UK eUniversities
Worldwide Limited

Creating eLearning Content Guide

For version 1h of the UKeU Learning Environment

Draft Edition

Document Version 1.0

3 June 2004

This document is published by:

UK eUniversities Worldwide Limited
14, Buckingham Gate,
London,
SW1E 6LB

Phone: 020-7932-4444
Fax: 020-7932-4494
email: info@ukeu.com
World Wide Web: http://www.ukeu.com

The information in this document is subject to change without notice and does not represent a commitment on the part of UK eUniversities Worldwide Limited. No part of this document may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of UK eUniversities Worldwide Limited.

Access to the eLearning platform described in this document is granted subject to contractual agreements and to UK eUniversities Worldwide Limited's published Terms and Conditions and all use of the elearning platform is subject to the terms of any such agreement andthe published Terms and Conditions. UK eUniversities Worldwide Limited accepts no liability for any damages incurred, directly or indirectly, from any errors, omissions or discrepancies between the eLearning platform and the information contained in this document.

Table of Contents

Chapter 1 Creating Courses in the LCMS

This chapter describes how to use the Learning Content Management System (LCMS) to create courses. This chapter includes the following sections:

- Overview
- Creating HTML pages for learning object files
- Defining a course

Overview

The LCMS is a tool for creating courses. A course is a programme of study that may be offered to students, and contains a hierarchy of components. The highest level of a course may be either a programme, which is comprised of one or more modules, or a single module. Modules are themselves comprised of one or more units and units are comprised of learning objects.

At the core of the LCMS is a component library, which records the details of the learning objects, units and modules you have created. This library is fully searchable and is designed to facilitate re-use of components across courses within an institution. It is possible that subject to suitable agreements components may be shared between institutions as well.

Alongside learning objects, the component library also lists assets, which are the external files such as graphics or animations which are referenced by learning objects. You may also group learning objects and their assets into collections or compound learning objects which can be used together.

There are two possible approaches to constructing a course. You may either use a “top-down” approach and start by defining your programme, adding modules to it, and adding learning objects and collections to the modules, or you may choose a “bottom-up” approach and start by first defining learning objects and later grouping them into units and modules. The LCMS supports both of these approaches to course construction and you may work in either way at any time.

After you have completed building your course, you must create a course package. This package conforms to the IMS specifications for e-Learning content and can be deployed through the UKeU Learning Management System (LMS) or exported for use with other IMS-compliant learning platforms. Packages are used for creating offerings in the LMS.

LCMS terminology

The following terms apply to the LCMS and are used in this chapter:

Component	A general term referring to anything that can be part of a course, including programmes, modules, units, collections, objects, and assets.
Course	A programme of study leading to an award that can be studied by students, which may be a single module course, or a programme comprising one or more modules.

Offering	A programme of study in which students can be enrolled. An offering is a course which begins on a specific date, and has a specific fee associated with it.
Learning object	The smallest possible unit of study in a course. It is usually the equivalent of a single student activity.
Compound learning object	Two or more learning objects of differing types which are presented to the student as a single activity.
Child component	A component that may be included as a child of another component. For details, see the table "Permitted hierarchy of components" on page 12.
Parent component	A component that may include other components as its children. For details, see the table "Permitted hierarchy of components" on page 12.
Asset	An additional file such as a graphic or audio file that is referred to by a learning object file.
Package	A structured collection of learning components used for a course offering.
View	A selection of the learning components in a package that can be seen by specific groups of users.

Procedural overview

This section describes the overall procedure for defining learning content in the LCMS. All of the activities listed in this section are discussed in further detail later in this chapter.

Roles in the system

Different tasks associated with the LCMS are associated with different roles. For example, learning objects are created by an author or a senior author but are approved by a content reviewer. In smaller organisations one individual user may fulfil more than one role.

For more information see "User roles in the LCMS" on page 7.

Specification documents

You must have a specification document for any top-level component (Programme or Module) that is going to be published as a course package. For information on what constitutes a course see the section "Defining a course" on page 22. The specification document for a top-level component must be approved before you start creating a course package.

A specification document usually details the objectives of the course, the topics covered, the course structure, the teaching strategy, and other matters as required by the regulations and practices at your institution.

Optionally you may also have specification documents for other components, and these must also be approved before creating a package which includes these components.

All specification documents are created outside the LCMS and are uploaded as completed documents. For more information see "Creating Specifications" on page 14.

Learning object HTML pages

The HTML pages used for learning objects are created outside the LCMS. When you create a learning object in the LCMS you define the name of your object and associated meta-data, including a reference to the content file which is imported into the LCMS. For more information see "Creating a Learning Object" on page 15.

Other learning resources

Other learning resources, such as graphics or multimedia files, which are used in the LCMS are known as assets. The files used for assets are also created outside the LCMS. When you create an asset in the LCMS you define the name of the asset and associated meta-data, including a reference to the externally-created content file which is imported into the LCMS. For more information see "Creating an Asset" on page 16.

Creating components

Components are created in the LCMS by defining their characteristics, known as meta-data. Where appropriate, the definition of a component includes a reference to an externally-created content file, which is then imported into the LCMS. For more information see "Creating a Learning Object" on page 15 and "Creating an Asset" on page 16.

Assembling components into modules or programmes

The LCMS allows you to choose the order in which you create components and the way in which you assemble them to create a course structure. For more information see "Defining a course" on page 22 and "Course construction strategy" on page 22.

The approvals process

Every component in the LCMS has an approval status associated with it. The status of a component may be **In Progress**, **Ready for Review,** or **Approved**.

Only users with defined responsibility for reviewing and approving content may change a component's status to **Approved**. In some institutions, the same person may be responsible for creating content and approving it.

All of the components you wish to include in a course package must have the status **Approved** before a course can be packaged. For more information see "Component status and the review process" on page 27.

Defining views

When you create a course package you must define which components may be viewed by different classes of users. You must define at least one view before you can create a package. The LCMS allows you to create an **Enrolled Student View**, which normally includes all the learning objects in a course, and a **Registered User View**, which normally includes a small selection of components. The **Registered User View** is available as a sample of the course. For more information see "Creating Delivery Views" on page 29.

Creating packages

Before course material can be published on the UKeU's learning environment (or on any other platform) it must be packaged for delivery. Packaging a course includes checking that all of the components you wish to include in the course package have been approved, and that the required specifications documents exist. For more information see "Packaging the course" on page 30. Course packages are used in the LMS to create and publish course offerings for students.

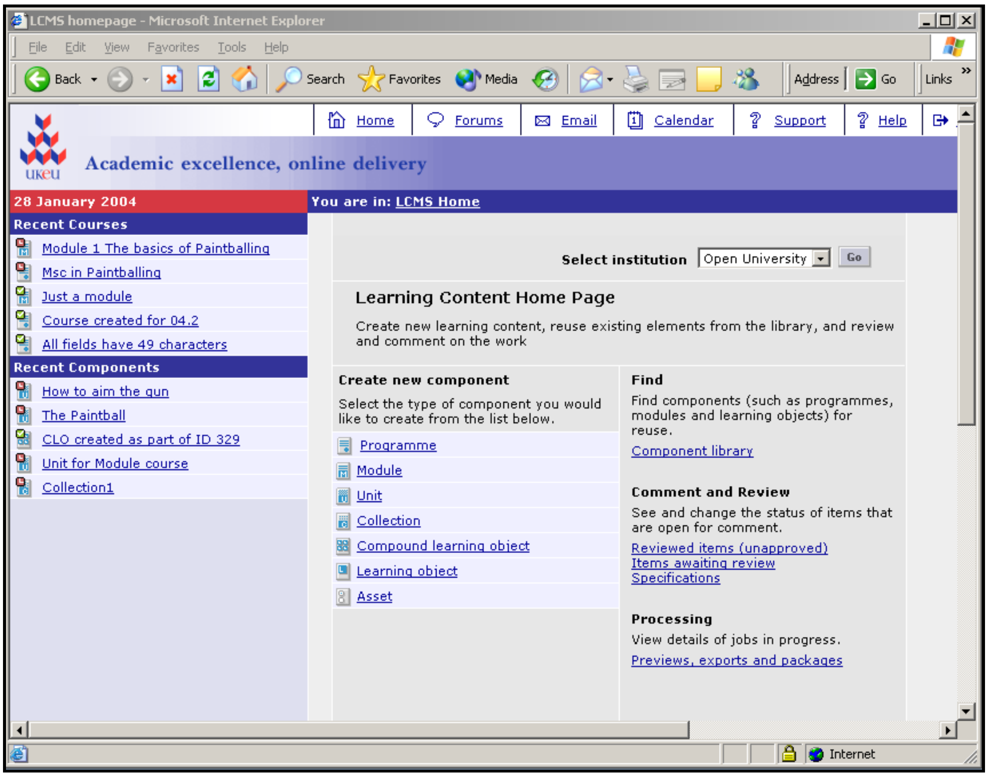
Using the LCMS

This section gives more details about using the LCMS.

Access to the LCMS

Access to the LCMS is restricted according to your role. Users who have the correct permission can access the LCMS from a link on the home page. To display the **LCMS Home** page, click the **LCMS** link in the **Learning Content Management System** area of the home page.

If you do not have access to this page please contact your UKeU representative.


LCMS homepage - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Address Go Links
Home Forums Email Calendar Support Help
ukeu
Academic excellence, online delivery
Internet

28 January 2004

You are in: LCMS Home

Recent Courses

- Module 1 The basics of Paintballing
- Msc in Paintballing
- Just a module
- Course created for 04.2
- All fields have 49 characters

Recent Components

- How to aim the gun
- The Paintball
- CLO created as part of ID 329
- Unit for Module course
- Collection1

Select institution Open University Go

Learning Content Home Page

Create new learning content, reuse existing elements from the library, and review and comment on the work

Create new component

Select the type of component you would like to create from the list below.

- Programme
- Module
- Unit
- Collection
- Compound learning object
- Learning object
- Asset

Find

Find components (such as programmes, modules and learning objects) for reuse.

Component library

Comment and Review

See and change the status of items that are open for comment.

Reviewed items (unapproved)
Items awaiting review
Specifications

Processing

View details of jobs in progress.

