered User provided during registration.

A screenshot of the "CoursePLAY - Create Admin Account" form. The form is set against a light blue background. It contains three input fields: "First Name:", "Family Name:", "E-mail:", and "Confirm E-mail:". To the right of these fields is a dark blue button labeled "Create Admin User". Below the input fields, there is a note: "The initial CoursePlay login details will be sent to this email address." The form is contained within a light blue rectangular area, which is itself on a dark blue background.

Click the **Create Admin User** button

The Admin User will then be sent an automated email via the email provided, confirming the link to the login pad which will enable them to access the CoursePlay Learning Management System as an Administrator. A password will also be provided for Admin use.

### Note:

You cannot add another Admin User once one has been created except by contacting support to have the database details reset.

Although it is not possible to re-initialize the CoursePlay installation once there is already a user called Admin, you may prefer to delete the start pad for security reasons.

## Creating A Developer Test Installation

You can also register a copy of CoursePlay for testing purposes so you can develop your CoursePlay at a test location with test contact details before uploading the completed installation and registering that URL as the licenced version.

You can only register a URL once for the purposes of testing. If you want to test multiple CoursePlay designs you can either create each one in a different location and run the test registration process again, or simply copy the new files over the CoursePlay installation you have already registered.

To run a test Installation use Windows Explorer to navigate to the *pads/start* directory of CoursePlay. Delete the *index.htm* and rename the file *index\_test.htm* to *index.htm* so that this becomes the default file for that directory and proceed with the registration process detailed above. When you want to run the full version simply delete the *index.htm* (the rename *index\_test.htm*) and rename the *index\_reg.htm* to *index.htm* so that becomes the active file for the *pads/start* directory. Move your CoursePlay directory to the final URL location and you can then run the above process to register your CoursePlay in the actual location.

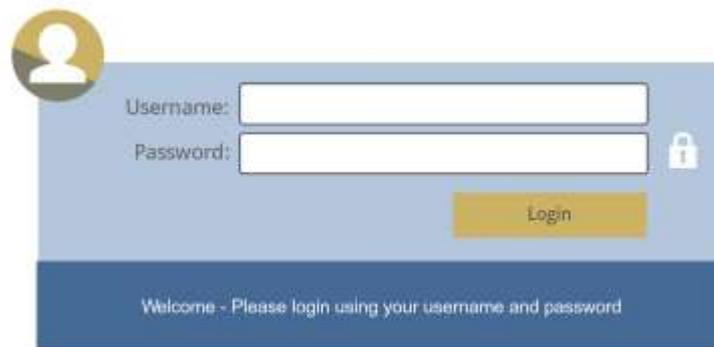
## Creating Your Content

### Edit Your CoursePlay Content Files

Now that you have an active CoursePlay installation with some course files installed you can create the front-end your learner will use to access the courses and reports on their progress.

You will probably edit and publish all the required items locally and before uploading them to your webserver

### User Login



This publication is the page which all users see when they come to login to your LMS. It is provided as *loginpad.imp* in the *pads-default* subdirectory of the *\_templates* directory.

Open this in Opus Pro and edit to suit your house design but do not change any actions or scripts.

Publish as a single page web publication into the *pads/defaults* directory of your local CoursePlay directory. Then rename the resulting *loginpad.htm* to *index.htm* and use an FTP package to copy all the files into the *pads/defaults* subdirectory of the CoursePlay directory on your webserver.

Note that CoursePlay handles all the backend scripting to verify the user and allow them to reset their password if they've forgotten it. You can simply redesign this interface as you wish.

### Courses Interface

This is the primary publication used for learners to access the courses and reports on their progress.

Run Opus Pro and open the *course-menu.imp* from the *pads-course-interfaces* subdirectory of *\_templates*.

Please do not edit the *verifyPage* in any way.

You can redesign, the *menuPage* if you wish, adding logos or instructional text for example.

The *browsePage* is the main page from which your learners will access their courses and comes with some placeholder text and images.

Even if you are not changing the CoursePlay layout or basic design, you will need to edit the text of the course/lesson titles and descriptions and change the associated thumbnail images to reflect your content.

