

COSE 2

Creation of Study Environments

Version 2.1

Reference Guide: Version 2.1 Supplement

Date: 22th July 2004

COSE - Creation of Study Environments



© Copyright Staffordshire University Enterprises Limited

All rights reserved

No part of this publication, either the whole or any part of the program or documentation, may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language nor computer language in any form or by any means, electronic, mechanical, optical, chemical, manual or otherwise, except as permitted under the licence agreement, without the express written permission of the author(s) or the publisher.

Credits:

COSE was developed at Staffordshire University under funding from the UK Joint Information Systems Committee as part of its JTAP programme.

The COSE Team includes:

Professor Mark Stiles - COSE Project Director and COSE Concept Designer
Sam Rowley
Mark Hewitt
Eddie Clarke
Ray Reid

Thanks to:

Mick Roach who led the work on the COSE Deconstruction Model.
The MMDU at Worldgate who designed the COSE iconography.
All at the JISC for the support they have given.
All the others around the UK who have given suggestions, feedback, and contributed to evaluations of the system.

Introduction	4
New COSE Content Packaging and Interchange	5
How to view Standalone Content Objects generated in COSE.....	6
Features and known issues	7
Caveats: Where interaction is required and MCQ components	8
Interoperability	12
Blackboard	12
TekniCAL Virtual Campus	12
RELOAD	14
Intralibrary	14
WebCT	16
Learnwise version 2 and Learnwise Olympus	16
New Accessibility Enhancements	18
Feedback	18
Required Software.....	18
Overview of the COSE User Interface.....	19
Our Approach.....	20
Guidelines	20
W3C WAI guidelines	20
Java guidelines	21
Implementation	21
Testing and Known Issues	26
Using COSE	27
Operating System Features	27
Assistive Software	27
Mouseless Navigation	28
Appendix 1	35
COSE Content Packaging and Interchange.....	35
Personal Metadata	35
Exporting Pagesets	37
Author interface	38
Publisher interface	40
Administrator's interface	42
Importing Pagesets	46
Supertutor Interface	46
Administrator Interface	48

Introduction

This supplement complements the COSE manual for version 2.063. The new features included in version 2.1 relate primarily to a **new content packaging option** allowing the generation of what we call standalone content objects (SCOs), a significant development in terms of interoperability and content re-use, and **accessibility enhancements** allowing use of screen readers. These are described in some detail here. The figures / screenshots show slight changes in the various interfaces compared with the manual description of previous releases but these are largely cosmetic and would cause no confusion for practised users. The manual has not been updated per se because of the substantial work involved in updating the figures therein, as well as the changes required in the Help pages and the how-to-use-COSE pagesets of the system itself to be consistent with these; and because a new version of the manual will be written for the next major release of COSE, version 3.0, describing, amongst other things, simplified user interfaces, in design

New COSE Content Packaging and Interchange

New in version 2.1 is the packaging option to export COSE pagesets (or content) as Standalone Content which adds flexibility for users in allowing a simple means of viewing pagesets without having to be connected to the server. Standalone Content also provides a solution in interoperability as such packages can be exported for use in other VLEs. Whereas content interchange previously referred to transferring content from one COSE system to another, content can now also be imported to systems of the principal commercial vendors (Blackboard, WebCT, Granada Learnwise / Olympus and TekniCAL's Virtual Campus) as well as to other content development applications and repositories (RELOAD, Intrallect's Intralibrary). While we consider Standalone Content to be a significant development in COSE and are confident that it will become established as an important feature, it is as well to be aware of certain limitations inherent in the production of such artefacts, and that work is on-going in implementing emerging specifications and standards which will likely influence further development. SCOs are currently packaged in conformance with the IMS Content Packaging specification v. 1.1.3.

Those parts of the manual relating to metadata (the section on 'Personal Metadata' at the start of Appendix E and Appendix F: 'Metadata used in COSE') are still relevant here. Appendix E is replicated in this supplement as Appendix 1 with minor edits to show the slight changes to the user interfaces made (mainly since v. 2.063) and, under 'Exporting Pagesets', how the new content packaging option is used.

Standalone Content packages, as native COSE packages, are generated to include the metadata input at different stages in the Export procedure, the content author, 'publisher' and administrator being prompted for such input. Metadata is incorporated with each content package *instance* according to the IMS Learning Object Metadata specification v. 1.2.2. A significant enhancement anticipated in COSE will be to allow searching on this metadata and similar metadata used in tagging other COSE objects to achieve *system* conformance. It seems clear other leading VLEs have implemented instance but not system conformance in respect of metadata, a significant omission and limitation at this time.

The second part of Appendix E relating to 'Importing Pagesets' is not relevant in context of SCOs: standalone content packages were designed to be free-standing, with the option of importing to other systems and applications. However, again, it is anticipated that a future release of COSE will allow import of a wide(r) range of generic packages (including SCOs) which could be viewed, searched for and disaggregated into their components for re-use.

For latest development work on enhanced use of metadata and import of generic packages, see <http://www.staffs.ac.uk/cose/X4L/documents.htm>.

How to view Standalone Content Objects generated in COSE

Like native COSE content packages, Standalone Content packages are generated as .zip files

1. Extract the zip file to a directory of your choice
2. Under the directory 021002\newjscripmtree, find the javascript-frameset.html page
3. Drag this page to a javascript-enabled browser
4. Expand the nodes of the pageset hierarchy by clicking on relevant nodes of tree: either click on the title of any given component or use the + and – icons to toggle a branch open and closed. The text component associated with any page is represented by the blue icon, illustrated below

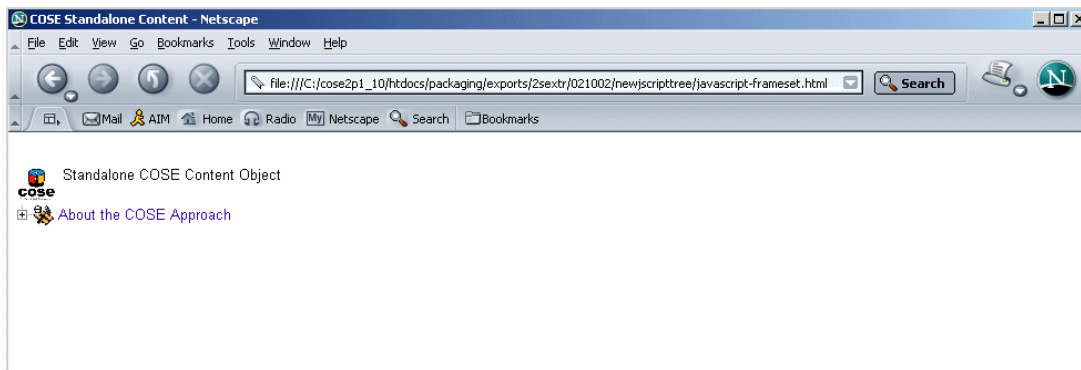


Figure 1. On dragging the javascript-frameset.html page to the browser

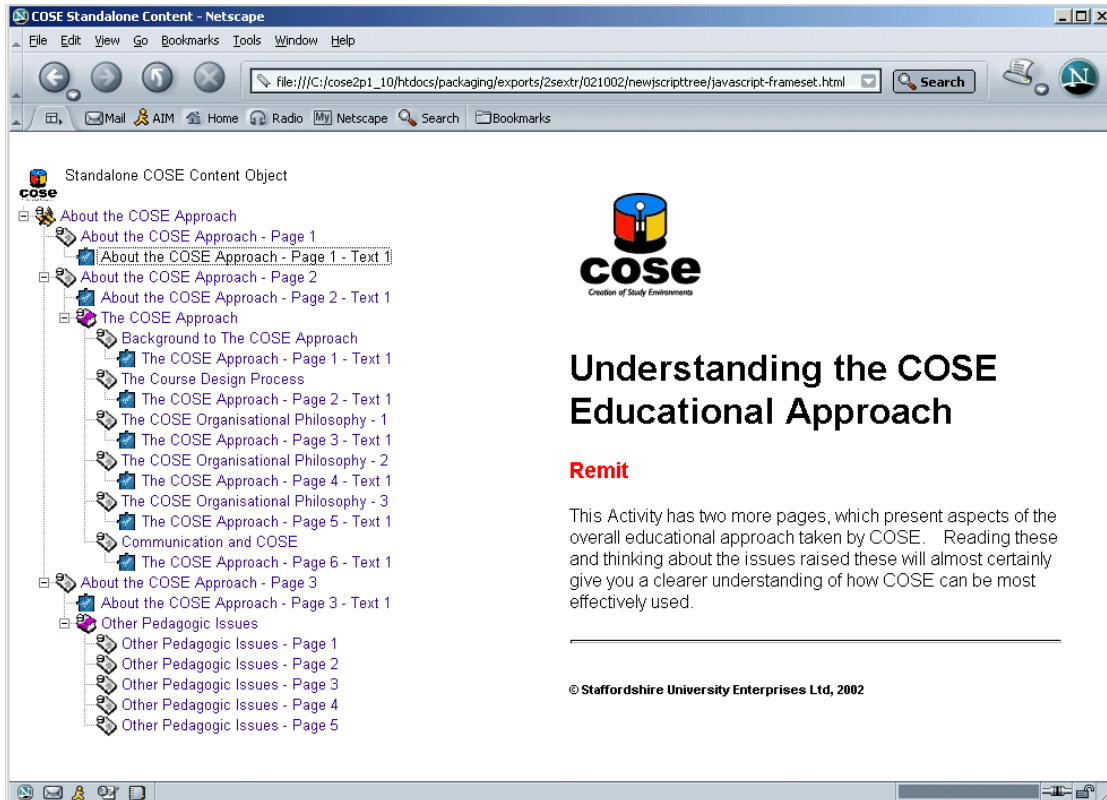


Figure 2. Expanding the pageset nodes and selecting the relevant text component icon in the navigation frame causes the web page to be rendered in the right hand frame