Previews, exports and packages

Figure 1-1 The LCMS Home page

Note: *The **Select institution** drop-down list shown in Figure 1-1 is only visible for users who are members of more than one institution.*

The LCMS home page includes a quick access list of **Recent Courses** and **Recent Components**. Click on the name of a course or component in these lists for direct access to the relevant component page

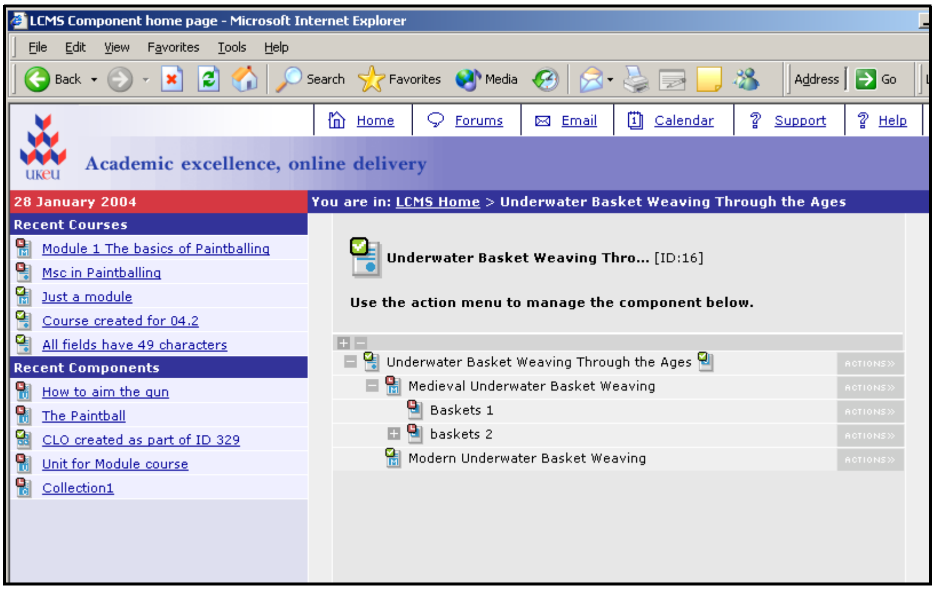
The activities you can perform from the LCMS Home page are:

- Creating new components (See “Creating components” on page 11)
- Searching for existing components (See “Using the component search tool” on page 23)
- Reviewing components (See “Component status and the review process” on page 27)
- Monitoring progress of package creation jobs (See “Monitoring Package Progress” on page 31)

Component page

When you select the details for any component the system displays the home page for the component (the component page). When you view the component page for a higher-level component such as a programme or a module, all its lower-level child components such as units and learning objects are also displayed.

Child components are displayed in an indented list, with each child component displayed below and slightly to the right of its parent component. This display is also known as a “tree structure” view.


LCMS Component home page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
Go
Home
Forums
Email
Calendar
Support
Help
ukeu
Academic excellence, online delivery
ACTIONS»

28 January 2004

You are in: LCMS Home > Underwater Basket Weaving Through the Ages

Recent Courses

- Module 1 The basics of Paintballing
- Msc in Paintballing
- Just a module
- Course created for 04.2
- All fields have 49 characters

Recent Components

- How to aim the gun
- The Paintball
- CLO created as part of ID 329
- Unit for Module course
- Collection1

Underwater Basket Weaving Thro... [ID:16]

Use the action menu to manage the component below.

- Underwater Basket Weaving Through the Ages
  - Medieval Underwater Basket Weaving
    - Baskets 1
    - baskets 2
  - Modern Underwater Basket Weaving

Figure 1-2 Example of a component page showing child components

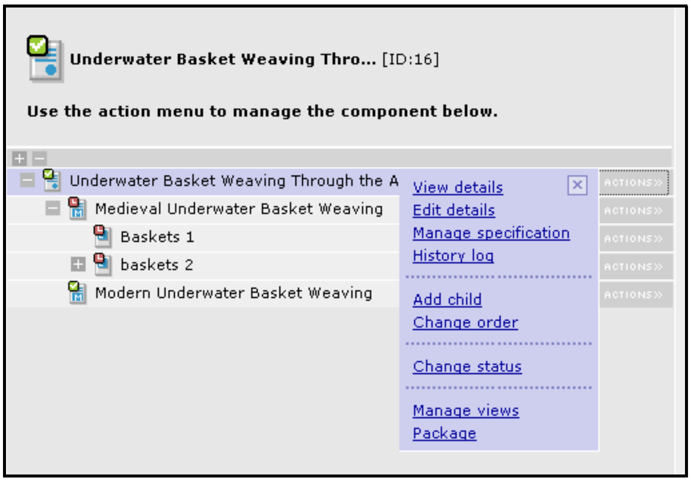
Each component and each of its child components is identified by its name and by an icon. A list of these icons is given in Table 1-14, LCMS component icons, on page 33. Each component icon is shown with a status indicator, and a list of these indicators is given in Table 1-15, LCMS Status icons, on page 34. For information on how to change the status of a component see “Component status and the review process” on page 27.

For details of how to add a child component, see the section “Adding a child component” on page 26.

Action menus

An **Action menu** is available for each component listed in a tree structure view page or in a component search results page.

Click the **Actions>>** button to the right of the component name to display the available action commands.



Underwater Basket Weaving Thro... [ID:16]

Use the action menu to manage the component below.

Figure 1-3 Example of an Action menu

The following table lists some of the standard commands available in action menus. The list of available action commands varies according to the type of component displayed, the component’s status, and your user permissions.

Table 1-1: Action Menu Commands

Command	Description
View details	View the full details for the component including all its meta data fields.
Edit details	Edit the full details for the component.
Manage Specification	Create, upload, or delete a specification for a component, and change the specification status.
History Log	View a list of all the changes to a component, and associated comments, and add your own comments.
Add Child	Add a new component or an existing component as a child of the current component.
Duplicate	Make a duplicate (an identical copy) of the current component. A duplicate includes references to child components.
Re-order	Change the order of child components.
Remove	Remove a child component.
Change Status	Changes the status of a component: **Make Ready For Review** – used by authors to pass work to reviewers **Return to In-Progress** – reviewers may return work to authors if it is not approved and authors may reclaim work sent for review. **Approve** – users with 'reviewer' privileges may set work to approved. **Logically Delete** – authors and senior authors may logically delete components that are not in-use (have no parents).
Quick Preview	View the component in a format similar to the student’s view.
Restore	Returns a component from the logically deleted state to the in-progress status.
Create Views	Create the delivery views for a course.
Package	Create the delivery file for a course. Only available for top-level components.

The component library

All the components that are created in the LCMS are stored in the component library, and may be reused across your institution. You can browse through the component library for components of a particular type, or you can search the library. For detailed instruction see the section “Using the component search tool” on page 23.

User roles in the LCMS

There are eight specified user roles in the LCMS, which are described in Table 1-2 below. Each user role is associated with a number of tasks such as specifying courses, defining content, assembling courses, approving course components and courses, and packaging courses.

Roles used for creating specifications and course content are separate from roles used for approving specifications and content. This is designed to accommodate organisations where these functions are performed by different people. However, in other organisations these functions may be performed by the same person, and therefore it is possible for an individual users to be assigned to two or more roles, and may therefore be able to perform a wider range of tasks.

Earlier implementations of the UKeU Learning Environment used a different system with fewer roles, which are included in Table 1-2 for comparison.

Table 1-2: Business roles

User role	Activities	Previous role
Specifier	• Create and upload of specification documents. • Create packages for preview **Note**: The creation of specification documents takes place outside the LCMS. Components must be created before a specification document can be attached to them.	CMS User
Specification Reviewer	• Review (and optionally approve) specification documents	CMS User
Author	• Create learning objects and components of all types. • Build hierarchies and courses • Uploading learning content • Populate views • Change the status of learning material as appropriate • Create package for preview or export Note: Authors can only edit components they own	CMS User
Senior Author	Senior authors may perform all the activities of an author and in addition may work on components owned by other authors (edit and change state) Senior authors may only operate within the context of their own HEI.	CMS User
Content Reviewer	• Review Component content. • Optionally change status of components. Content reviewers may only operate within the context of their own HEI.	CMS User

Table 1-2: Business roles

User role	Activities	Previous role
Package Creator	• package candidate courses for export, preview or publication • Create Zip files for publication into the LMS • Import packages	PM User
LCMS Administrator	• Physically delete components from the component library after they have been flagged logically deleted. • Change owner for components	UKeU System Admin-istrator
Previewer	• Previewers are not members of your institution, but they can view (but not change) LCMS content, and can publish LCMS packages to their own institution.	-

Assigning user roles in the LCMS

User roles for the LCMS are assigned by the UKeU administrator or the HEI administrator. Two conditions must be fulfilled to allow users to work in the LCMS:

- users must have the role **LCMS User** assigned to them
- users must be added to the appropriate LCMS role group

Creating HTML pages for learning object files

Learning object files are HTML files which contain learning material and may also contain other objects (which are included in the LCMS as assets). Learning object files are created outside the UKeU Learning Environment.

When you create your learning object files you must make sure that they can be viewed in any browser, and that they conform to accessibility standards such as the Web Content Accessibility Guidelines (WCAG). The files should meet the guidelines for a Conformance Level “Double-A” site in which all Priority 1 and 2 checkpoints are satisfied. More information on accessibility can be found at `http://www.w3.org/WAI/WCAG1-Conformance`).

UKeU cascading style sheet (CSS)

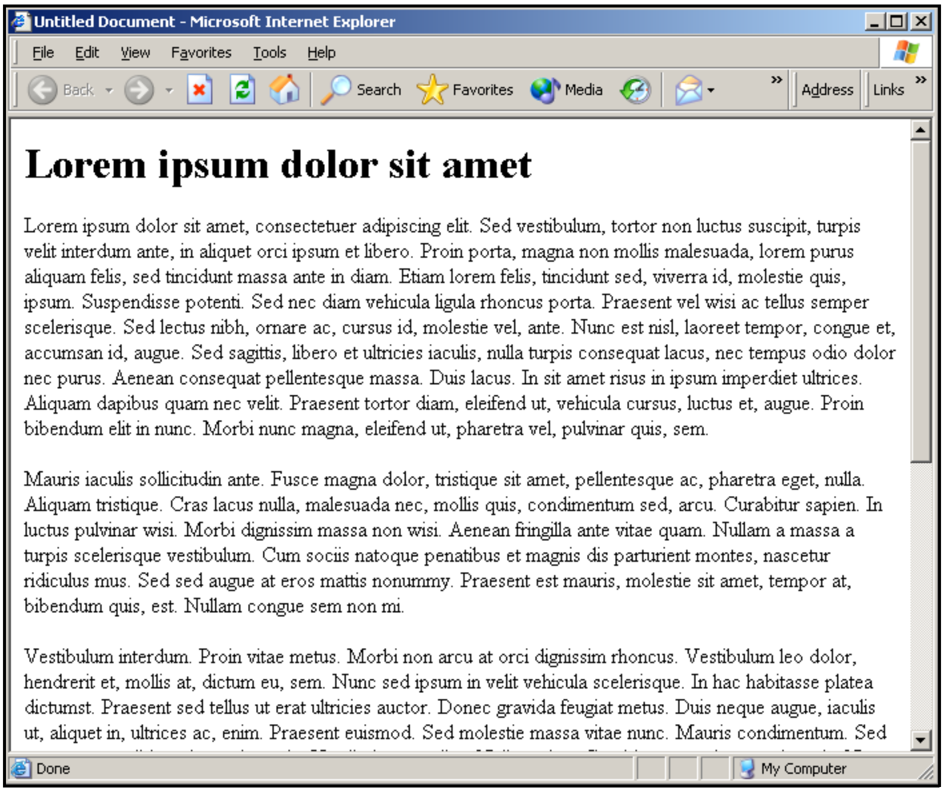
When you create your HTML files, you need to ensure that the text and format styles conform to UKeU guidelines. This aids students by ensuring visual consistency across the UKeU Learning Environment.