The screenshot displays the CoursePLAY interface. At the top left is the CoursePLAY logo with the tagline 'learning course launchpad'. At the top right is a 'Click to Logout ()' link. A dark blue banner across the top contains the text 'Please choose the course to review or launch'. Below this banner are four course cards, each with a thumbnail image on the left, a title and description in the middle, and 'My Progress' and 'Launch' buttons on the right. The cards are: Course 1 (mechanics), Course 2 (violin), Course 3 (nature), and Course 4 (beer).

You can add additional copies of this page if you require. You can also create interstitial pages to break the courses into modules or broader course structures. These pages would simply connect to the appropriate course menu pages. You can also add any other learner elements you may require such as more detailed course descriptions, instructions etc. You simply need to ensure that you include the verify page to get and verify the current user at the beginning and use a version of the *browsePage* as the final launch page for courses.

Each individual frame on the *browsePage* launches an individual lesson in the CoursePlay database. These are connected to SCORM packages which you upload to the LMS using the relevant SCOID, as detailed in the following two sections.

Once you have completed your editing, and connected the courses, simply publish as a standalone, single page web publication into the *pads/course-interfaces* subdirectory of your local CoursePlay installation. Then rename the resulting page to *index.htm*. Use an FTP program to upload all the content in the publish directory and any subdirectories to the *pads/courseinterfaces* directory of your CoursePlay installation.

### Testing Your Files

Until you create a functioning CoursePlay installation on your webserver, you can only preview the design of your content elements. Running the files in Opus preview or as local files will result in errors as the publication cannot connect to the database so you cannot test them fully without uploading to a CoursePlay installation.

Similarly, any SCORM courses you create will generate errors until they are installed onto a SCORM LMS, however, you should still be able to test the basic design and functionality by publishing as an HTML5 web publication and running that file locally.

## Adding SCORM Courses to the Database

Once you have initialized an Admin account you can also start adding courses and users. CoursePlay currently supports compliant SCORM v1.2 courses from Opus Pro or any other SCORM authoring tool.

### To add a SCORM package to CoursePlay:

1. Log into CoursePlay using the new ADMIN account from above.
2. On the CoursePlay Admin Launchpad, click the **Package Manager** option. The Package Uploader Screen will appear.



3. On the Package Uploader Screen:
  - a: Enter a brief title – such as *Health & Safety Induction 2016*
  - b: Either click the upload area or drag-and-drop the zip file that is your SCORM package to select the content for the course.
  - c: Click the **Upload** button. The uploaded amount will be displayed on the progress bar.
  - d: If successful, the package will be registered in the CoursePlay database and the **Report** button will show.
  - e: Click the report button which will load the Content Packages Report in a new window.
  - f: Locate the package you uploaded and make note of the SCOID/s. If the SCORM package contained more than one lesson then you will have multiple SCOIDs. You will need the SCOID to connect the link to the course in your course menu publication.
  - g: Close the report and return to the Admin Launchpad.

More information on the Content Packages Report can be found later in this guide. For those using Opus to create the SCORM Content Package, the **Read Guide** button on the Uploader gives more information on completing the Manifest Element Dialogue displayed during Publishing.

The Uploader uses the Content Package's name for storage and extraction. It is not possible to upload two packages with the same name. It is also advised that you use a sensible and standard naming format using alphanumerical and dash characters only such as:

*dw-health-safety-induction-2016.zip*

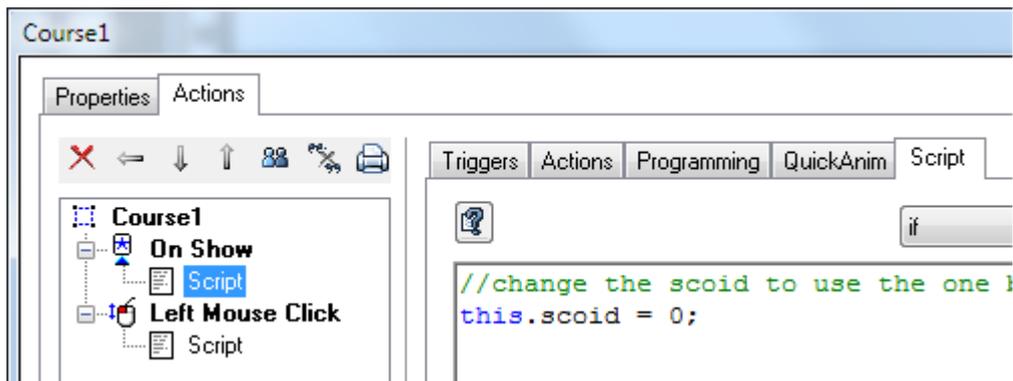
The package name must also not include spaces or the *Remove Package* feature on the Admin pad will not work. The default filename will come from your Opus Publication title, so make sure this reflects the naming requirements.

## Connecting Courses

Once you are happy with the design of your course menu publication you can connect the courses themselves.

On the *browsePage*, right click on each Course frame and select **Edit Actions**.

Double-click on the **Script** under the **On Show** trigger.



Change the following line of script to include the SCOID value you noted when uploading the course:

```
this.scoid = [your-scoid-goes-here];
```

If you have access to the CoursePlay database you can also get the SCOID from the *scorm\_learningobjects* table.

## Publish the Course Interface

Once you have completed all the editing you require and connected all the courses you can publish the project as a standalone web publication. You may need to change the save location in order to keep your original course interface design.

1. Open the folder containing the published files and locate the HTML5 file – probably *sample user interface.htm* – and rename this file to *index.htm*
2. Load your FTP program and locate the *pads/courseinterfaces* directory of the CoursePlay installation on your webserver.
3. Upload the new *index.htm* file into the directory above replacing any existing file.
4. Close your FTP program and Opus – saving any changes.

## Testing the Courses

You can now check the course menu to ensure everything is working properly. When you run any course a link will be available in the player which will show SCORM debug information in a separate window so that you can analyse and fix any problems.

1. Log into CoursePlay using the ADMIN account.
2. On the Admin Launchpad, click the **Courses** icon. This loads the published version of the course interface publication created in the step above.
3. After the Course Menu has loaded, click the **View Courses** button.

## CoursePlay User Guide

4. On the sample course you edited, click the **My Progress** button. This will display a summary of results for your attempt at the course.
5. Close the Progress dialogue and click the **Launch** button.

This will the load Content Package, using the SCOID, which you uploaded above. The content will be displayed in a new window so you need to ensure that popups are enabled for your CoursePlay domain.

6. Complete your course and then return to the Courses Pad.
7. You can either close CoursePlay or use My Progress to check your results first.

## Adding Users

You can now add other users so that learners can access your courses. CoursePlay comes with two user types by default – Standard Users and Administrators but you can also add different user types if you wish.

You have already added the Admin user so now let's see how to add a Standard User.

### Adding a Standard User

1. Log into CoursePlay using the ADMIN account.
2. On the Admin Launchpad, click the User Manager icon.
3. After the User Manager Pad loads, click the add user icon.
4. On the **Add User** Form, complete the details and click submit.

The screenshot shows the 'CoursePLAY Add User Form' interface. At the top, it says 'CoursePLAY Add User Form' and 'complete all fields and submit'. The form has several input fields: 'ID:', 'Username:', 'Forename:', 'Surname:', and 'Email:'. Below these are two dropdown menus: 'User Type:' with options 'USER', 'user', and 'admin'; and 'Default Pad:' with the option 'courses-menu'. A blue 'submit' button is located to the right of the form fields. At the bottom left, there is a yellow 'Back' button with a left-pointing arrow.

The form will be validated locally and when sent, any return notifications will appear in the feedback box.

5. When successful, use the back buttons to return to the Admin Launchpad.
6. Click the users report to load the PHP Report in a new window.

This will show a list of current users. For more information see the Reports Section.

### Standard User Accounts

Technically, there is no difference between the terms USER and ADMIN until you define the LEVELS and RIGHTS.

CoursePlay is supplied with these two values already configured.