Features and known issues

As with native COSE packages, SCOs use a similar naming convention based on a given system's system identifier and a system counter known as the system entry number. While the latter is opaque to the user, a system identifier is required (must have been assigned) in order to generate the name for a given package. If the system identifier were 0007 and the system entry number were 00031 then the package name for the SCO generated would be 0007_00031s.zip. The s suffix of the name distinguishes standalone content packages and native COSE packages

The packaging process which generates SCOs does not enforce strict adherence of the pageset hierarchy described in the manual, reflecting the same feature of the COSE Editor.

Pageset and page icons of SCOs are associated with blank html pages and differ in this regard to the COSE Browser / Editor which renders in the Browser the first component associated with a page or pageset

Caveats: Where interaction is required and MCQ components

The first advantage of packaging COSE content as SCOs lies in the flexibility for viewing content off-line, dispensing with the requirement of being connected to any given COSE server / system, but this clearly precludes use of the collaborative tools available therein. Normal use of email or even other groupware, would get round this restriction, albeit with the inconvenience (perhaps) of configuring group structures as set up in COSE. While it is useful to be able to view content disconnected from COSE, use off-line disallows use of hyperlinks to external resources. This might have been anticipated by the content author, but it is more likely that the 'standalone' content be used with the web, and the content generated with that in mind.

Multi-choice questions in COSE are generated as self-tests (answers supplied) or assessed tests. Both options are accommodated in generating associated standalone content packages though with some modifications. Scores and answers can be displayed for self tests. Assessed tests and the answers given in the conduct of those tests can be saved for cutting and pasting to email for submission to the relevant tutor or assessor. The options available for randomising questions and / or choice answers, for setting different score values for different questions, and for allowing multiple answers to a given question are accommodated in generating the standalone content. It should be noted though, that while different questions (tests) can be generated from a given set using the randomising questions option when a user is connected to the server, standalone content

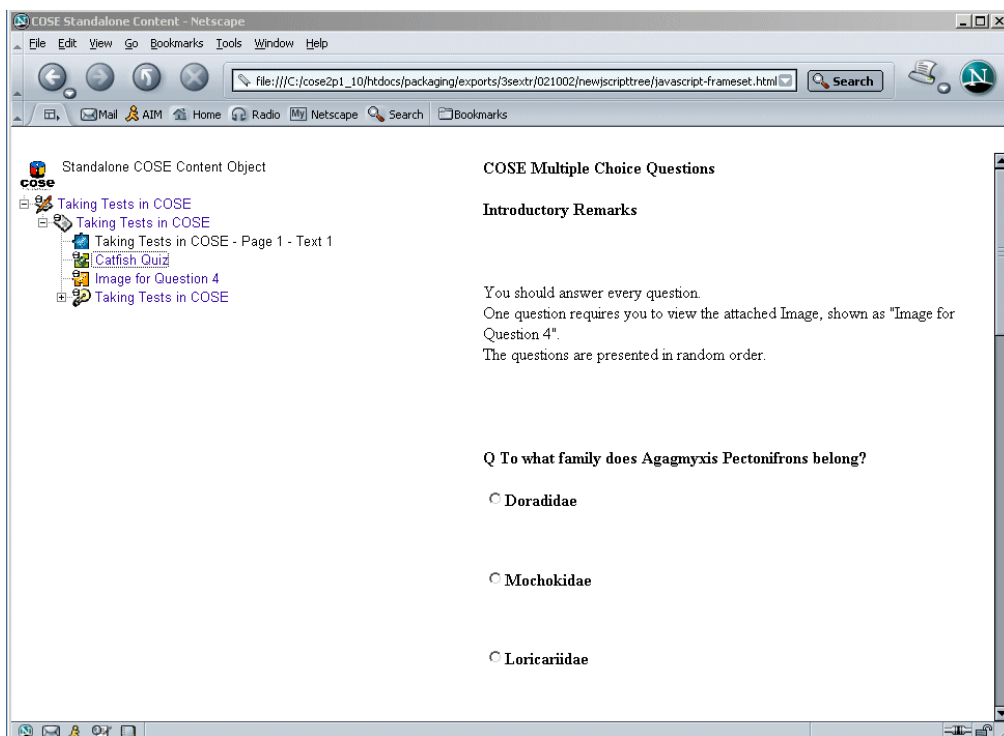


Figure 3. SCO content package with a self-test MCQ (1)