UKeU provides you with a cascading style sheet (CSS file). You must include this CSS file in the <HEAD> section of all your learning object HTML files. This style sheet defines the appearance of page text and features including four levels of heading, body paragraphs, ordered and unordered lists, highlighted text, links, and footnotes.

The style sheet also defines the appearance of logical page divisions and layout features, such as pull-quotes (short sections of quoted text), or activities (sections designed for setting an activity requiring feedback).

A reference copy of the CSS file is included in Appendix A.

The illustrations in Figure 1-4 and Figure 1-5 show an HTML page before and after applying the UKeU stylesheet.



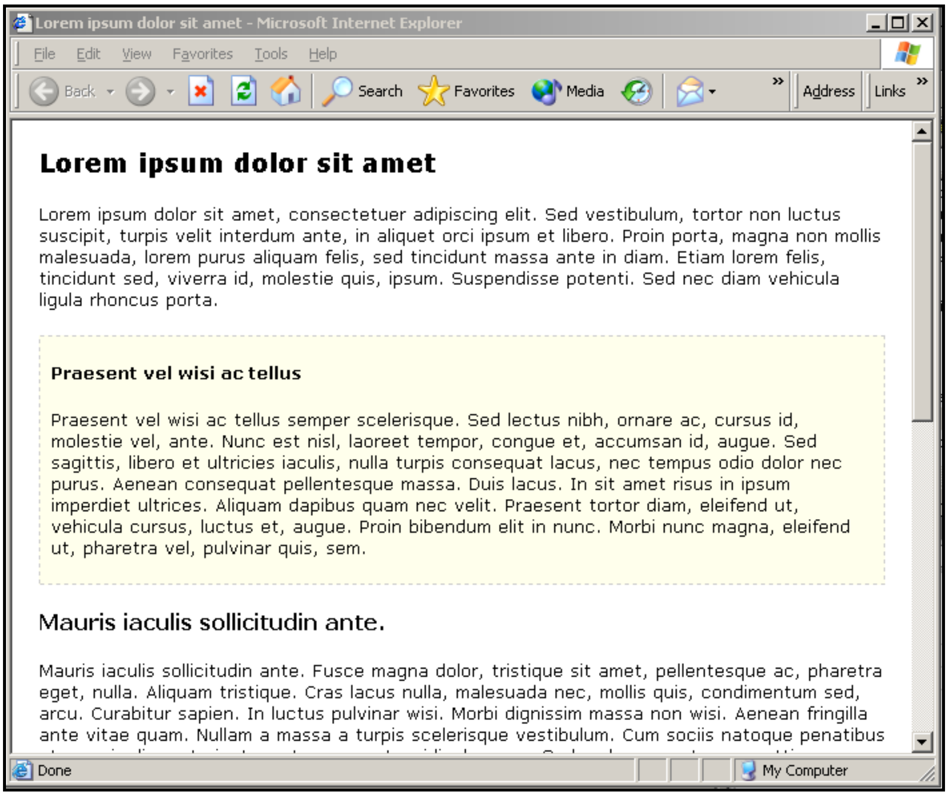
Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed vestibulum, tortor non luctus suscipit, turpis velit interdum ante, in aliquet orci ipsum et libero. Proin porta, magna non mollis malesuada, lorem purus aliquam felis, sed tincidunt massa ante in diam. Etiam lorem felis, tincidunt sed, viverra id, molestie quis, ipsum. Suspendisse potenti. Sed nec diam vehicula ligula rhoncus porta. Praesent vel wisi ac tellus semper scelerisque. Sed lectus nibh, ornare ac, cursus id, molestie vel, ante. Nunc est nisl, laoreet tempor, congue et, accumsan id, augue. Sed sagittis, libero et ultricies iaculis, nulla turpis consequat lacus, nec tempus odio dolor nec purus. Aenean consequat pellentesque massa. Duis lacus. In sit amet risus in ipsum imperdiet ultrices. Aliquam dapibus quam nec velit. Praesent tortor diam, eleifend ut, vehicula cursus, luctus et, augue. Proin bibendum elit in nunc. Morbi nunc magna, eleifend ut, pharetra vel, pulvinar quis, sem.

Mauris iaculis sollicitudin ante. Fusce magna dolor, tristique sit amet, pellentesque ac, pharetra eget, nulla. Aliquam tristique. Cras lacus nulla, malesuada nec, mollis quis, condimentum sed, arcu. Curabitur sapien. In luctus pulvinar wisi. Morbi dignissim massa non wisi. Aenean fringilla ante vitae quam. Nullam a massa a turpis scelerisque vestibulum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Sed sed augue at eros mattis nonummy. Praesent est mauris, molestie sit amet, tempor at, bibendum quis, est. Nullam congue sem non mi.

Vestibulum interdum. Proin vitae metus. Morbi non arcu at orci dignissim rhoncus. Vestibulum leo dolor, hendrerit et, mollis at, dictum eu, sem. Nunc sed ipsum in velit vehicula scelerisque. In hac habitasse platea dictumst. Praesent sed tellus ut erat ultricies auctor. Donec gravida feugiat metus. Duis neque augue, iaculis ut, aliquet in, ultrices ac, enim. Praesent euismod. Sed molestie massa vitae nunc. Mauris condimentum. Sed

Figure 1-4 HTML page before applying the UKeU stylesheet



Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed vestibulum, tortor non luctus suscipit, turpis velit interdum ante, in aliquet orci ipsum et libero. Proin porta, magna non mollis malesuada, lorem purus aliquam felis, sed tincidunt massa ante in diam. Etiam lorem felis, tincidunt sed, viverra id, molestie quis, ipsum. Suspendisse potenti. Sed nec diam vehicula ligula rhoncus porta.

Praesent vel wisi ac tellus

Praesent vel wisi ac tellus semper scelerisque. Sed lectus nibh, ornare ac, cursus id, molestie vel, ante. Nunc est nisl, laoreet tempor, congue et, accumsan id, augue. Sed sagittis, libero et ultricies iaculis, nulla turpis consequat lacus, nec tempus odio dolor nec purus. Aenean consequat pellentesque massa. Duis lacus. In sit amet risus in ipsum imperdiet ultrices. Aliquam dapibus quam nec velit. Praesent tortor diam, eleifend ut, vehicula cursus, luctus et, augue. Proin bibendum elit in nunc. Morbi nunc magna, eleifend ut, pharetra vel, pulvinar quis, sem.

Mauris iaculis sollicitudin ante.

Mauris iaculis sollicitudin ante. Fusce magna dolor, tristique sit amet, pellentesque ac, pharetra eget, nulla. Aliquam tristique. Cras lacus nulla, malesuada nec, mollis quis, condimentum sed, arcu. Curabitur sapien. In luctus pulvinar wisi. Morbi dignissim massa non wisi. Aenean fringilla ante vitae quam. Nullam a massa a turpis scelerisque vestibulum. Cum sociis natoque penatibus

Figure 1-5 HTML page after applying the UKeU stylesheet

Using layout styles

The layout styles in the CSS file are implemented by using attributes applied to a `<div>` tag. The layout styles and their corresponding `<div>` tag attributes are listed in the following table.

Table 1-3: CSS file layout styles

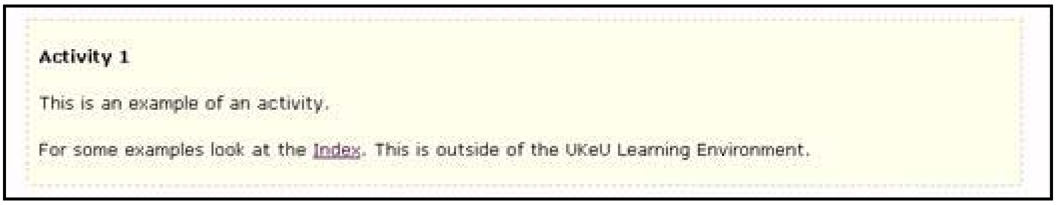
Element	Attribute Name	Description
Images	`imagebox`	This provides a container for captions and labels. The label should be placed inside the <div> tag after the image. Add a line break to separate them. This also provides additional padding so that there is a gap between the edge of the image and the start of another element.
Pullquotes	`pullquotes`	This provides a background and border.
References	`references`	This provides a background and border.
Activities	`activities`	This provides a container for activities to separate it from the rest of the page.
Question	`question`	This provides a style for a list of possible answers on a multiple-choice question. The list is indented to separate it from the question.

For example, to apply the activity style to your text, use the `activities` class, as follows:

```
<div class = "activities">
<h2>Activity 1</h2>
<p>This is an example of an activity.</p>
<p>For some examples look at the <a href="http://
www.ukeu.com">Index</a>.<br>
This is outside of the UKeU Learning Environment.</p>

</div>
```

This will display the text in a box that differentiates it from the rest of the page. For example, this page will display in the following format:



Activity 1

This is an example of an activity.

For some examples look at the Index. This is outside of the UKeU Learning Environment.

Figure 1-6 An example of the "activities" class style

Another example is to apply the question style to your text for multiple-choice questions, use the `question` class.

```
<div class = "question">
<p>This is an example of questions.</p>

</div>
```

If you are using the customised Dreamweaver template, you can apply these styles to your text using the customised UKeU toolbar for Dreamweaver, which is distributed with the template.