ADMIN is set at level 1 with access to all reports and pads.

USER (Standard) is set at level 10 with access to the Course Menu pad and the current user's Results Report.

Later we will see how to create different types of accounts and give them access to existing or new pads and reports. For now, let's test our new account.

1. On the login page, enter the USERNAME for the new USER account you created above and press return or click into the password box.
2. Enter the password for the account and press return or click login. The password will be sent to the email address supplied on the new user form.

If a password is entered incorrectly, a reset button will appear. Simply enter your email address to receive a new password.

3. At the top of the page is a logout link together with the current user and access level.
4. Because USER account level access is greater than 1, logging in loads the Course Menu pad instead of the Admin Launchpad page.

If you needed to change the *login pad*, it can be found in `/_templates/pads-default` folder.

## Controlling Access Rights for Users

CoursePlay is supplied with a USER and ADMIN type account but you may wish to add more granular access or use a more appropriate noun type.

1. Log into CoursePlay using an ADMIN account.
2. On the Admin Launchpad page, click the **User Manager** icon.
3. On the User Menu, click the **add level** icon and the *Add Level Type* form will appear.

The screenshot shows a web form titled "CoursePLAY Add Level Type Form" with the instruction "complete all fields and submit". The form has a light blue background. It contains two input fields: "User Type:" and "User Level:". To the right of the "User Level:" field is a blue "submit" button. At the bottom left, there is a yellow "Back" button with a left-pointing arrow icon.

4. Enter a new account type and level and click **submit**.

The noun used need only be meaningful to you as the administrator of the CoursePlay installation. It could be SUPERADMIN, OFFICEMANAGER, STUDENT, TEACHER or UNICORN.

**Note: the name of the user type must NOT include spaces.**

You can use these for controlling access to courses and reports programmatically using rights and access controls or you can use them to present different content by simply providing different course menus as detailed in the following sections.

## Offering Different Course Menus to Different Users.

The simple way to differentiate the material available to different types of user is to provide different default menus so the user is only offered the options you want.

You can do this by setting the default pad for the user type or by displaying different pages within the same course menu publication.

The former is best where the course menu publication is quite large or there are a lot of variations as this keeps the size of individual course menus to a minimum.

### Default Pad

You can send different users to different content by designing a different course interface and placing this in a dedicated sub-directory of the *pads* directory of your server-side CoursePlay installation.

Thus, you might design a *manager-course-menu.imp* which you publish to *pads/managementcourses* subdirectory you have created.

**Note: the subdirectory name must NOT include spaces.**

Then when you use the *Add User* form to add the user MANAGER, you can then specify this directory as the location for the menu pad for the chosen user type. In the example above you would type *managementcourses* into the *Default Pad* edit box on the form.



The screenshot shows a form with two rows. The first row is labeled 'User Type:' and has three buttons: 'MANAGER' (white), 'user' (yellow), and 'admin' (yellow). The second row is labeled 'Default Pad:' and has a single text input field containing the text 'managementcourses'.

Remember, you must then use the *User Level* form to set the user access level for your new user. Unless you are using the programmatic access control detailed in the following we recommend you set all non-Admin users to level 10 and then control the content they see yourself as above.

## CoursePLAY Add Level Type Form

complete all fields and submit



The screenshot shows a form with two rows and a submit button. The first row is labeled 'User Type:' and has a text input field containing 'MANAGER'. The second row is labeled 'User Level:' and has a text input field containing '10'. To the right of these fields is a blue button labeled 'submit'.

Note that the default pad you direct users too does not have to be limited to just course or report interfaces. For example, you can add some of the elements from the admin pad for some users, or design your own content or reports and allow the user type access to those via dedicated options on their pad publication.

You must also assign a module code to this user type using the Add Module form.

### Different Content in the Same Publication

Alternatively you can redirect users to different content within the same publication. This might be useful where you have users who share a range of courses with only slight variations or where you want to provide different introductory material for the same course menus.

Whenever a learner logs in, CoursePlay automatically puts their CoursePlay usertype into a variable in Opus Pro called *CP\_currUsertype*.