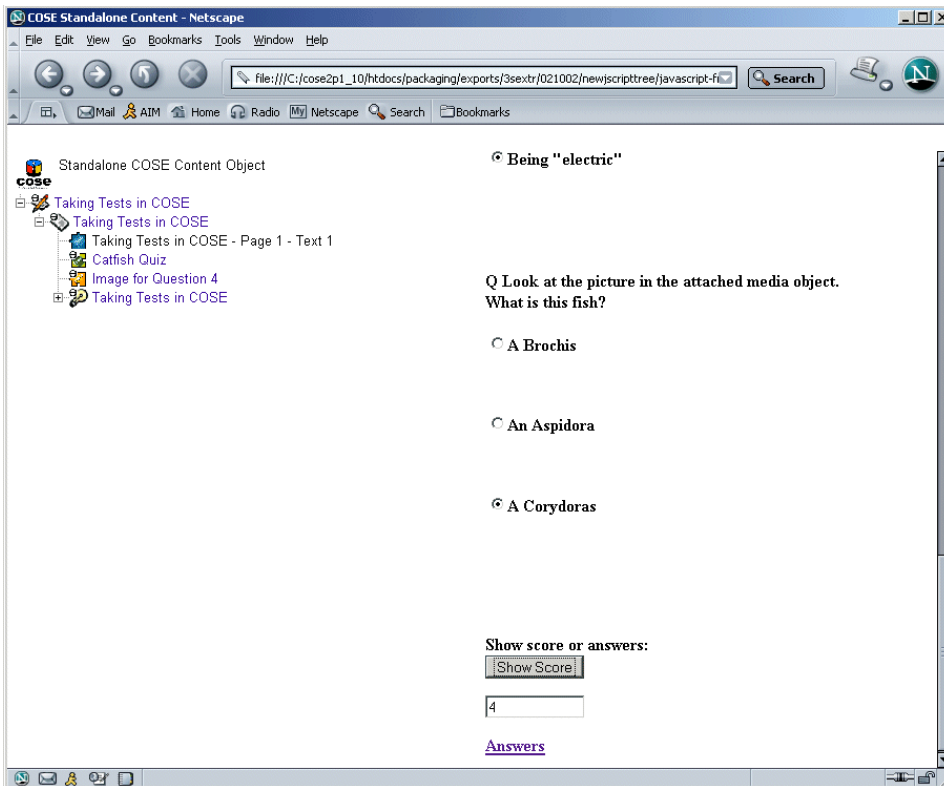


Figure 4. SCO content package with a self-test MCQ (2): showing score

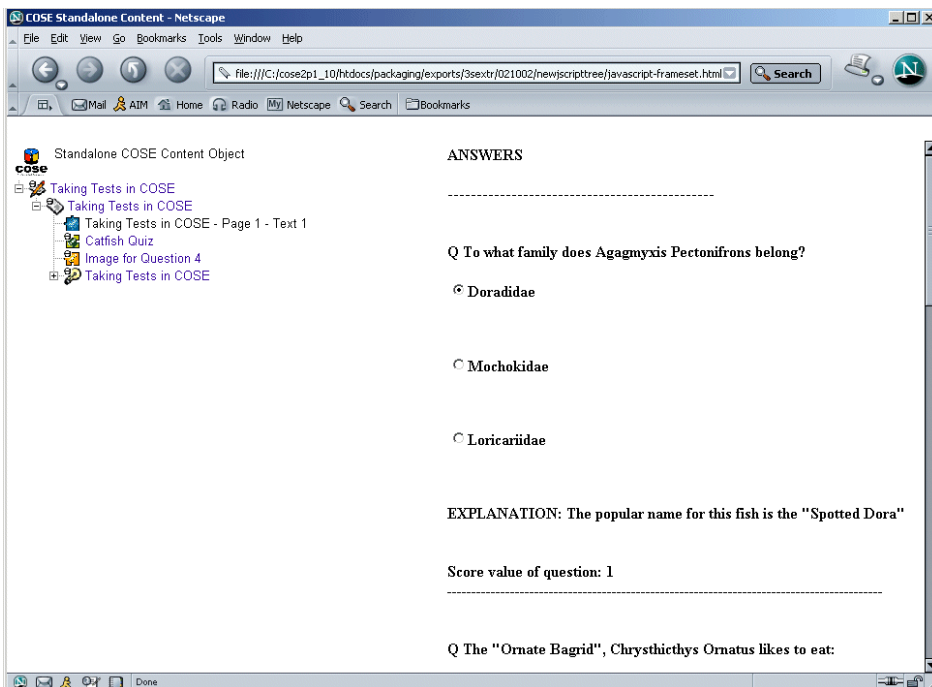


Figure 5. SCO content package with a self-test MCQ (3): showing answers

offers only one set of randomised questions (one test) so is less useful in a class where students could be expected to copy answers from each other. SCOs with MCQ components then, should be used for self-tests rather than for assessed tests; at least, they are not recommended for any formal assessment. The figures here illustrate standalone content containing self-test MCQs and assessed MCQs. There is no answers page packaged with assessed MCQs. (See also for comparison, the exemplar task pageset, 'Taking Tests in COSE' under the Published Pageset tab of the Management tools).

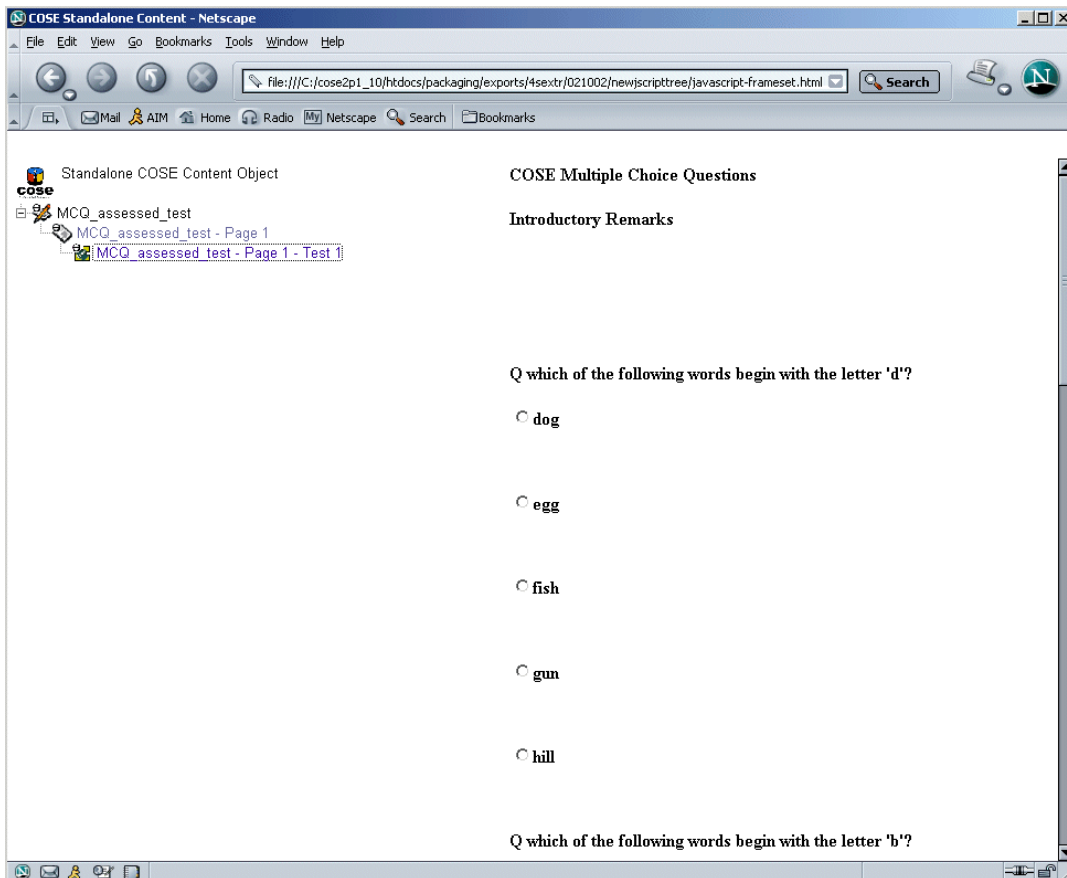


Figure 6. SCO content package with an assessed MCQ (1)

Interoperability

The second advantage of packaging COSE content as SCOs lies in the flexibility it offers in transferring content between systems, so providing a partial solution to the ‘lock-in’ problem common with commercial vendors. The investment institutions make in adopting a particular system can often have serious consequences where it is not possible to migrate the content (or other data) generated in that system to another system (e.g. another vendor’s VLE is adopted in preference) after a period of time. While this migration can be facilitated through adopting international specifications and standards, these are still evolving and the migration problem depends on the nature of the content and other data. Some simple examples of difficulties in migrating content have been discussed above under **Caveats** in our own experience with SCOs.

Notwithstanding these difficulties, what follows demonstrates substantial progress in allowing COSE content to be migrated to other systems. The fact that SCOs can be imported to these other VLEs and other content management systems indicates that they might well be compatible with third party systems which we have not been able to access or test as yet. Systems importing SCOs have yet to be fully tested in terms of being to use or reuse content components and metadata, and in being able to export imported content in its original form. This leads to the more substantive point that COSE Content exported to other systems is at best only half of the problem to be addressed; content migration and interoperability in respect of COSE also depend on being able to import content exported from other systems. As noted above, the next phase of development will allow import of a wide(r) range of generic packages for re-use.

Blackboard

Blackboard version 5.5 is used here at Staffordshire as well as COSE. Figure 9 shows an imported COSE SCO. Using the ‘Unpackage this file’ option in adding content to Course Documents after setting up the course as System Administrator, the dummy course we set up here has the unfriendly name ATCA_0088_00002s, a conjunction of the abbreviated title for the SCO and the name of the package based on some arbitrary system identifier. As found in importing to other systems, there may be an issue relating to screen real estate: the navigation frame of the SCO takes up a significant area of the browser window and scrolling may be required if the content is many layered. There may be different ways to work round this, but a more significant problem which seems unique to Blackboard is that it takes appreciable time to refresh the screen when toggling content components. It may be that upgrading to version 6.0 in the next months provides a solution with more efficient content browser.

TekniCAL Virtual Campus

Figure 10 shows the same SCO imported to TekniCAL’s Virtual Campus VLE (version 4). Subjects (or courses) are created and modified by the content they contain through resources contained in packages which are created separately (all this by the Content Manager tool under the Staff tab). When uploading a resource to a package, options on import type include ‘IMS object’ and this was used for the SCO; the user is prompted to