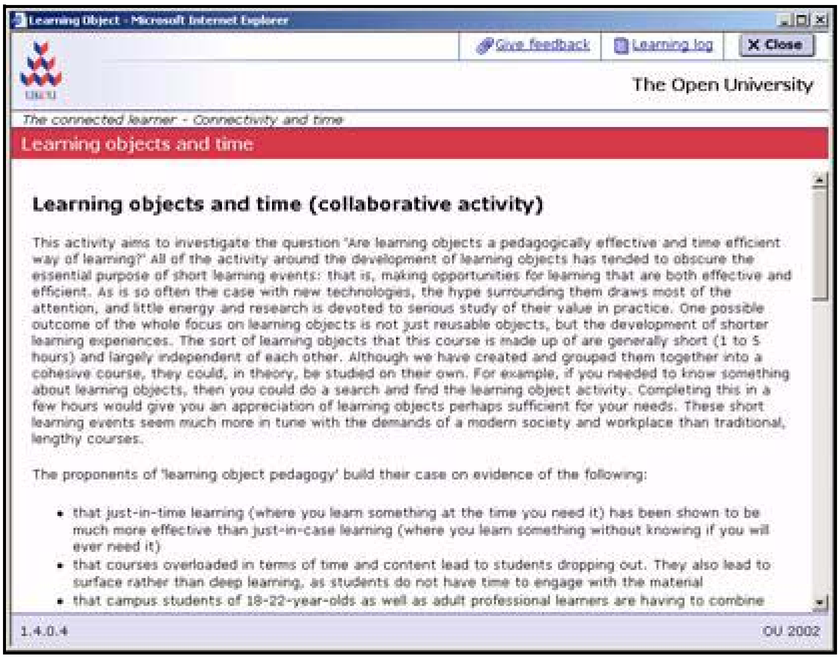
Embedded Media

Embedded media can be inserted in the learning object file in addition to text and HTML tags. Any embedded media needs to be inserted with no text to the left or right.

The supported media types are: `jpg, jpeg, gif, pdf, doc, mpg, mpeg, zip, html, htm, xml, xsl, swf, ppt, txt, text`.

UKeU branding

The overall appearance of the page to students is determined by a set of UKeU visual design features, known collectively as branding features. These features are part of an HTML wrapper that is added to all learning objects automatically. The branding includes features that identify the learning object as belonging both to your institution and your course, and to the UKeU platform. The reason for this branding is to ensure visual consistency across the UKeU platform, which in turn improves the student's learning experience.


Learning Object - Microsoft Internet Explorer
Give feedback
Learning log
X Close
UKeU
The Open University
The connected learner - Connectivity and time
Learning objects and time
1.4.0.4
OU 2002

Learning objects and time (collaborative activity)

This activity aims to investigate the question 'Are learning objects a pedagogically effective and time efficient way of learning?' All of the activity around the development of learning objects has tended to obscure the essential purpose of short learning events: that is, making opportunities for learning that are both effective and efficient. As is so often the case with new technologies, the hype surrounding them draws most of the attention, and little energy and research is devoted to serious study of their value in practice. One possible outcome of the whole focus on learning objects is not just reusable objects, but the development of shorter learning experiences. The sort of learning objects that this course is made up of are generally short (1 to 5 hours) and largely independent of each other. Although we have created and grouped them together into a cohesive course, they could, in theory, be studied on their own. For example, if you needed to know something about learning objects, then you could do a search and find the learning object activity. Completing this in a few hours would give you an appreciation of learning objects perhaps sufficient for your needs. These short learning events seem much more in tune with the demands of a modern society and workplace than traditional, lengthy courses.

The proponents of 'learning object pedagogy' build their case on evidence of the following:

- that just-in-time learning (where you learn something at the time you need it) has been shown to be much more effective than just-in-case learning (where you learn something without knowing if you will ever need it)
- that courses overloaded in terms of time and content lead to students dropping out. They also lead to surface rather than deep learning, as students do not have time to engage with the material
- that campus students of 18-22-year-olds as well as adult professional learners are having to combine

Figure 1-7 HTML page showing UKeU branding

Customised template for Macromedia Dreamweaver

A customised Dreamweaver template is available from the UKeU Developer's Network (http://www.ukeu.net), or directly from your UKeU Project Manager. This template includes all the style and layout tags in the UKeU Cascading Style Sheet, and may be helpful in creating Learning Objects.

If you use the customised Dreamweaver template the UKeU Cascading Stylesheet is added to your HTML pages automatically when you use the **Apply Template** command in Dreamweaver.

Creating components

The LCMS allows you to create components in any order, and then to arrange them into logical groupings for your courses. All the components you create are accessible in the component library which you may search or browse through. The component library is shared between all content creators in an institution.

Once a component has been created it can be used as many times as it is needed. For example, if you have a graphic element that you wish to use in all the HTML pages for your course, you can create it once as an **Asset** and then use that **Asset** as a child component for as many learning objects as you require.

Permitted characters in component names

- You should only use characters from the standard ASCII 256 character set.
- Avoid Word SmartQuotes
- The following characters are not allowed: <, >, &

Component hierarchy

Components in the LCMS can be assembled into modules and programmes according to a fixed hierarchical pattern. This defines which child components are permitted for each type of component.

The permitted hierarchy is shown in the Table 1-4 .

Table 1-4: Permitted hierarchy of components

Hierarchical Level	Permitted Child Component Types
Root level	Programme Module
Programme	Module Compound learning object Simple learning object
Module	Unit Collection Compound learning object Simple learning object
Unit	Collection Compound learning object Simple learning object
Collection	Compound learning object Simple learning object
Compound Learning Object	Simple learning object
Simple Learning Object	Assets

Note: *A compound learning object may not have another compound learning object as a child component (you cannot nest compound learning objects).*

Note: *An asset may not have any child components.*

Assignment learning objects

Assignments for students are created in the LMS but are based on learning objects that have been created as assignment learning objects in the LCMS. This means that if you wish to include assignments in the courses you publish, you must include assignment learning objects in your course packages.

To create an assignment learning object, select one of the **Assignment** types form the drop-down list of educational types when creating a learning object. The current version of the UKeU Learning Environment supports two kinds of assignment learning objects, documents and discussions. Future versions of the Learning Environment will support additional assignment types. For more information about assignments and assessment in the UKeU Learning Environment, please see the *HEI Guide*.

Importance of meta-data

The information that you enter to define a component is known as meta-data (data that describes other data). The categories of meta-data used by the UKeU LCMS are derived from industry-standard specifications such as the IMS Meta-Data Specification. For more information about this specification, see the IMS Web site at `http://www.imsproject.org/`

You should allow time to enter as much meta-data as possible when you create a new component, as this helps you and other system users to correctly identify your components, and allows the system to catalogue your components more precisely.

Keywords

You can search for components by keyword in the LCMS Component Library. You can enter up to ten keywords, separated by commas, for each component. The maximum total length for all the keywords for a component is 1,000 characters.

When you create a child component by selecting the **Add new...** option from the parent component, the child component automatically inherits all of the parent component's keywords. The inherited keywords can be changed or deleted as needed.

Component version numbers

You must assign a version number to any component you create or edit. This is a mandatory field for all components. The version number must be in the format *x.y* where both *x* and *y* are numeric, for example *2.0* or 1*3.05*. We recommend that every time you change a component you also change its version number, as this can help you keep track of your work.

Version numbers are used by the system to keep track of changed course packages. When you upload a course package to the LMS to replace an existing course package, the system compares the version number of the top-level component (programme or module) that you are uploading to the version number of the top-level component of the existing package.

If the version number of the top-level component (programme or module) in the course package that you are uploading is not greater than the version number of the top-level component you are replacing the LMS rejects your package, and you are not able to replace the existing package.

For this reason, whenever you package a course which contains a changed component at any level you must increase the version number of the top-level component before you create the package.

Component sub directory

The UKeU learning environment needs to organise course components into a directory structure that is accessible over the Web. The sub directory field is used to identify components with an offering.

You must define a sub directory for every module you create. You may also define a sub directory for a unit, a collection or a compound learning object.

Within a course offering, sub directory names must be unique at each component level. This means that two or more components which have the same parent component must have different sub directory names.

Creating new components

There are links to create new components on the LCMS home page, under the general heading **Create new**.

The following sections give details on the creation of each of these types of components:

- Programme (See “Creating a Programme” on page 21)
- Module (See “Creating a Module” on page 20)
- Unit (See “Creating a Unit” on page 19)
- Collection (See “Creating a Collection” on page 18)
- Compound Learning Object (See “Creating a Compound Learning Object” on page 17)
- Learning Object (See “Creating a Learning Object” on page 15)
- Asset (See “Creating an Asset” on page 16)

Creating Specifications

You can define a Specification Document for any component you create in the LCMS. You must create a Specification Document for any top level component that you publish as a Course. A top-level component may be either a Programme or a Module. The status of the Specification Document must be Approved before the Course is packaged for publication.

The Specification Document is created outside the LCMS, and may contain any information relevant to the component.

The LCMS displays a Specification status icon to the right of the component name for all top level components.

To add a Specification Document for a component follow these steps:

1. Select **Manage specification** from the **Action** menu for the component.
2. On the **Add Specification** page, click the **Browse** button to open the **File Open** dialog for your operating system. Navigate to the correct directory and select the required file, and click the **Open** button to choose the file.
3. Optionally, add a comment about the Specification Document.
4. Click the appropriate radio button to set the status for the Specification Document to either In Progress, Ready for Review, or Approved.
5. Click the **Save** button to upload the Specification Document.

To change the status of an existing Specification Document for a component follow these steps:

1. Select **Manage Specification** from the **Action** menu for the component.
2. On the **Manage Specification** page, click the **Review Specification** link.
3. Click the appropriate radio button to set the new status as either In Progress, Ready for Review, or Approved.
4. Optionally, add a comment.
5. Click the **Save** button to change the status of the Specification Document.
6. To return to the component page from the **Manage Specification** page, either click the **Back to <component name>** link on the left-hand side of the page, or click on the component name at the top of the page.

Creating a Learning Object

To create a new learning object, follow these steps:

1. On the LCMS home page, select the **Learning Object** link from the list of components under the heading **Create new**.
2. On the Create Learning Object page enter the meta-data details for the new object.

   Table 1-5 lists and explains the meta-data fields on this page.

Table 1-5: Meta-data fields for a new learning object

Field name	Mandatory or Optional	Description
Name	Mandatory	A name for the learning object
Description	Mandatory	A description of the learning object
Keywords	Optional	Keywords for the learning object
Learning outcomes	Optional	A summary of the learning outcomes for the learning object
Authors notes/ requirements	Optional	Notes or requirements for other component authors
Version	Mandatory	The version number for the learning object, in the format *x.y* where both *x* and *y* are numeric. Version numbers must be increased manually for all new versions.
Estimated time	Optional	The estimated time for studying the learning object
Owner's name	Optional	The name of the subject matter expert responsible for the learning object
Email address	Optional	The email address of the subject matter expert
Copyright status	Mandatory	Whether or not copyright applies to the learning object
Copyright statement	Optional	The copyright text to be included in the learning object
Copyright cost	Mandatory if Copyright applies	Whether or not there is a copyright cost associated with the learning object

3. Select the educational type for this learning object.

   Choose the educational type that best represents the learning material in the learning object. Educational types give users an indication of the type of material in the learning object, and users can search course material by educational type. You must provide a content file for all the educational types except **Discussion** or **Assignment (forum)**.