You can therefore create a simple IF...THEN or SWITCH..CASE conditional statement to evaluate the user type and go to a particular page in your publication.

We have provided a simply example of this as *course-menu-types.imp* in the *\_templates/pad/courseinterfaces* directory of the CoursePlay package.

You can use the same principle to use Opus Actions to **Show** or **Hide** content on the same page. This might be useful where you want to add a single course option for one user type while all the other course options are the same.

### Controlling Access via Module Codes and Rights

CoursePlay includes a powerful and flexible system for controlling access to content and reports via coding to allow for more automatic assignment of courses.

This functionality is largely for future developments to CoursePlay and will be developed further in future versions. Additional documentation will also follow in due course. However, if you are familiar with php you can use the information below to build on this functionality yourself. Or contact us to get a price for custom development by CoursePlay developers.

Primary access control is not implemented until you allocate a level and specific module rights to the user type.

The module codes built into this version of CoursePlay are:

- USERMNGR – Level 3 -User Manager Pad, Scripts & Reports
- CPM – Level 3 - Content Package Manager Pad, Scripts & Reports
- CLP – Level 10 - Courses Launch Pad
- SRR – Level 3 - Standard Results Reports

If you are unsure and simply wish to add new labels for learners to use with the manual differentiation of content detailed in the preceding section then we recommend you use CLP for any new user levels you create.

If you create a new module code you will need to manually create php to verify those rights in the various reports and pads you want to assign to those modules.

Secondary access control is more generic than the noun type above. Each type is also allocated an access level from zero upwards. The levels can be used to collectively manage type access and or customise reports or pads. This means you can label users differently using the name of the user type but provide the same levels of access using the numeric user level.

For example, when successfully logging in, the pad checks whether the access level is less than two. This directs any type of level zero or one to the admin page and everybody else to the Courses Pad specified as the default pad for that user level.

Therefore if we created a new type called MANAGER with LEVEL 1, any user of that type logging in would also be directed to the admin page. However, they would not yet have access to any of the reports or pads because MANAGER has not been issued any access rights unlike ADMIN.

The DEVELOPER type is an inbuilt override that provides automatic access to any pad or report without the need to allocate rights. As such for security, it is not possible to create a DEVELOPER type user from the *Add User Pad*.

## Accessing the Current User Information

Whenever a user logs in CoursePlay automatically updates a set of Opus variables allowing you to get some details of your current user as follows:

The unique database username for the current user is stored in the Opus variable *CP\_currUsername*.

User Type – is stored in the Opus variable *CP\_currUserType*

The user's first name is stored in *CP\_currForename*.

## CoursePlay Database Structure

If you are familiar with php you can also develop your own reports, scripts and Opus Pro PostWebData actions to access additional information from the CoursePlay database.

The tables and rows available are listed below.

### Table: *lms\_accessrights*

The table used for controlling the access rights of different user types as created using the Add Rights form in Courseplay.

#### Specification

ID - *int11 NOT NULL AUTO\_INCREMENT*,  
CODE - *varchar8 NOT NULL - From lms\_modules*,  
TYPE - *varchar15 NOT NULL*,  
Primary Key: ID

### Table: *lms\_modules*

Stores details of the modules created using the CoursePlay Add Modules form.

#### Specification

ID - *int11 NOT NULL AUTO\_INCREMENT*,  
DESCRIPTION - *varchar60 NOT NULL -Brief description of module*,  
LEVEL - *int11 NOT NULL -From lms\_userlevels table*,  
CODE - *varchar8 NOT NULL - Module ID*,  
ACTIVE - *varchar1 NOT NULL -Enable or Disable*,  
Primary Key: ID

### Table: *lms\_userlevels*

Stores details of the user levels created using the CoursePlay Add Levels form.

### Specification

ID - *int11 NOT NULL AUTO\_INCREMENT*,  
TYPE - *varchar15 NOT NULL - Name of the level type*,  
LEVEL - *int11 NOT NULL*,  
Primary Key: ID

### Table: **lms\_users**

This is the key table of user information. Use the unique USERNAME from this table to access information about your current user. The username is automatically stored in the Opus variable LOGIN\_USERNAME when a user logs in.