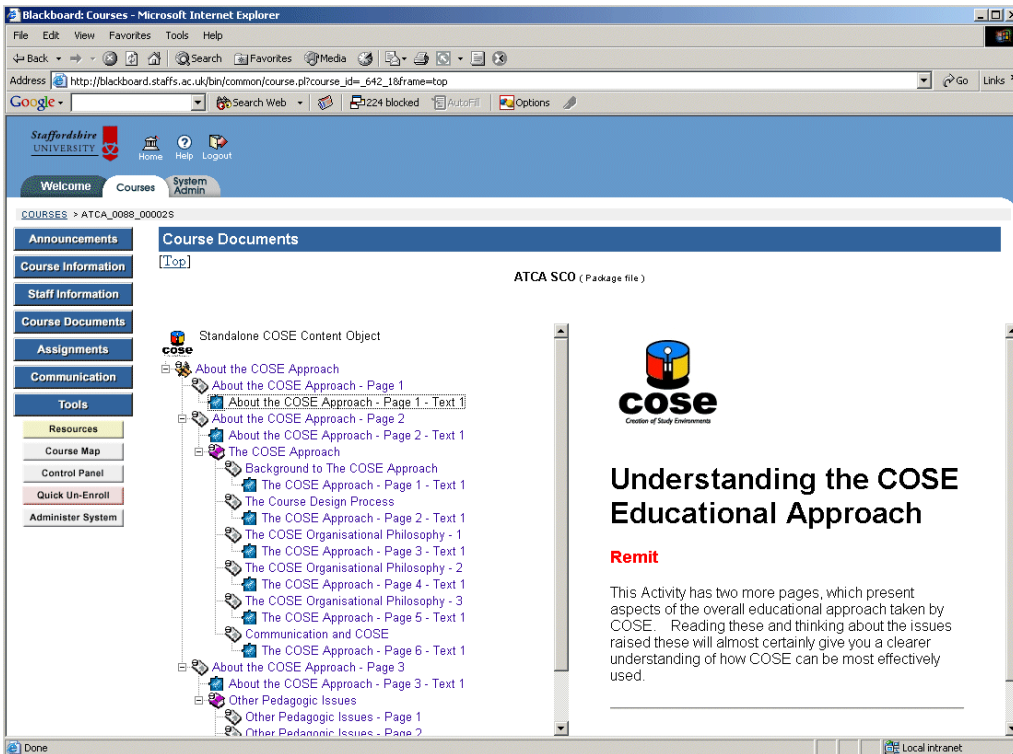


Figure 9. SCO content package imported to Blackboard

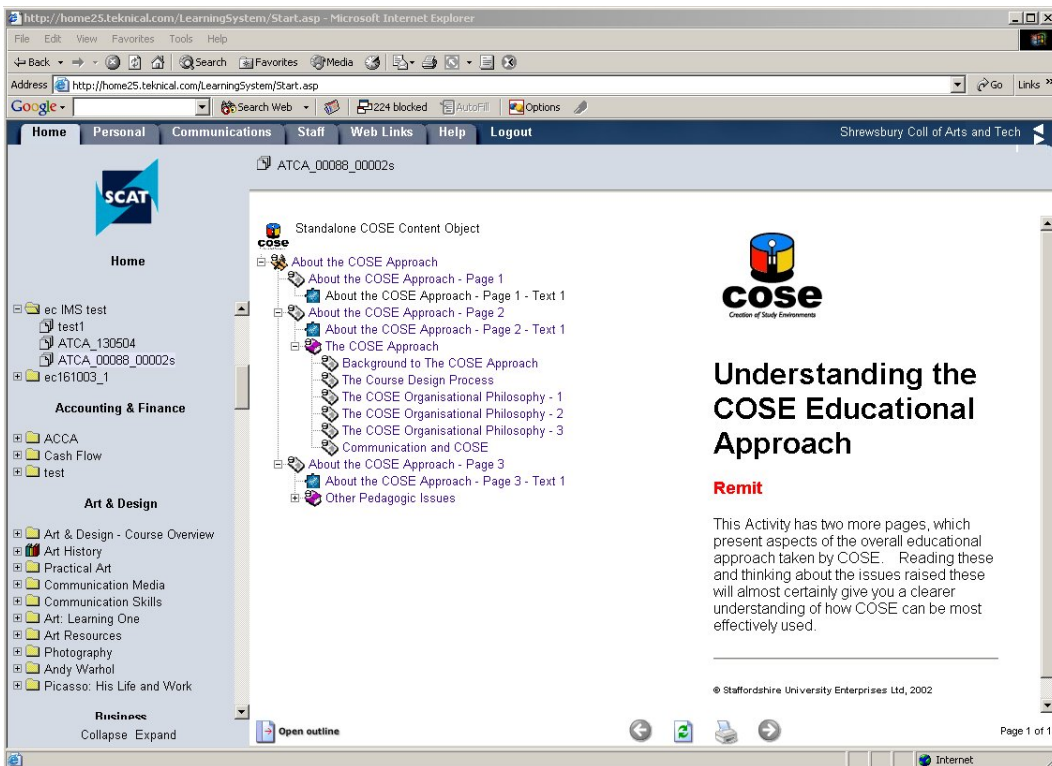


Figure 10. SCO content package imported to TekniCAL's Virtual Campus

select the resource as a zip file or top level IMS (CP) directory. The SCO is navigated by toggling nodes of the navigation frame: the SCO doesn't respond to the arrows of the Virtual Campus browser (select resource of relevant subject under the Home tab of the user interface).

RELOAD

Figure 11 shows the same SCO again imported to the popular Reload (Reusable eLearning Object Authoring and Delivery / Metadata and Content Packaging) Editor version 1.1.3. SCOs aren't properly rendered in the package viewer being revised in version 1.2. There may be a solution in displaying the content associated with a given node in a separate window (see the Intrallect Content Repository), else another packaging option may be introduced in COSE to produce more generic packages with fixed navigation / organisation of the content.

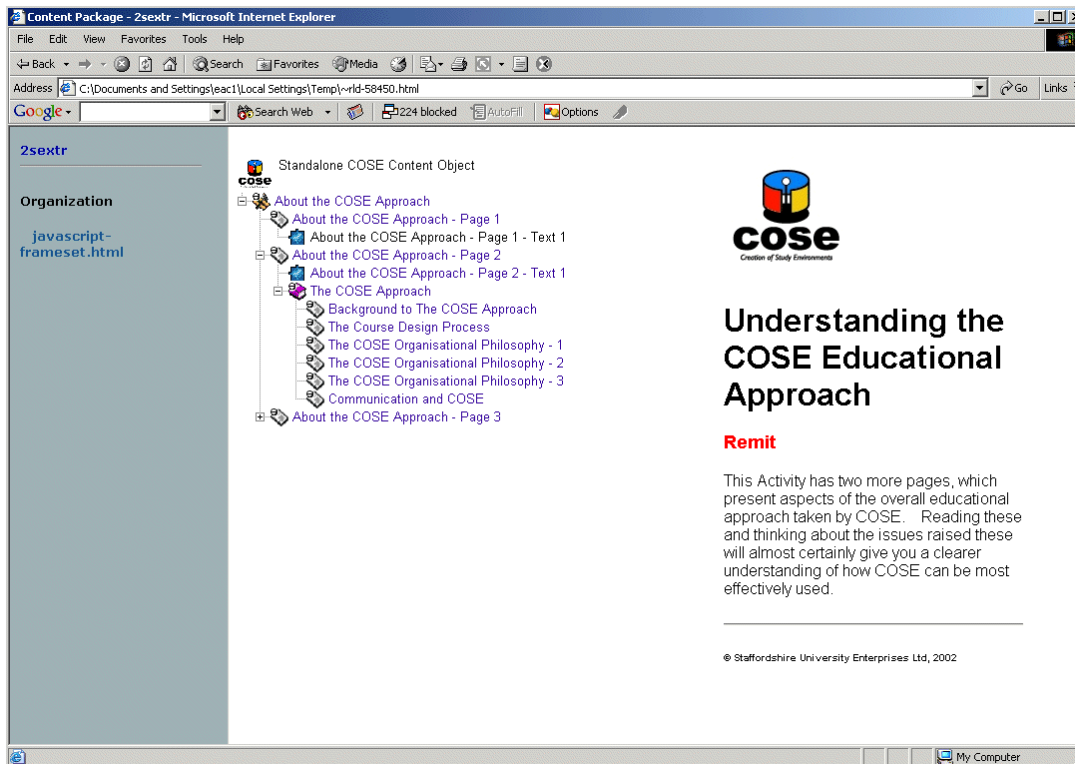


Figure 11. SCO content package imported to the Reload Editor version 1.1.3

Intralibrary

Figures 12a and 12b show an earlier version of the the same SCO imported to Intrallect's Content Repository, this service being offered as part of the Jorum project. Displaying the content associated with a given node in a separate window is a slight modification of normal SCO behaviour tested before the COSE 2.1 release. The resource view of the content shown in Figure 12a presumes selection of the javascript-frameset.html resource in the left-hand frame. The package can be displayed using the organisational view, shown for RELOAD above.