   Each educational type has its own icon, which is displayed when students view the course. There is a full list of educational types and their associated icons in Table 1-16, Learning Object Educational Types and their Icons, on page 34.

4. For all educational types except **Discussion** or **Assignment (forum)**, you must define at least one media type. You may select more than one media type, by clicking on more than one option in the list. You can search for learning objects by media type. The available media types are as follows:

- Article
- Audio
- Document
- Graphic
- Interactive
- Presentation
- Spreadsheet
- Web Content

5. For all educational types except **Discussion** or **Assignment (forum)**, you must select a file for your learning object. Click the **Browse** button and use your operating system's **File Open** dialog to select the learning object file.

   File names for learning object files must be unique within a course package. If you need to use the same file more than once, either create a copy with a new file name, or use the **Reuse** command from the parent component to include the learning object more than once.

6. Click the **Save** button to create the Learning Object.

Note: *When you select either the **Discussion** or **Assignment (forum)** educational type, the learning environment automatically creates a discussion forum when your course offering is published. For more information see the Course Administration chapter in the HEI Guide.*

Creating an Asset

To create a new asset, follow these steps:

1. On the LCMS home page, select the **Asset** link from the list of components under the heading **Create New**.
2. On the Create Asset page enter the meta-data details for the new Asset.

Table 1-6 lists and explains the meta-data fields on this page

Table 1-6: Meta-data fields for a new asset

Field name	Mandatory or Optional	Description
Name	Mandatory	A name for the asset
Description	Mandatory	A description of the asset
Keywords	Optional	Keywords for the asset
Learning outcomes	Optional	A summary of the learning outcomes for the asset
Authors notes/ requirements	Optional	Notes or requirements for other component authors
Version	Mandatory	The version number for the asset, in the format *x.y* where both *x* and *y* are numeric. Version numbers must be increased manually for all new versions.
Owner's name	Optional	The name of the subject matter expert responsible for the asset
Email address	Optional	The email address of the subject matter expert

Table 1-6: Meta-data fields for a new asset

Field name	Mandatory or Optional	Description
Copyright status	Mandatory	Whether or not copyright applies to the asset
Copyright statement	Optional	The copyright text to be included in the asset
Copyright cost	Mandatory	Whether or not there is a copyright cost associated with the asset

3. Select the media type for the asset. The available media types are as follows:
   - Article
   - Audio
   - Document
   - Graphic
   - Interactive
   - Presentation
   - Spreadsheet
   - Web Content
4. Select a file for your new asset. Click the **Browse** button and use your operating system's File Open dialog to select the asset file.
5. Click the **Save** button to create the Asset.

Creating a Compound Learning Object

To create a new compound learning object, follow these steps:

1. On the LCMS home page, select the **Compound Learning Object** link from the list of components under the heading **Create New**.
2. On the Create Compound Learning Object page enter the meta-data details for the new compound object.

   Table 1-7 lists and explains the meta-data fields on this page.

Table 1-7: Meta-data fields for a new compound learning object

Field name	Mandatory or Optional	Description
Name	Mandatory	A name for the compound learning object
Description	Mandatory	A description of the compound learning object
Keywords	Optional	Keywords for the compound learning object
Learning outcomes	Optional	A summary of the learning outcomes for the compound learning object
Authors notes/ requirements	Optional	Notes or requirements for other component authors

Table 1-7: Meta-data fields for a new compound learning object

Field name	Mandatory or Optional	Description
Version	Mandatory	The version number for the compound learning object, in the format *x.y* where both *x* and *y* are numeric. Version numbers must be increased manually for all new versions.
Sub-directory	Optional	The name of the sub-directory for referencing the assets related to this compound learning object
Estimated time	Optional	The estimated time for studying the compound learning object
Owner's name	Optional	The name of the subject matter expert responsible for the compound learning object
Email address	Optional	The email address of the subject matter expert
Copyright status	Mandatory	Whether or not copyright applies to the learning object
Copyright statement	Optional	The copyright text to be included in the learning object
Copyright cost	Mandatory	Whether or not there is a copyright cost associated with the learning object

3. Click the **Save** button to create the Compound Learning Object

Note: *Within a course offering, sub directory names must be unique at each level. Two or more components which have the same parent component must have different sub directory names.*

Creating a Collection

To create a new collection, follow these steps:

1. On the **LCMS home** page, select the **Collection** link from the list of components under the heading **Create New**.
2. On the **Create Collection** page enter the meta-data details for the new collection.

   Table 1-8 lists and explains the meta-data fields on this page.

Table 1-8: Meta-data fields for a new collection

Field name	Mandatory or Optional	Description
Name	Mandatory	A name for the collection
Description	Mandatory	A description of the collection
Keywords	Optional	Keywords for the collection
Learning outcomes	Optional	A summary of the learning outcomes for the collection
Authors notes/ requirements	Optional	Notes or requirements for other component authors

Table 1-8: Meta-data fields for a new collection

Field name	Mandatory or Optional	Description
Version	Mandatory	The version number for the collection, in the format *x.y* where both *x* and *y* are numeric. Version numbers must be increased manually for all new versions.
Sub-directory	Optional	The name of the sub-directory for referencing the assets related to this collection
Estimated time	Optional	The estimated time for studying the collection
Owner’s name	Optional	The name of the subject matter expert responsible for the collection
Email address	Optional	The email address of the subject matter expert
Copyright status	Mandatory	Whether or not copyright applies to the collection
Copyright statement	Optional	The copyright text to be included in the collection
Copyright cost	Mandatory	Whether or not there is a copyright cost associated with the collection

3. Click the **Save** button to create the Collection.

Note: *Within a course offering, sub directory names must be unique at each level. Two or more components which have the same parent component must have different sub directory names.*

Creating a Unit

To create a new unit, follow these steps:

1. On the **LCMS home** page, select the **Unit** link from the list of components under the heading **Create New**.
2. On the **Create Unit** page enter the meta-data details for the new Unit.

   Table 1-9 lists and explains the meta-data fields on this page.

Table 1-9: Meta-data fields for a new unit

Field name	Mandatory or Optional	Description
Name	Mandatory	A name for the unit
Description	Mandatory	A description of the unit
Keywords	Optional	Keywords for the unit
Learning outcomes	Optional	A summary of the learning outcomes for the unit
Authors notes/ requirements	Optional	Notes or requirements for other component authors

Table 1-9: Meta-data fields for a new unit

Field name	Mandatory or Optional	Description
Version	Mandatory	The version number for the unit, in the format *x.y* where both *x* and *y* are numeric. Version numbers must be increased manually for all new versions.
Sub-directory	Optional	The name of the sub-directory for referencing the assets related to this unit
Estimated time	Optional	The estimated time for studying the unit
Owner's name	Optional	The name of the subject matter expert responsible for the unit
Email address	Optional	The email address of the subject matter expert
Copyright status	Mandatory	Whether or not copyright applies to the unit.
Copyright statement	Optional	The copyright text to be included in the unit
Copyright cost	Mandatory	Whether or not there is a copyright cost associated with the unit

3. Click the **Save** button to create the Unit.

Note: *Within a course offering, sub directory names must be unique at each level. Two or more components which have the same parent component must have different sub directory names.*

Creating a Module

To create a new module, follow these steps:

1. On the **LCMS home** page, select the **Module** link from the list of components under the heading **Create New**.
2. On the **Create Module** page enter the meta-data details for the new Module.

   Table 1-10 lists and explains the meta-data fields on this page.

Table 1-10: Meta-data fields for a new module

Field name	Mandatory or Optional	Description
Name	Mandatory	A name for the module
Description	Mandatory	A description of the module
Keywords	Optional	Keywords for the module
Learning outcomes	Optional	A summary of the learning outcomes for the module
Authors notes/ requirements	Optional	Notes or requirements for other component authors

Table 1-10: Meta-data fields for a new module

Field name	Mandatory or Optional	Description
Version	Mandatory	The version number for the module, in the format *x.y* where both *x* and *y* are numeric. Version numbers must be increased manually for all new versions.
Sub-directory	Mandatory	The name of the sub-directory for referencing the assets related to this module
Module credits	Optional	The credits available for this module
Level	Optional	The educational level for the module
Owner's name	Optional	The name of the subject matter expert responsible for the module
Email address	Optional	The email address of the subject matter expert
Copyright status	Mandatory	Whether or not copyright applies to the module
Copyright statement	Optional	The copyright text to be included in the module
Copyright cost	Mandatory	Whether or not there is a copyright cost associated with the module

3. Click the **Save** button to create the Module.

Note: *Within a course offering, sub directory names must be unique at each level. Two or more components which have the same parent component must have different sub directory names.*

Creating a Programme

To create a new programme, follow these steps:

1. On the **LCMS home** page, select the **Programme** link from the list of components under the heading **Create New**.
2. On the **Create Programme** page enter the meta-data details for the new Programme.

   Table 1-11 lists and explains the meta-data fields on this page.

Table 1-11: Meta-data fields for a new programme

Field name	Mandatory or Optional	Description
Name	Mandatory	A name for the programme
Description	Mandatory	A description of the programme
Keywords	Optional	Keywords for the programme
Learning outcomes	Optional	A summary of the learning outcomes for the programme
Authors notes/ requirements	Optional	Notes or requirements for other component authors

Table 1-11: Meta-data fields for a new programme

Field name	Mandatory or Optional	Description
Version	Mandatory	The version number for the programme, in the format *x.y* where both *x* and *y* are numeric. Version numbers must be increased manually for all new versions.
Owner's name	Optional	The name of the subject matter expert responsible for the programme
Email address	Optional	The email address of the subject matter expert
Copyright status	Mandatory	Whether or not copyright applies to the programme
Copyright statement	Optional	The copyright text to be included in the programme
Copyright cost	Mandatory	Whether or not there is a copyright cost associated with the programme

3. Click the **Save** button to create the Programme.

Note: *All Programme courses must contain at least one Module.*

Defining a course

A course which can be offered to students may be based on a Programme, which is a collection of one or more Modules, or may be based on one Module only.

The general outline of a course and its scope is given in a specification document. Depending on the procedures at your institution, the course specification or the course content may require approval by a supervisory body. Although this approval takes place outside the LCMS, the system does require that any course being packaged for publication must have a specification document associated with it, and that document must have the status **Approved**.

Course construction strategy

The LCMS allows course designers the flexibility to choose between different strategies for constructing a course. These are known as the "top-down" and the "bottom-up" approaches. There is also a "mixed" approach to course construction.

Top-down approach

In this approach, the course designer begins by defining the highest level component of the course which is either a programme or a module. After this has been defined, the designer adds child components at the next level down. The **Add child** command on the **Action** menu on a component page can be used for this purpose. This command allows you to create components as children of the current component, or to reuse existing components.