### Specification

UID - *int5 NOT NULL AUTO\_INCREMENT*,  
USERNAME - *varchar12, unique*  
EID - *varchar30 Unique - External Identifier*,  
FORENAME - *varchar30*,  
SURNAME - *varchar30*,  
EMAIL - *varchar60*,  
PASSWORD - *char64*,  
USERTYPE - *varchar15*,  
DEFAULTPAD - *varchar20 - Pad location for login redirect*,  
ACTIVE - *text CHARACTER SET utf8 COLLATE utf8\_unicode\_ci*,  
REGISTERED - *datetime DEFAULT CURRENT\_TIMESTAMP*,  
REGISTERED\_BY - *varchar32 DEFAULT NULL*,  
LAST\_ACCESS - *datetime DEFAULT CURRENT\_TIMESTAMP*,  
ACCESSED\_IP - *varchar65 DEFAULT NULL – stores the IP used to access Courseplay*  
PROXY\_CHECK - *varchar65 DEFAULT NULL – checks whether a proxy has been used*  
Primary Key: UID,

### Table: **scorm\_learningobjects**

Stores the details of the SCORM courses uploaded to CoursePlay. The SCOID is key to accessing information on a particular course element.

### Specification

SCOID - *int11 NOT NULL AUTO\_INCREMENT - Unique ID used for results table*,  
PACKAGEDIRECTORY - *varchar30 - From Packages Table*,  
PACKAGEMANIFESTID - *varchar320 - Package ID from manifest.xml*,  
PACKAGETITLE - *varchar320 - Package Title from Manifest*,  
SCOTITLE - *varchar320 - SCO Title in manifest*,  
SCOIDENTIFIER - *varchar120 - SCO Item Identifier in manifest*,  
SCOIDREF - *varchar120 - SCO Resource Identifier in manifest*,  
SCOLAUNCHURL - *varchar200 - SCO Launch file defined in manifest*,  
SCOACTIVE - *varchar5 - Used to enable / disable SCO*,

DATAFROMLMS - *varchar320* - *Manifest Element*,  
MAXTIMEALLOWED - *varchar320 DEFAULT NULL* - *Manifest Element*,  
MASTERYSCORE - *varchar320 DEFAULT NULL* - *Manifest Element*,  
TIMELIMITACTION - *varchar320 DEFAULT NULL* - *Manifest Element*,  
Primary Key: SCOID.

#### **Table: scorm\_packages**

Stores details of the SCORM Packages uploaded from which the learning objects have been derived. Remember that a single SCOMR package may include more than one learning object or SCO.

##### **Specification**

UID - *int5 NOT NULL AUTO\_INCREMENT* - *Just an autonumber for the table*,  
PACKAGEDIRECTORY - *varchar30 unique* - *Based on name of uploaded zip file*,  
PACKAGEDESCRIPTION - *varchar120* - *As defined by the user on upload*,  
PACKAGEUPLOADER - *varchar30* - *Name of user uploading the zip file*,  
PACKAGEINSTALLED *date NOT NULL* - *Date file was uploaded*,  
Primary Key: UID.

#### **Table: scorm\_results**

This is the key table for accessing a learner's progress and whether the lesson has been completed or other SCORM data models. The ELEMENTID is the name of the data model and the ELEMENTVALUE is the current value for that ELEMENT.

Thus to check the current Lesson Status for a particular lesson for the current user you would get the USERNAME from the LOGIN\_LEARNER\_ID variable in Opus. Match it with the SCOID for the lesson you want to check and then search the ELEMENTID row for the data model *cmi.lesson\_status* and check the ELEMENTVALUE row for COMPLETED or one of the other potential values as detailed in the SCORM specification.

##### **Specifications.**

UID - *int5 NOT NULL AUTO\_INCREMENT*,  
USERNAME - *varchar12*,  
ELEMENTID - *varchar300*,  
ELEMENTVALUE - *text*,  
SCOID - *int11 NOT NULL*,  
Primary Key: UID.