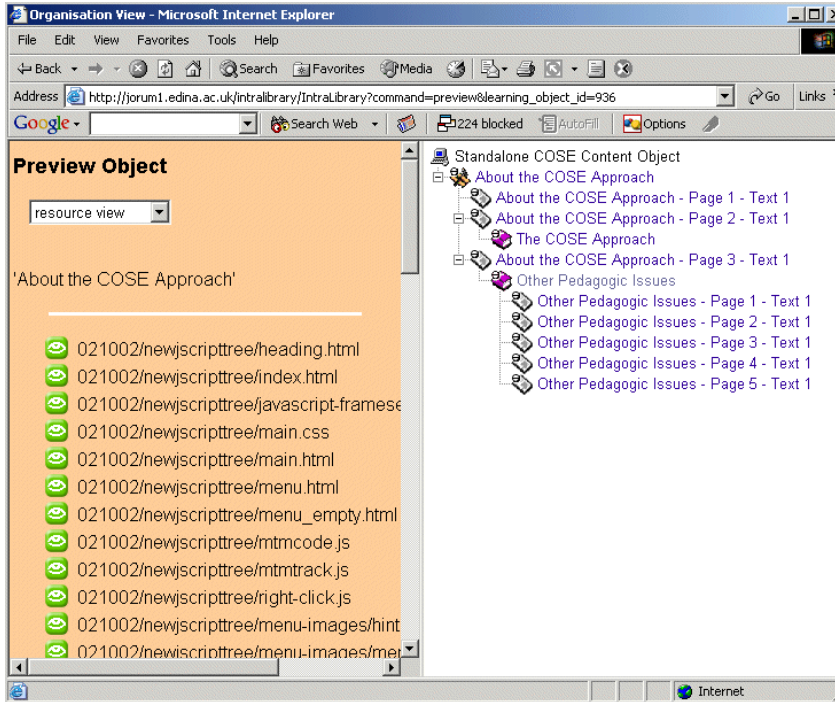


Figure 12a. Earlier version of the SCO content package imported to Intrallect's content repository Intralibrary version 2.0: resource view

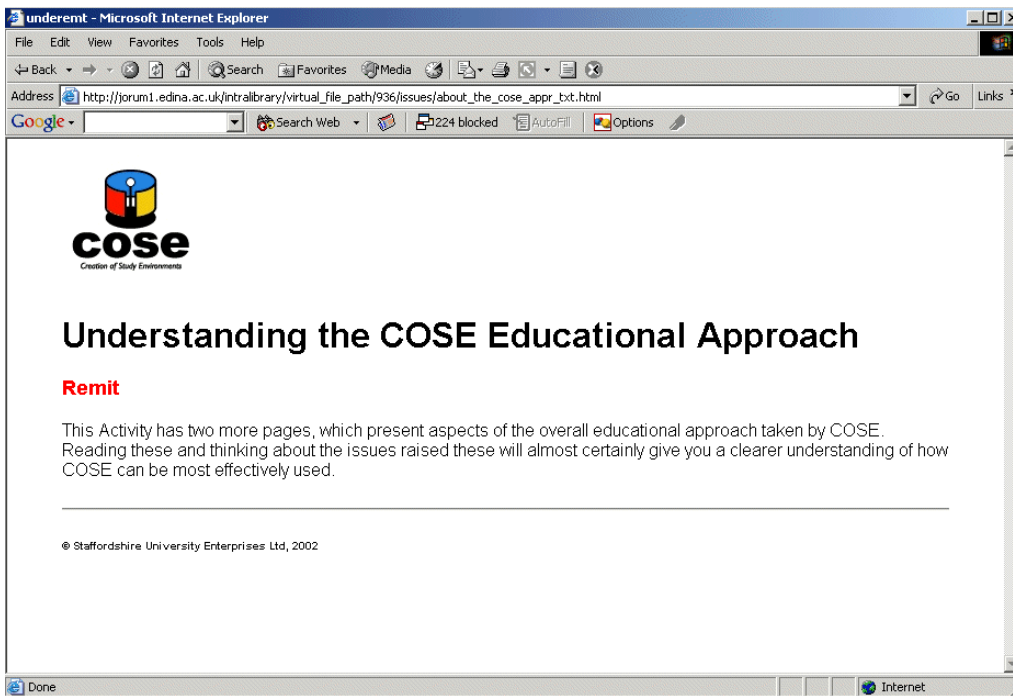


Figure 12b. Earlier version of the SCO content package imported to Intrallect's content repository Intralibrary version 2.0: selected node is displayed in new window

WebCT

Figure 13 shows an earlier version of a different SCO imported to WebCT Campus Edition version 4.1 (courtesy of Peter Travis, Henley College, Coventry).

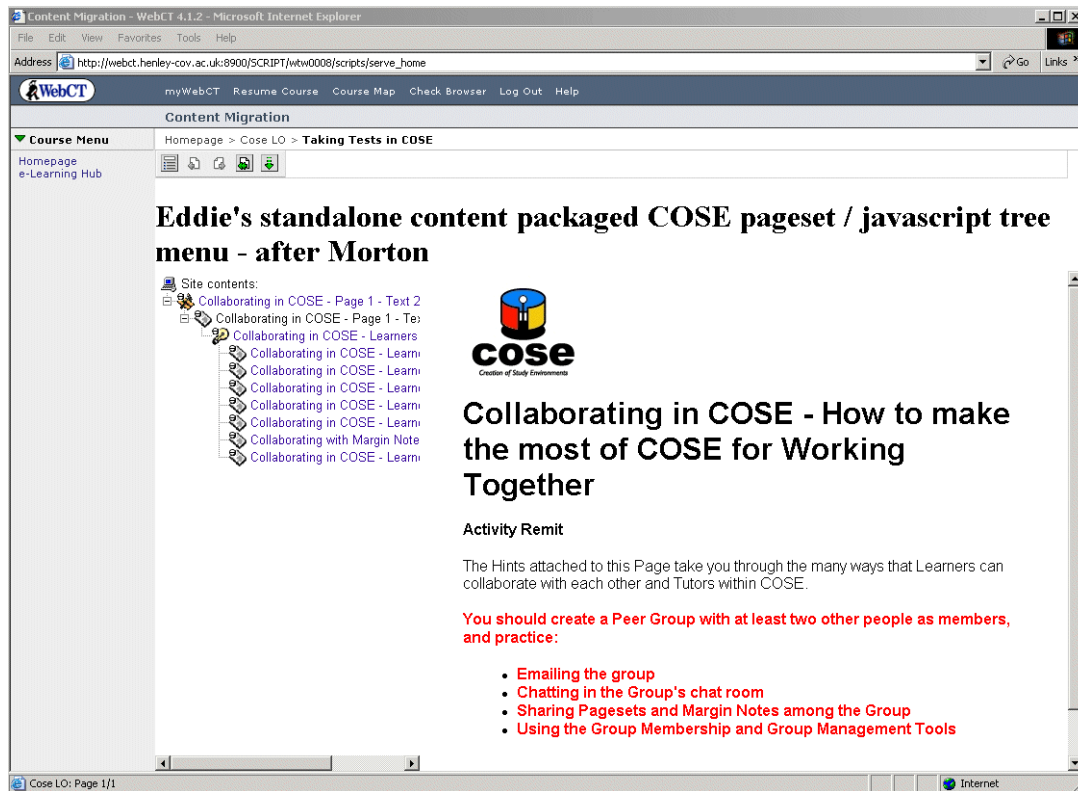


Figure 13. SCO content package imported to WebCT's Campus Edition version 4.1

Learnwise version 2 and Learnwise Olympus

Figures 14a and 14b show an early version of a more complex SCO imported to Learnwise Olympus which was also shown to be successfully imported to Learnwise version 2.0. The view of the content opening in a new window, shown in Figure 14b, presumes selection of the javascript-frameset.html page from the folder where the package was imported to. Further testing is required to determine whether the package can be displayed using the default organisational view, shown for RELOAD above.

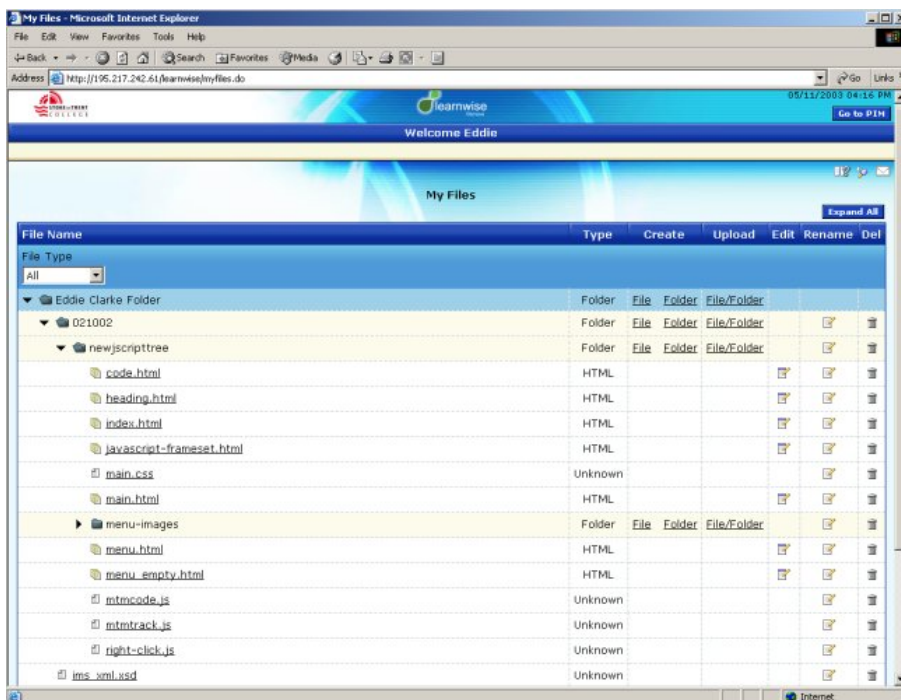


Figure 14a. SCO content package imported to Learnwise Olympus: view of package folder