Bottom-up approach

In this approach, the course designer begins by creating individual learning objects and assets and then creating the parent components. Parent components may be created independently, after which the designer may use the **Add child** command from the **Action** menu to add components.

Alternatively, after using the **Component Search Tool**, designers may use the **Create New** command from the **Action** menu on the **Basket** to create a new parent component for selected items. For more information see “Using the component search tool” on page 23 and “Using search results and the Basket” on page 25.

Mixed approach

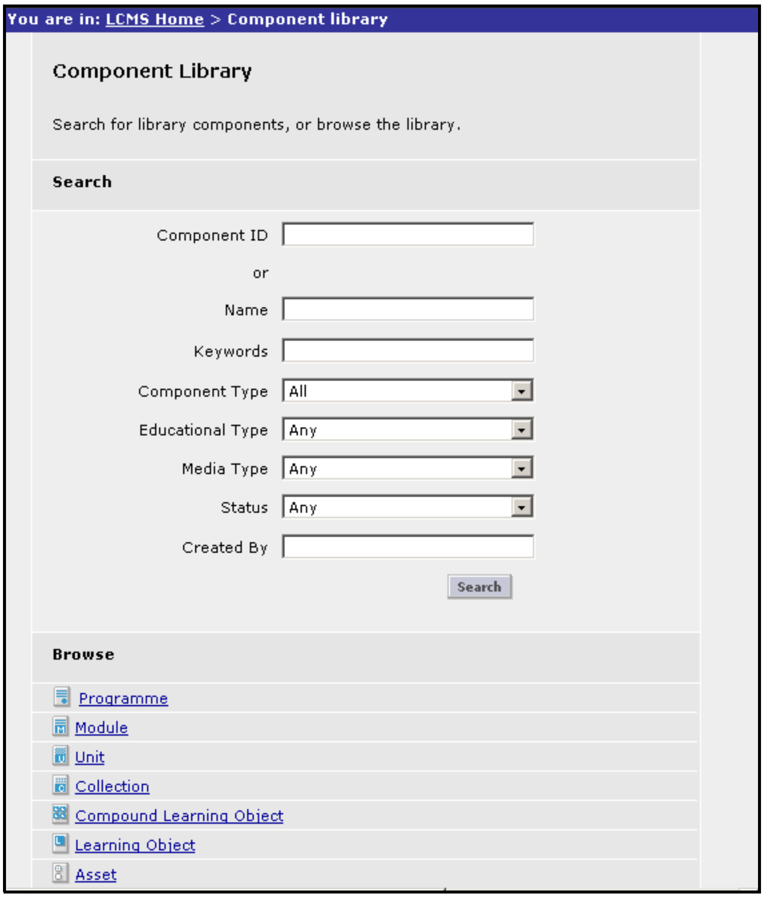
The inbuilt flexibility of the LCMS means that course designers may in practice use any combination of top-down or bottom-up approaches to construct their courses in the way that best suits their needs.

When you select the **Add child** command from the home page of a higher level component, the system displays a link to the component search tool, named **Reuse existing component**. This is one of the ways in which the LCMS supports components re-use and allows for a mixed approach to course construction.

Using the component search tool

The component search tool allows you to look for existing components. To use the component search tool follow these steps:

1. On the **LCMS Home** page select the link to the **Component Library**.
2. The LCMS displays the **Component Library** home page, as shown in Figure 1-8.



You are in: LCMS Home > Component library

Component Library

Search for library components, or browse the library.

Search

Component ID

or

Name

Keywords

Component Type: All

Educational Type: Any

Media Type: Any

Status: Any

Created By

Browse

- Programme
- Module
- Unit
- Collection
- Compound Learning Object
- Learning Object
- Asset

Figure 1-8 Component Library home page

3. On the **Component Library home** page, enter your search criteria. You may search by one or more of the following criteria:
   - Component ID
   - Name
   - Keywords
   - Component Type
   - Educational Type
   - Media Type
   - Status
   - Created By

Note: *The fastest search results are returned when you search by Component ID.*

4. Click **Search** to find components that meet your criteria.

   The system displays the components that match your search criteria.

Browsing the component Library

As an alternative to searching, you may browse the Component Library by selecting one of the component type links under the title **Browse**.

After you select a component type, the browse feature displays an alphabetic selection list. Components are displayed in the same format as search results.

Using search results and the Basket

The results of a component library search and the results of browsing are displayed in a simple list.

Sorting the results list

You can sort the results list by one of the **Name**, **ID**, or **Creator**. The default sort order is by component ID.

To sort the results list, click the underlined column header **Name**, **ID**, or **Creator**. Click again to toggle between ascending or descending sort order.

Search results page action menu

There is an action menu for each component listed in the search results page. The Actions available are as follows:

Table 1-12: Action menu options for the search results page

Option	Meaning
View details	select this option to view the component page for the selected component (see component page)
Copy	select this option to create a duplicate of this component (see copying components)
Delete	select this option to delete the component (see deleting components)

Selecting search results

You can select components from the results of a component library search and use them to create a new parent component. For information on which components can be added as children of another component see Table 1-4, Permitted hierarchy of components, on page 12

To select a component from the search result list, click the check box next to the component name. To select or unselect all components check or clear **All** or **None** at the top of the list.

Using the Basket to create a new parent component

You may copy selected components to an area at the top left-hand corner of the screen known as the **Basket**. Components in the **Basket** may be used to create a new parent component.

Note: *All the components in the basket are used at once to create one new parent component.*

To place a component in the **Basket** click the check box next to the component name, and click the **Add to basket** button. To remove a component from the **Basket** click the red check box next to the component name.

To use the **Basket** components in a new component, select the **Action** menu below the **Basket**. On this menu the **Create new...** option allows you to create a new component. The type of component you may create depends on the selected components in the **Basket**. For example, if you have selected Units, you may create a new Module or a new Programme. If you have selected Learning Objects, you may create a new Compound Learning Object, or Collection, or Unit, or Module, or Programme.

After you select the **Create new...** option the LCMS displays the appropriate Create page for the component you choose. Details of these pages are given in other parts of this chapter. After you create the component, the system displays a tree structure view for the component showing the components selected from the Basket as child components.

Adding a child component

You can add a child component to any component except an Asset. A list of permitted child types is shown in Table 1-4, Permitted hierarchy of components, on page 12. The child component can be a new one that you create, or alternatively you can use the component search tool to reuse an existing component.

To add a child component to an existing component, do the following:

1. From the **Action** menu next to the component name select the **Add Child** command.
2. The LCMS allows you to create a new child component, or reuse an existing component. The system displays a number of links for creating a new component. The links displayed in each case depend on the type of component for which you are creating a child.

   Alternatively you may select the **Reuse Existing Component** link.

Creating a new child component

When you choose to create a new component as a child, the system displays the create component page for the type of component you have chosen. For details of how to create a new component see the appropriate subsection under "Creating components" on page 11.

Enter all the meta-data required to define your new child component, including if appropriate, the details of the source file.

When you have completed your definition, click **Save**. The system saves the new component and displays the original component page. The new child component is automatically listed at the correct point in the component hierarchy.

Reusing an existing child component

If you select the **Reuse Existing Component** link, the system then displays the Component Search Tool page, as described in the section "Using the component search tool" on page 23.

You can either search or browse for the component you wish to use.

The display of search results is described in the section "Using search results and the Basket" on page 25. Because you started the search in order to reuse an existing component, the **Action** menu only contains one command, **Reuse**. You may either Reuse an individual component or reuse multiple components by copying them to the Basket. The only command available for the **Basket** is also **Reuse**.

After you click the reuse command the system displays the original component page. The reused child components are automatically listed at the correct point in the component hierarchy.

Removing a child component

To remove a child component select **Remove** from the **Action** menu for the child component.

Component status and the review process

All components in the LCMS have a status associated with them. The possible statuses are as follows:

Table 1-13: Component status

Status	Meaning
In progress	This is the default status for all new components.
Ready for review	The component author has completed their work and the component should be reviewed by another person.
Approved	The component has successfully passed a review by another person
Deleted	The component has been logically deleted

Note: *Components that have been logically deleted are only visible to users who are LCMS Administrators.*

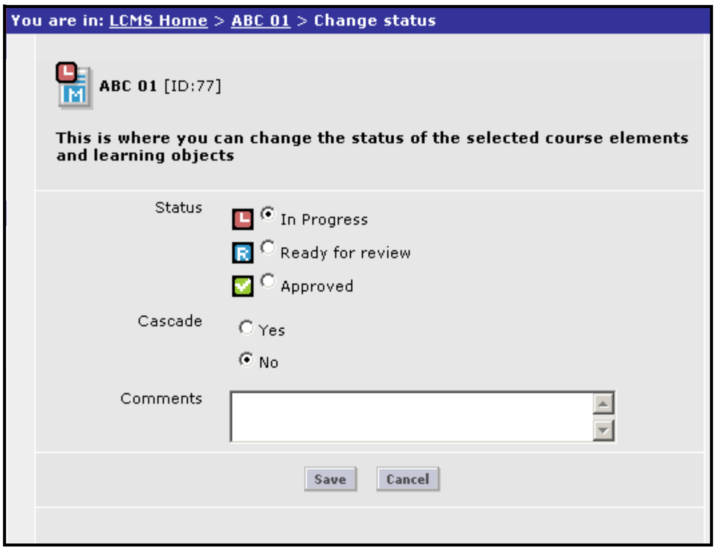
In addition, there is a status of **Missing** which applies only to Specifications.

A list of the icons indicating status is given in Table 1-15, LCMS Status icons, on page 34.

Changing status

If you have the correct user permission, you may change the status of a component. To change the status, follow these steps.

1. Select **Change Status** from the **Action** menu. The system displays the **Change Status** page.



You are in: LCMS Home > ABC 01 > Change status

ABC 01 [ID:77]

This is where you can change the status of the selected course elements and learning objects

Status: ● In Progress ○ Ready for review ○ Approved

Cascade: ○ Yes ● No

Comments:

Figure 1-9 Change Status page

2. Click the appropriate radio button to select the new status.

3. Optionally click the **Yes** button for the **Cascade** command. For more details see the section "Cascading changes" on page 28.