## Reports

A series of preset report pages have been included which provide the key results and user information from the database. These are written in PHP and automatically generate the HTML page from the data. The styling for these is set in the relevant PHP file.



The screenshot shows the 'Content Packages Report' page from CoursePLAY. It features a table with the following columns: PACKAGE, SCORM, Resource Module Title, Launch SCORM Player, and Launch Results Report. The table lists five packages with their respective details and links.

PACKAGE	SCORM	Resource Module Title	Launch SCORM Player	Launch Results Report
<b>PACKAGE</b> Title: CMI Runtime Workshop Description: Interakti's LMS RTE Evaluation Module Directory: ims-manifest	5	Runtime Workshop SC0 with RTE Manifest Elements	<a href="#">SCORM111.htm</a>	<a href="#">click here</a>
<b>PACKAGE</b> Title: CoursePLAY's SCORM Basics Description: Sample Open SCORM Module by Interakti Directory: ims-dev-op-scormbasics	6	SCORM Basics Example1	<a href="#">scormbasics.htm</a>	<a href="#">click here</a>
<b>PACKAGE</b> Title: SampleCourse01 Description: Paul's Sample Course Directory: SampleCourse01b	20	SampleCourse01	<a href="#">SampleCourse01.htm</a>	<a href="#">click here</a>
<b>PACKAGE</b> Title: Interakti Description: Health and Safety Induction 2016 Directory: scp	31	Compliance Package	<a href="#">scp.htm</a>	<a href="#">click here</a>

If you are familiar with PHP you can use these as reference to create your own report pages if you prefer.

The majority can be launched from the Admin Launchpad and you can check the actions in that publication to review how they are structured. You can of course launch them from elsewhere and or provide access to other user types. A brief summary of the reports is as follows:

### Packages Report

This report displays all Content Packages that have been uploaded to the CoursePlay site. It displays both the package title and resource module title extracted from the *imsmanifest* file together with the packages directory and uploader's description.

From the report, you can launch the Resource Module directly into the SCORM player.

### Resource Module Results Report

This report is available as a link from the report above. It shows a breakdown of all SCORM data for a given Resource Module by user's full name.

### Users Report

This report displays a summary of registered CoursePlay users. It shows, key information such as USERNAME and USERTYPE together with useful security data such as who created the account and the last time the user logged into the system.

The database also contains the user's email address and the IP address of the last login.

## Username Results Reports

This report is available as a link from the report above. It shows a breakdown of all SCORM data for a given USERNAME by Resource Module.

## Results Report

This report displays all SCORM results for all USERS for all RESOURCE MODULES.

Technically, CoursePlay uses a custom element called *ims.debug\_history* that records all SCORM calls from the last attempt. Given the potential length of this field, it is not practical to include it on any results report.

However, it could be useful for debugging problematic packages and you can either access it from the database or modify a report to display this element.

## Rights Report

This report displays all the current modules and which USERTYPEs have access rights.

## User Results Summary Report

This is the only report available to everyone logged into CoursePlay and displays a plain speak summary of the SCORM results for all the Resource Modules taken by the current user. By default, it is launched from the sample Course pad.

## Frequently Asked Questions

### **Can I replace a Content Package to update the files?**

Not at the moment using the CoursePlay Uploader. This is to prevent overwriting files that contain information that has already been entered into the database. However, if you must and accept the risk, you can use FTP to locate the package's directory and overwrite a particular file this way.

### **I've lost my ADMIN password, how do I get back in?**

Simply enter the username ADMIN and any random value for the password. This will display the RESET password option. On the reset screen, enter the ADMINs email address to receive a new password.

### **Is it possible to restrict access to specific SCORM Resource Modules?**

At the moment, access to a given resource module is managed by linking them to an access-controlled launch pad. In a future release, CoursePlay will include a dynamic course manager that allows you to create courses and allocate users to these courses.

### **We need a specific function or module built. Is this possible?**

CoursePlay includes a lot of core functionality, and this will continue to develop over time. However, if you need something sooner or bespoke to your installation, please contact Digital Workshop to discuss the requirements.