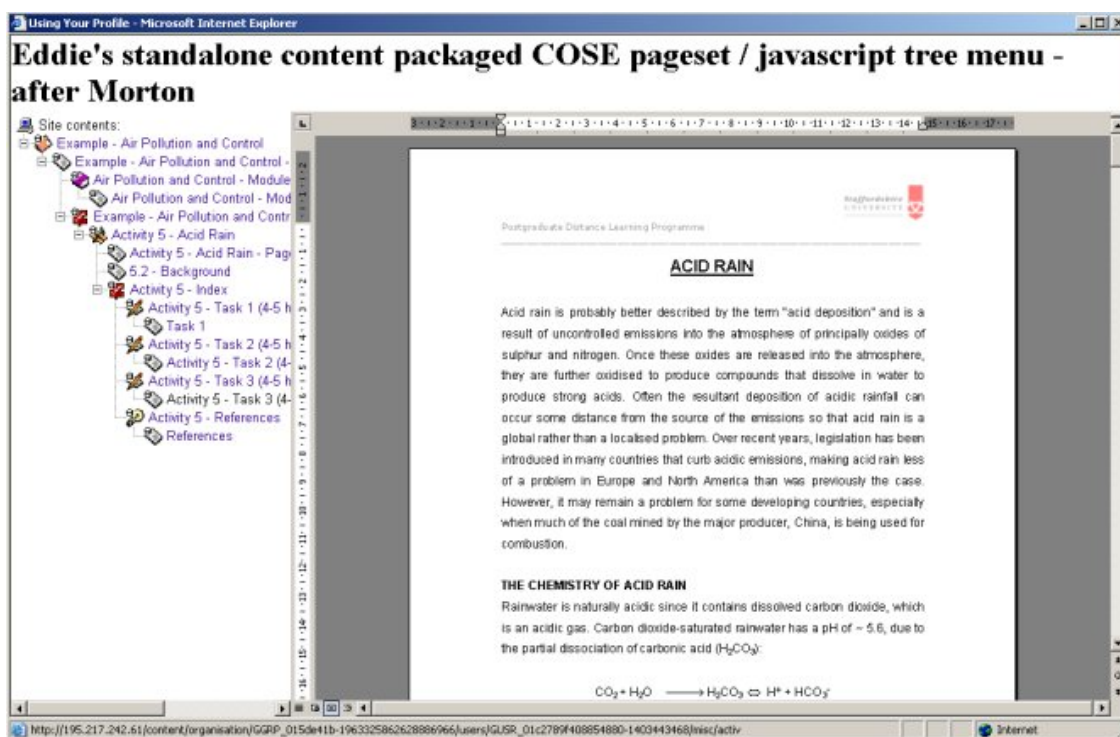


Figure 14b. SCO content package imported to Learnwise Olympus: view of SCO in new window

New Accessibility Enhancements

This section describes our approach to making COSE accessible by users of assistive software. Changes made to COSE in order to achieve this are documented, along with progress we have made, the problems we have encountered and what remains to be done.

Feedback

Development and testing of the accessibility modifications is ongoing so any feedback from users on the software or documentation will be gratefully received. Requests for enhancement and bug reports can be submitted using the COSE Bugzilla installation at <http://cose.staffs.ac.uk/bugzilla> or send an email to cose-support@staffs.ac.uk with your comments.

Required Software

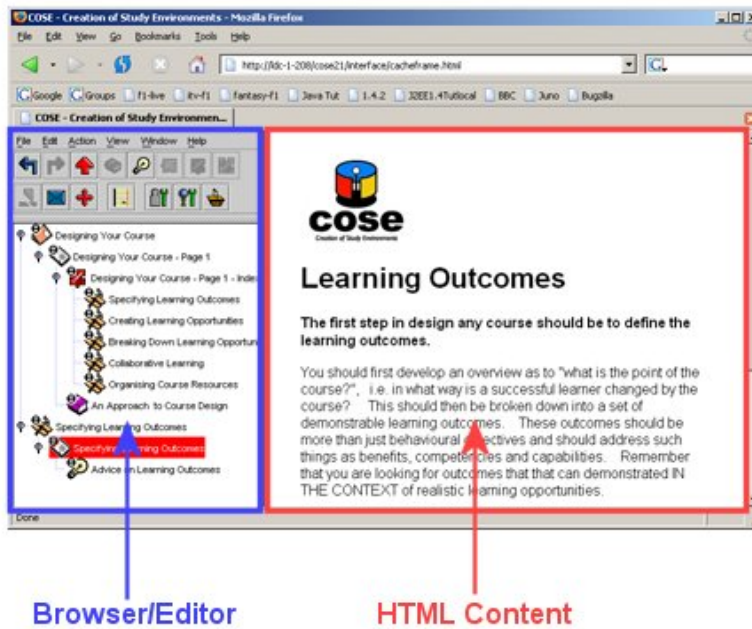
To use the accessibility features of COSE you will need to download and install the Java Access Bridge <http://java.sun.com/products/accessbridge/> to allow assistive software to interrogate the COSE applet. None of the features described below will work without it.

Overview of the COSE User Interface

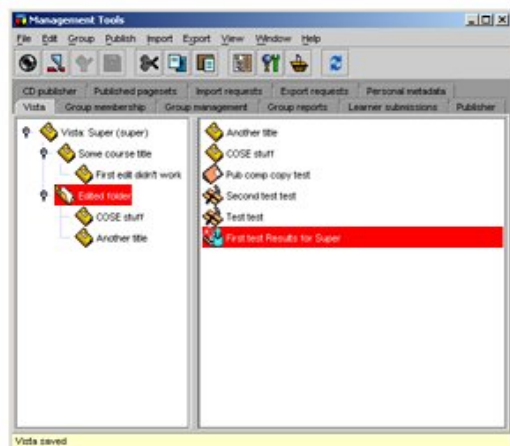
The COSE client is a Java applet which is downloaded and viewed using a web browser running the Sun Java Plug-in. The COSE applet contains a suite of tools to allow the user to manage COSE content and COSE groups and users.

COSE does not provide a tool for the creation of the HTML pages which form the basis of COSE content. The user is free to use an application of their choice to create the HTML pages. The user then imports the HTML pages and uses the COSE Editor to link them together in a structure to form a COSE pageset.

The COSE Browser allows the user to browse COSE pagesets. It and the Editor appear in the left hand frame of the web browser. The COSE Browser displays the structure of the pageset in the form of a tree structure. As the user navigates through the pageset, the HTML content of the currently selected page is displayed in the right hand frame.



The COSE Management Tools appear in a separate window and allow the user to manage groups and users and organise their work. The Search Tools window allows the user to perform searches and the Basket window allows users to store links to useful resources. The Management Tools window is shown below.



Our Approach

Given the architecture of the COSE client detailed above, it is apparent that to make COSE accessible both the content and the user interface need to be accessible. Our approach has two strands of development. The first is to follow Java accessibility guidelines (see below) in the creation of the user interface. The second is to promote the creation of accessible HTML pages by creators of COSE content so that the pagesets created with them will also be accessible.

Guidelines

The relevant guidelines and documentation used to guide the COSE accessibility work are:

W3C WAI guidelines

Of the World Wide Web Consortium Web Accessibility Initiative guidelines considered <http://www.w3.org/WAI/Resources/#gl>, only the Web Content Accessibility guidelines are of direct relevance. These will form the basis of the instructional content we are beginning to develop to assist COSE content creators to create HTML pages that are accessible.

With respect to the other WAI guidelines, the COSE client is neither an authoring tool nor a user agent. These roles are performed by the various HTML authoring software used by the COSE content creators and the web browser (with Java Plug-in) respectively.

Java guidelines

The two Java guideline documents most used to plan the implementation of the accessibility modifications were:

"IBM Guidelines for Writing Accessible Applications Using 100% Pure Java"
<http://www-306.ibm.com/able/guidelines/java/snsjavag.html>. This document has been around for a long time. It was the most informative and formed the basis of the plan along with the associated checklist <http://www-306.ibm.com/able/guidelines/java/accessjava.html> .

The "Developing Accessible JFC Applications"
<http://www.sun.com/access/developers/developing-accessible-apps/>. This document gave a different view of the use of tooltips rather than accessible descriptions to reduce the work required when changing or localizing the user interface.

Implementation

The IBM checklist is reproduced below together with an indication of the progress made and comments on the work done or remaining to be done.

IBM Java accessibility checklist - version 3.1			
1	<i>Keyboard Access.</i>	Yes No Planned N/A	Comments
1.1	Provide keyboard equivalents for all actions.	Yes	
1.2	Do not interfere with keyboard accessibility features built into the operating system.	Yes	Mnemonics and accelerators have been chosen to avoid operating system keyboard accessibility features.