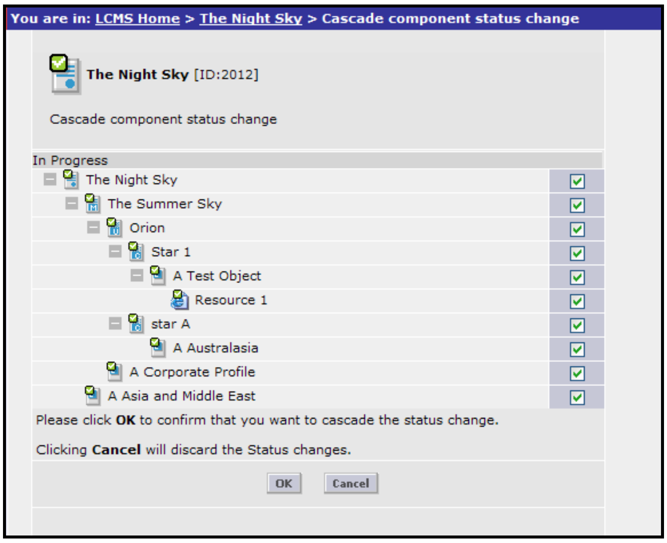
4. Optionally add a **Comment**. This comment is added to the History Log for this status change event.

5. Click **Save** to complete the change of status, and to return to the component page.

Note: *If you edit a component that has the status* ***Approved****, the status reverts to* ***In progress****. The system displays a warning when this is about to happen.*

Cascading changes

You can automatically apply changes to some or all associated child components by selecting the **Yes** button for the **Cascade** command. If you select this option, the system displays the **Cascade component status change** page (shown in Figure 1-10) after you click the **Save** button.



You are in: LCMS Home > The Night Sky > Cascade component status change

The Night Sky [ID:2012]

Cascade component status change

In Progress

- The Night Sky ☑
  - The Summer Sky ☑
    - Orion ☑
      - Star 1 ☑
        - A Test Object ☑
          - Resource 1 ☑
      - star A ☑
        - A Australasia ☑
    - A Corporate Profile ☑
  - A Asia and Middle East ☑

Please click **OK** to confirm that you want to cascade the status change.

Clicking **Cancel** will discard the Status changes.

Figure 1-10 The cascade component status change page

This screen displays a list of all the child components, with a check box along side all the ones whose status can be changed. Check or clear the check box to indicate which child components should have their status changed, and click OK to continue.

Previewing components and courses

The LCMS includes options for previewing components and courses.

Previewing components

You may preview a Learning Object or an Asset by selecting the Quick Preview option from the **Action** menu. The Learning Object or Asset is displayed in its native state, that is, without the UKeU branding.

The review and approval process

If you are responsible for reviewing components created by other users you can access components awaiting review directly from the LCMS Home page. The following options are available:

Items awaiting review

Follow this link for a list of items that have the status **Ready for Review**.

Specifications

Follow this link for a list of **Specifications** awaiting review.

Reviewed items (unapproved)

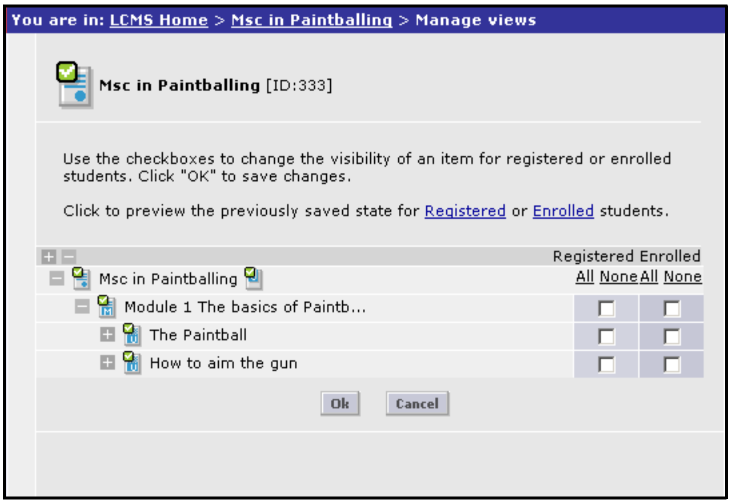
When reviewers do not approve components they change the components' status from **Ready for Review** to **In progress**. These components are listed in the **Reviewed items (unapproved)** list, for further action by content authors.

A separate list of this type is maintained for each institution.

Creating Delivery Views

After you have completed constructing your course you must define the views for the course before you create the course package.

A "View" is a selection of components from your course which may be seen by defined groups of users. The LCMS currently supports two such views, one for Registered Users, and one for Enrolled Students. Registered Users are users who have submitted minimum personal contact information to UKeU to enable them to view the course catalogue. Typically these users would only see a small sample of the course components. Enrolled Students are users who have applied for and have been accepted as students on the course. Typically Enrolled Students would see all the components of the course.



You are in: LCMS Home > Msc in Paintballing > Manage views

Msc in Paintballing [ID:333]

Use the checkboxes to change the visibility of an item for registered or enrolled students. Click "OK" to save changes.

Click to preview the previously saved state for Registered or Enrolled students.

	Registered	Enrolled
Msc in Paintballing	All None	All None
Module 1 The basics of Paintb...	□	□
The Paintball	□	□
How to aim the gun	□	□

Figure 1-11 The Manage Views page

To select a component to appear in a view, click the check box alongside the component name. To select all or none of the listed components, click the **All** or **None** links at the top of the list.

The LCMS saves the details of any set of views for a course, so that you can compare your current set of views to those that have already been packaged. This enables you to check that you have not mistakenly omitted parts of the course when you are packaging a revised version.

To view a previously saved view click either the **Registered** or **Enrolled** link.

Note: *You cannot include any components in a package which do not have the status* ***Approved****.*

Creating a package file

The learning package is a reusable file that contains the resources and meta-data necessary to deliver a course. This file may be uploaded to a Learning Management

System (LMS) and should contain enough information for the LMS to interpret how it must be displayed.

Before you can create the learning package file you must make sure that you have completed all the required stages:

- all the course content has been created, and all the components you wish to include in the course package have the status **Approved** (See “Creating new components” on page 14)
- the top level component (which may be either a programme or a module) has a **Specification** associated with it, and the Specification has the status **Approved** (See “Creating Specifications” on page 14)
- the delivery views for the package have been defined (See “Creating Delivery Views” on page 29)

The format of the learning package file is known as a Package Interchange File (PIF) and this format conforms to the IMS Content Packaging Specification. In the UKeU Learning Environment, the file is deployed as a compressed zip file.

The PIF contains the following components:

- The IMS manifest file called `imsmanifest.xml`
- Physical files (content, media, and so on)

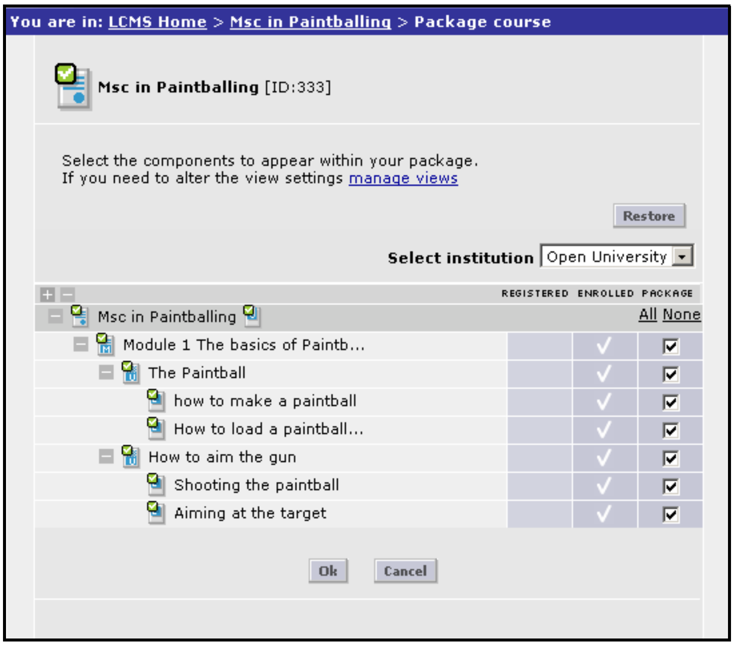
The IMS manifest file contains the meta-data entered when you created the programme specification. The PIF file is deployed as a compressed zip file.

Packaging the course

You can create a course package from either a programme or a module. However, the programme or module must be the top-level item in a component hierarchy. You cannot create a module course package from a module that is not the topmost component in a hierarchy.

To package a course, select **Package** from the **Action** menu. The system displays the Package course page.

Note: *The Package command is not available if the top level component of your course does not have an approved specification associated with it.*



You are in: LCMS Home > Msc in Paintballing > Package course

Msc in Paintballing [ID:333]

Select the components to appear within your package.
If you need to alter the view settings manage views

Restore

Select institution Open University

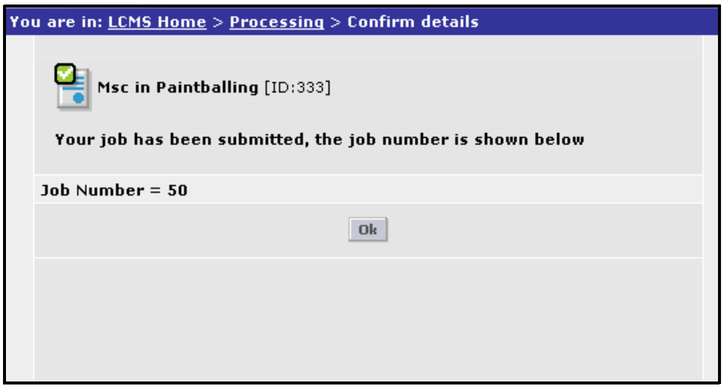
	REGISTERED	ENROLLED	PACKAGE
Msc in Paintballing			All None
Module 1 The basics of Paintb...		✓	☑
The Paintball		✓	☑
how to make a paintball		✓	☑
How to load a paintball...		✓	☑
How to aim the gun		✓	☑
Shooting the paintball		✓	☑
Aiming at the target		✓	☑

Figure 1-12 The Package course page

You may return from the Package course to the **Manage views** page by clicking the **Manage views** link.