2	<i>Object information.</i>	Yes No Planned N/A	Comments
2.1	Implement the Java Accessibility API by: - using the Java Foundation Classes (JFC) / Swing components and/or - following the guidelines for " Building Custom Components " when extending the Java Foundation Classes and when implementing the Java Accessibility API on custom components.	Yes	All COSE user interface components are Swing components or are extended from Swing components.
2.2	Set the focus.	Planned	This task has yet to be addressed. Currently, the user interface components need to be navigated to before input can be entered.
2.3	Set the name on all components and include the description on icons and graphics. If an image is used to identify programmatic elements, the meaning of the image must be consistent throughout the application.	Yes	Accessible names have been added to all components. Tooltips have been added wherever it is appropriate to do so and accessible descriptions have been added as needed for situations where a tooltip is not appropriate.
2.4	Associate labels with controls, objects, and icons.	Yes	Labels have been associated with text fields where appropriate using the <code>setLabelFor()</code> method. We have not yet implemented labelling using the Java 1.3 <code>AccessibleRelation</code> class which may well give us more fruitful results with screen reader software.
3	<i>Sound and multimedia.</i>	Yes No	Comments

		Planned N/A	
3.1	Provide an option to display a visual cue for all audio alerts .	N/A	There are no audio alerts in the COSE user interface.
3.2	Provide accessible alternatives to significant audio and video .	N/A	There is no audio or video in the COSE user interface.
3.3	Provide an option to adjust the volume .	N/A	
4	<i>Display.</i>	Yes No Planned N/A	Comments
4.1	Use color as an enhancement, not as the only way to convey information or indicate an action.	Yes	Any component state information that used to be conveyed using colour in COSE 1.x releases is now conveyed by an icon, allowing accessible names and descriptions of the state to be associated with the icon.
4.2	Support system settings for high contrast for all user interface controls and client area content.	Planned	At the moment, COSE uses the Java Look and Feel. We plan to allow the user to choose the Look and Feel which will enable the user to select Windows Look and Feel and inherit high contrast settings.
4.3	When color customization is supported, provide a variety of color selections capable of producing a range of contrast levels.	N/A	Colour customization is not supported in COSE.

4.4	Support system settings for size, font and color for all user interface controls.	Planned	COSE uses the Java Look and Feel and a modified version of the DefaultMetalTheme which overrides the default bold-style fonts in favour of plain-style. We do not provide a facility for the user to change the Look and Feel or Theme at the moment. This will be investigated with a view to implementing it in a future 2.x release.
4.5	Provide an option to display animation in a non-animated presentation mode.	N/A	The COSE user interface has no animation.
5	Timing.	Yes No Planned N/A	Comments
5.1	Provide an option to adjust timed responses or allow the instruction to persist.	N/A	The COSE user interface has no timed responses.
5.2	Avoid the use of blinking text, objects, or other elements.	Yes	
6	Documentation.	Yes No Planned N/A	Comments
6.1	Provide documentation in an accessible format.	Planned	We will make the COSE manual PDF file more accessible for COSE version 2.1 and will make the manual available in HTML format at the same time or shortly thereafter. It is planned to test and improve the accessibility of both formats over time.

6.2	Provide documentation on all accessibility features , including keyboard access, as part of the regular product documentation.	Yes	This is included as of COSE version 2.1.
7	<i>Verify accessibility.</i>	Yes No Planned N/A	Comments
7.1	Test for accessibility using available tools.	Yes	The COSE user interface has been tested by using the keyboard only and with JAWS and ZoomText screen reader software. We plan to expand the testing starting with Java accessibility testing tools.

Testing and Known Issues

In testing, the JAWS Virtual Viewer feature conflicted with attempts to invoke the accessible description associated with a component. Upon pressing Insert + F1, JAWS should read out the accessible description for the currently selected component. It failed to read accessible descriptions in version 5 and would only allow invoking of accessible descriptions in version 4.5 if the user tabbed to the JAWS application and then back to COSE. By default it responded to Insert + F1 by displaying the JAWS Virtual Viewer window.

When COSE is first launched, because the COSE applet is signed, a Java security warning dialog box is displayed (see below) and the system then waits for the user to acknowledge it. The screen reader software fails to read the security dialog so the user has no way of knowing that the system is waiting for a response. A workaround is to hit the tab key twice to move the focus to the Always key then hit the return key to acknowledge the dialog and continue loading COSE. Selecting always will mean that the dialog does not appear again.



The COSE Browser/Editor has a focussing issue. The user can select the applet frame in the web browser without using a mouse but the applet appears not to receive keyboard focus. None of the mnemonics or accelerators work and mouseless navigation also fails to have any effect. Once the applet is selected with a mouse or mouse equivalent, e.g. MouseKeys, then the applet features work.

Feedback on any other problems or issues will be gratefully received. Please see the Feedback section above for details of how to report issues.

Using COSE

The user can navigate COSE using a combination of operating system accessibility features, assistive software and mouseless navigation features built into the web browser, the Java Runtime Environment and COSE itself.

Operating System Features

These are features built into the operating system or desktop environment which allow mouseless operation of the environment, etc. See the links below for information about what features are available in each operating system.

Windows Accessibility

<http://www.microsoft.com/enable/?gssnb=1>

Linux/UNIX/Solaris Accessibility

Linux Accessibility Resource Site (LARS)

<http://lars.atrc.utoronto.ca/>

Disability access to GNOME

<http://developer.gnome.org/projects/gap/>

The KDE Accessibility Project

<http://accessibility.kde.org/>

Apple Accessibility

<http://www.apple.com/accessibility/>

Assistive Software

Two screen reader software applications have been used in the testing of COSE:

- JAWS <http://www.freedomscientific.com/>
- ZoomText <http://www.aisquared.com/>

Screen readers should read out the name of each component as the interface is navigated, allowing the user to hear where they are in the interface. A more detailed description of the purpose of each component should be read out if the user requests it via a keystroke combination. As described in the Testing section above, these features work in some versions of the screen reader software and only partially work in others. Work is continuing to find out why this is the case.

Mouseless Navigation

The lists of key bindings referenced below show the keys that the user can use to navigate and perform actions without using a mouse. The lists are in no way exhaustive and are offered by way of information rather than being definitive lists of all possible bindings. We aim to refine and simplify this information over time.

Web Browser Key Bindings

This link gives a comparison list of common Firefox, Internet Explorer and Opera web browser key bindings.

<http://texturizer.net/firefox/keyboard.html>

Java Key Bindings

The Java Key Bindings referenced by the link below are organized by the type of component they relate to. Each name is the Java class name of the component. The names are reasonably indicative of the nature of the component, e.g. JApplet for the applet, JButton for buttons. Clicking on a name brings up the relevant keystrokes and actions for the component.

<http://java.sun.com/j2se/1.3/docs/api/javaw/swing/doc-files/Key-Index.html>

COSE Key Bindings

Mnemonics (access keys) and accelerators (shortcut keys) have been added to COSE menus, menu items and buttons to enable easy navigation and invocation of the menu items and action buttons. The mnemonics and accelerators are listed below.

Mnemonics

Management Tools

Alt-F = File Dropdown menu

N = New

P = Project

A = Activity

T = Task

H = Hint

E = Theory

S = Page set

F = Folder

G = Group

B = Open in Browser

E = Open in Editor

H = Share

L = Clear Shares

U = Submit

A = for Assessment

P = for Publication
X = for Export
A = Copy as
H = Hint
E = Theory
S = Save
C = Close

Alt-E = Edit Dropdown menu

T = Cut
C = Copy
P = Paste
D = Delete
R = Preferences

Alt-G = Group Dropdown menu

T = Report
S = Stop
A = Add
R = Remove
P = Promote
D = Demote
E = Send email
M = Show more groups
F = Show fewer groups

Alt-P = Publish Dropdown menu

P = Publish
R = Reject
Q = Request CD

Alt-I = Import Dropdown menu

A = Add
R = Remove
S = Submit
U = Upload
P = Unpack
I = Install
E = Reject

Alt-X = Export Dropdown menu

M = Metadata
R = Reject
S = Submit
X = XML
P = Package

Alt-V = View Dropdown menu

- L = List
- D = Details
- S = Sort
- C = Clear
- R = Refresh

Alt-W = Windows Dropdown menu

- B = Browser/editor
- S = Search Tools
- A = Basket

Alt-H = Help Dropdown menu

- T = Tool F1
- I = Icons
- A = About

Browser

Alt-F = File Dropdown menu

- E = Open in Editor

Alt-E = Edit Dropdown menu

- T = Cut
- C = Copy
- P = Paste
- R = Preferences

Alt-A = Action dropdown menu

- N = Next
- P = Previous
- R = Previous Page set

Alt-V = View dropdown menu

- H = Hint
- O = Theory
- D = Media Object
- I = Index
- T = Test
- A = Margin Notes