To package a course, follow these steps:

1. To select a component to be included in the package, click the check box alongside the component name. To select all or none of the listed components, click the **All** or **None** links at the top of the list.
2. Click the **OK** button.
3. The LCMS displays the Confirm Details page



You are in: LCMS Home > Processing > Confirm details

Msc in Paintballing [ID:333]

Your job has been submitted, the job number is shown below

Job Number = 50

Figure 1-13 The Confirm Details page

4. Make a note of the Job number for your package and click **OK**.

Revising a previously created package

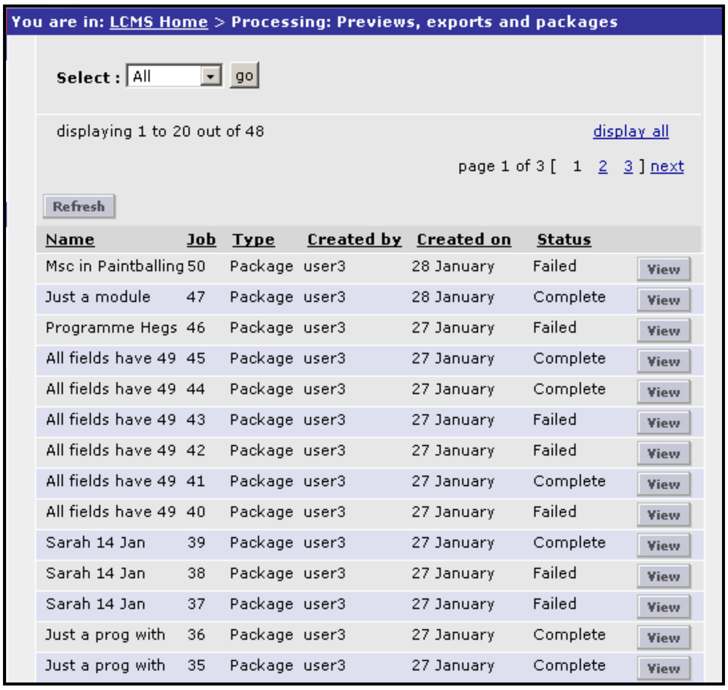
If you wish to add additional components to a previously created package you may do so and create a new version of the package to be uploaded into the LMS.

There are some restrictions in the LMS in the way that revisions to previously created packages are handled. A revised package may contain additional components that had not been included previously, but the LMS only accepts revised packages that contain all the components that were previously included. This means that you can add components to a package but you may not remove anything.

When you are working with a revised version of a previously created package, you need to ensure that you have not omitted any components that were previously included in the package. To do this, click the **Restore** button to revert to the last set of **Include in package** settings for the package.

Monitoring Package Progress

To review the progress of a package creation job, follow the Previews, exports and packages link on the LCMS Home page. This displays the Package status page.



You are in: LCMS Home > Processing: Previews, exports and packages

Select : All go

displaying 1 to 20 out of 48 display all

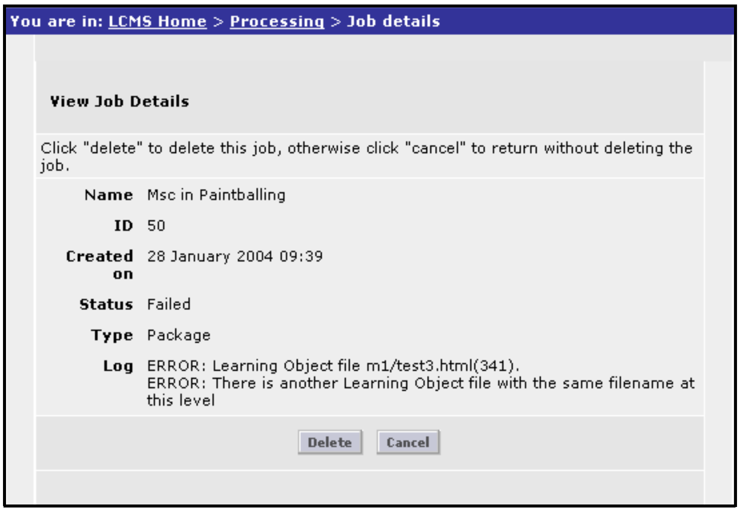
page 1 of 3 [1 2 3] next

Name	Job	Type	Created by	Created on	Status	
Msc in Paintballing	50	Package	user3	28 January	Failed	View
Just a module	47	Package	user3	28 January	Complete	View
Programme Hegs	46	Package	user3	27 January	Failed	View
All fields have 49	45	Package	user3	27 January	Complete	View
All fields have 49	44	Package	user3	27 January	Complete	View
All fields have 49	43	Package	user3	27 January	Failed	View
All fields have 49	42	Package	user3	27 January	Failed	View
All fields have 49	41	Package	user3	27 January	Complete	View
All fields have 49	40	Package	user3	27 January	Failed	View
Sarah 14 Jan	39	Package	user3	27 January	Complete	View
Sarah 14 Jan	38	Package	user3	27 January	Failed	View
Sarah 14 Jan	37	Package	user3	27 January	Failed	View
Just a prog with	36	Package	user3	27 January	Complete	View
Just a prog with	35	Package	user3	27 January	Complete	View

Figure 1-14 The Package Progress page

You can identify your package by its name or by the Job number associated with it. You can sort the details on this page by Name, Job No., Type, Status, the date the package was created or the name of the user who created it. To sort the list click the appropriate column heading. Click again to toggle the sort order between ascending and descending.

To view more details of a particular package, click the **View** button next to the package name. The system displays the Package Details page, which contains more information about the package. If package creation has failed, the Package Details page includes an error message explaining the reason for the failure.



You are in: LCMS Home > Processing > Job details

View Job Details

Click "delete" to delete this job, otherwise click "cancel" to return without deleting the job.

Name	Msc in Paintballing
ID	50
Created on	28 January 2004 09:39
Status	Failed
Type	Package
Log	ERROR: Learning Object file m1/test3.html(341). ERROR: There is another Learning Object file with the same filename at this level

Figure 1-15 The Package Details page

After a package has been created it is available in the LMS and can be used to create module or programme course offerings. The package file is automatically available in the LMS for the institution in which it was created. For details of how to create a package file for use in another institution, please see "Creating a package file for a different institution" on page 33.

For details of how to create course offerings from package files, please see the UKeU's *HEI Guide*, Chapter 2, Course Administration.

Creating a package file for a different institution

LCMS course packages belong to the institution in which they were created. You cannot create an LCMS package file for use in an institution that you are not a member of. If you want to use a course package in a different institution, then the package needs to be created by a person who is a member of that institution.

The special LCMS role, **Previewer**, has been designed for the specific purpose of publishing packages for use in institutions other than the one in which they were created.

You can grant the Previewer role in the LCMS for your institution to a user who belongs to another institution. The Previewer can view but not change the components in your institution's LCMS, and can package a course. The course package created by the Previewer is available in their institution's LMS.

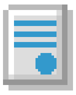
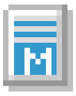
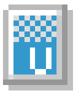
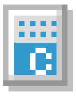
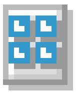
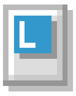
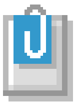
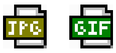
Reference section: icons

This section lists the icons used in the LCMS.

LCMS icons

The LCMS makes use of graphical icons to identify components. These icons are shown in Table 1-14, "LCMS component icons" below.

Table 1-14: LCMS component icons

Icon	Component
	Programme
	Module
	Unit
	Collection
	Compound Learning Object
	Learning Object
	Specification
	Assets (examples)

LCMS Status icons

The status of each component is shown by a supplementary icon indicating its status. These icons are shown in Table 1-15, “LCMS Status icons” below.

Table 1-15: LCMS Status icons

Icon	Status
🕒	In progress
R	Ready for review
✓	Approved
✕	Deleted
?	Missing (shown for Specifications only)

Educational type icons

Table 1-16: Learning Object Educational Types and their Icons

Icon	Educational Type
i	Information
💭	Conceptual Material
🧩	Problem
eg	Example
▦	Task
💬	Discussion
?	Reflection

Table 1-16: Learning Object Educational Types and their Icons

Icon	Educational Type
✍	Application
⚖	Evaluation
🔭	Discovery
🎭	Scenario
🔍	Investigation
✎	Questionnaire
www	Complex Objects
www	Unclassified
✓📄	Assignment (document)
✓💬	Assignment (forum)
✓www	Assignment (web)
✓👥	Assignment (collaborative)
✓☰	Assignment (automatic)
✓☰	Formative assessment (automatic)

Appendix A The UKeU Cascading Style Sheet

```
<style type="text/css">
<!--
body {background-color: #FFFFFF; margin-top: 20px; margin-
right: 20px; margin-bottom: 20px; margin-left: 20px; font-
family: Verdana, Arial, Helvetica, sans-serif;}

h1 {font-size: 110%; font-weight: bold}
h2 {font-size: 100%; font-weight: bold}
h3 {font-size: 80%; font-weight: bold}
h4 {font-size: 80%; font-style: italic; font-weight: bold}

td h1 {font-size: 140%;}
td h2 {font-size: 125%;}
td h3 {font-size: 100%;}
td h4 {font-size: 100%;}

th h1 {font-size: 140%;}
th h2 {font-size: 125%;}
th h3 {font-size: 100%;}
th h4 {font-size: 100%;}

p {font-size: 80%; color: #000000}

div {font-size: 80%;}

a:link {color: #000033}
a:visited {color: #330033}
a:hover {color: #FF0000}
a:active {color: #FF6600}

th {font-size: 80%; font-weight: bold; text-align: left}
td {font-size: 80%}
td td {font-size: 80%}
td p {font-size: 100%}

td div {font-size: 100%}

th div {font-size: 100%}

td ul {font-size: 100%}
td ol {font-size: 100%}

ul {font-size: 80%}
ol {font-size: 80%}

img {margin-top: 3px; margin-right: 10px; margin-bottom:
3px; margin-left: 10px}
.imagebox {float: left; font-size: 80%; font-weight:
normal; text-align: center}
caption {font-size: 80%; font-weight: bold; background-
color: #E1E1E1; width: 100%}
blockquote {font-style: italic; color: #000000; font-size:
80%;}
td blockquote {font-size: 100%;}

.references {background-color: #EEF4F7; padding-top: 0px;
padding-right: 8px; padding-bottom: 0px; padding-left: 8px;
border: #CCCCCC; border-style: solid; border-top-width:
1px; border-right-width: 1px; border-bottom-width: 1px;
border-left-width: 1px}

.pullquotes {font-weight: normal; color: #000000;
background-color: #dfdff1; border: #333399; border-style:
solid; border-top-width: 1px; border-right-width: 1px;
border-bottom-width: 1px; border-left-width: 1px}
.footnotes {font-size: 80%}
.activities {background-color: #FFFFEC; padding-top: 0px;
padding-right: 8px; padding-bottom: 0px; padding-left: 8px;
border: #CCCCCC; border-style: dashed; border-top-width:
1px; border-right-width: 1px; border-bottom-width: 1px;
border-left-width: 1px}



.question {background-color: #ECFFF1; padding-top: 0px;
padding-right: 8px; padding-bottom: 0px; padding-left: 8px;
border: #CCCCCC; border-style: solid; border-top-width:
```

```
2px; border-right-width: 2px; border-bottom-width: 2px;
border-left-width: 2px}

.answerstyle {margin-left: 50px}

.questionfeedback {background-color: #BFFFE1; padding-top:
0px; padding-right: 8px; padding-bottom: 0px; padding-left:
8px; border: #CCCCCC; border-style: solid; border-top-
width: 2px; border-right-width: 2px; border-bottom-width:
2px; border-left-width: 2px}


.highlight {background-color: #FFFF66; font-weight: bold}


-->

</style>
```

Index