Alt-W = Windows dropdown menu

- M = Management Tools
- S = Search Tools
- A = Basket

E = Email
R = Resize

Alt-H = Help dropdown menu

T = Tool
I = Icons
A = About

Editor

Alt-F = File dropdown menu

B = Open In Browser
U = Upload File
P = Properties
R = Reload
S = Save

Alt-E = Edit dropdown menu

T = Cut
C = Copy
P = Paste
U = Move Up
N = Move Down
M = Mark as Default Page
D = Delete
R = Preferences

Alt-I = Insert dropdown menu

P = Page
T = Text Component
N = Index Component
M = Multiple Choice Test

Alt-W = Window dropdown menu

M = Management Tools
S = Search Tools
A = Basket
L = Email
R = Resize

Alt-H = Help dropdown menu

T = Tool
I = Icons
A = About

Search Tools

Alt-F = File dropdown menu

- B = Open in Browser
- E = Open in Editor
- C = Close

Alt-E = Edit dropdown menu

- T = Cut
- C = Copy
- D = Delete
- R = Preferences

Alt-A = Action dropdown menu

- A = Start Search
- O = Stop Search
- L = Add List
- D = Add Selection

Alt-V = View dropdown menu

- L = List
- D = Details
- S = Sort
- T = Type
- N = Name
- R = Refresh
- E = Reset

Alt-W = Window dropdown menu

- B = Browser/Editor
- M = Management Tools
- A = Basket

Alt-H = Help dropdown menu

- T = Tool
- I = Icons
- A = About

Basket

Alt-F = File dropdown menu

- N = New Folder
- B = Open in Browser
- E = Open in Editor
- S = Save
- C = Close

Alt-E = Edit Dropdown menu

T = Cut
C = Copy
P = Paste
R = Preferences

Alt-V = View Dropdown menu

L = List
D = Details
S = Sort
T = Type
N = Name
R = Refresh

Alt-W = Window dropdown menu

B = Browser/Editor
M = Management Tools
S = Search Tools

Alt-H = Help dropdown menu

T = Tool
I = Icons
A = About

Accelerators

Management Tools

Open in Browser	- Ctrl-B
Open in Editor	- Ctrl-E
Share selected item	- Ctrl-H
Save	- Ctrl-S
Show Browser/Editor	- Ctrl-W
Show Search Tools	- Ctrl-T
Show Basket	- Ctrl-A
Refresh current section	- F5
Show Help	- F1

Browser

Previous	- Up Arrow
Next	- Down Arrow
Up	- Ctrl-
Show Theory	- Ctrl-O
Show Hint	- Ctrl-H
Show Media	- Ctrl-D
Show Index	- Ctrl-F
Show Test	- Ctrl-Q
Show Email Window	- Ctrl-L

Show Resize Tool	- Ctrl-R
Show Margin Notes	- Ctrl-G
Show Management Tools	- Ctrl-M
Show Search Tools	- Ctrl-T
Show Basket	- Ctrl-K

Editor

Save	- Ctrl-S
Upload	- Ctrl-U
Up	- Arrow Up
Down	- Arrow Down
Delete	- Ctrl-D
Show Browser	- Ctrl-B
Show Email Window	- Ctrl-L
Show Resize Tool	- Ctrl-R
Show Component Editor	- Ctrl-P
Reload	- Ctrl-E
Show Management Tools	- Ctrl-M
Show Search Tools	- Ctrl-T
Show Basket	- Ctrl-K
Show Help	- F1

Search Tools

Open in Browser	- Ctrl-B
Open in Editor	- Ctrl-E
Cut	- Ctrl-X
Copy	- Ctrl-C
Help	- F1

Basket

Open in Browser	- Ctrl-B
Open in Editor	- Ctrl-E
Cut	- Ctrl-X
Copy	- Ctrl-C
Paste	- Ctrl-V
Save	- Ctrl-S
Help	- F1

Appendix 1

COSE Content Packaging and Interchange

Metadata is used in COSE to describe Pagesets in the context of the Export (content packaging) routine, so that when a pageset is exported to another system as a pack, a set of descriptors is carried along with it (see also Appendix F of the COSE manual - Metadata used in COSE). When unpacked at the host system, the descriptors can be used to help identify a resource as being particularly useful or interesting according to when or where it was generated, by whom, according to how it is classified in some well-known classification system or by a number of other criteria. This metadata consists not of arbitrary data of our own device but rather is consistent with data set agreed upon by an international consortium of educational and commercial enterprises who have been working for a number of years now to achieve interoperability between virtual learning environments, otherwise known as instructional management systems (IMS). [For further details about how COSE is using IMS and other specifications to achieve interoperability project see the COSE website <http://www.staffs.ac.uk/cose/>].

Such a set of descriptors would of course also be useful in searching for resources within a COSE system (and, using related standards, for other purposes, such as maintaining student profiles). COSE development continues our work to incorporate such features in a future release. The use of metadata in the context of exporting resources is described here. Content can be exported (packaged) either for exchange between COSE systems (native / 2.05 style) or as standalone content objects (SCOs) which were described earlier.

Personal Metadata

This metadata has to be input by different users of the system according to the process outlined in the figure below. Some metadata relates to the people involved in the generation / authorship and publication of a resource. Rather than having to enter these details each time a Pageset is to be exported, an author or 'publisher' [in this context, this is the person who approves the export; cf. pageset publication] can store values for their personal metadata in a system file. It is the Personal Metadata tab in the COSE Management Tools which allows them to do this. Other metadata is input as various input dialogue boxes appear during the process of exporting, as described in **Exporting Pagesets**.

The screenshot shows the 'Management Tools' application window. The 'Personal metadata' tab is active, displaying a form with the following fields: Surname, First name or Initial, Email address, Organisation, Address line 1, Address line 2, Address line 3, Town or City, Country or State, Country, Postcode or Zip code, Department, School, University or College, and Module or Award. All fields are currently empty. At the bottom of the form are 'Save' and 'Clear' buttons. A yellow status bar at the very bottom of the window displays the text 'Profile data loaded'.

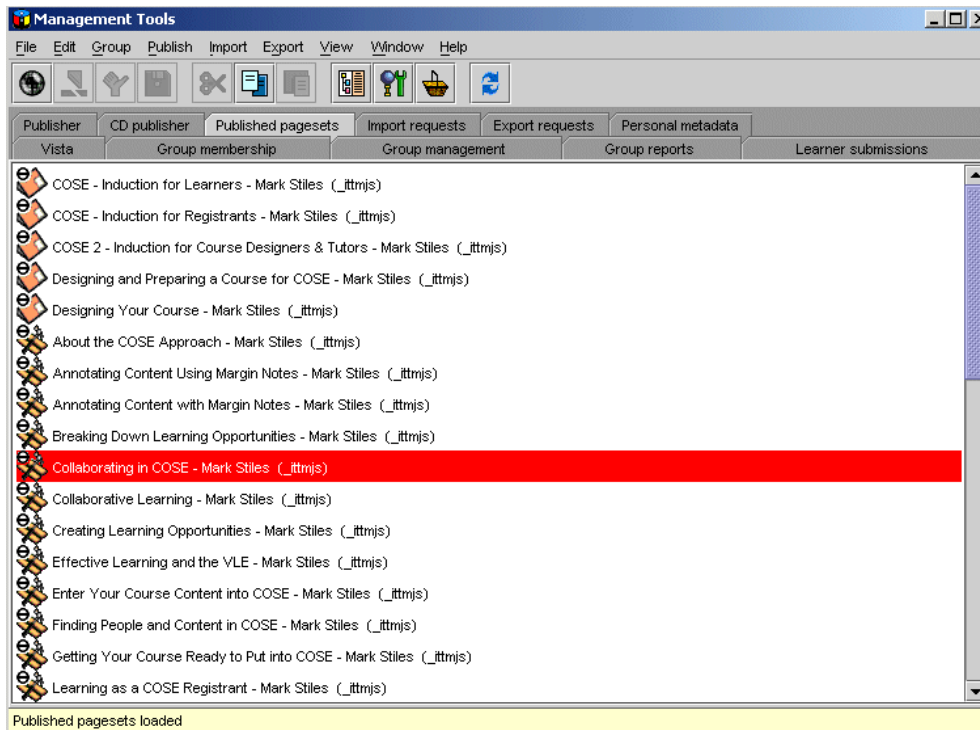
On first view, the Personal Metadata tab shows a number of labelled, editable textfields, which will contain no data; the status bar displays the message 'Profile data loaded', though none has been entered at this point. The user enters appropriate values in the data fields and clicks on the button labelled Save data to server. The status bar then displays the message 'Profile data saved to server', as illustrated. The Metadata shown is personal to the current COSE user.

The screenshot shows the 'Management Tools' application window with the 'Personal metadata' tab active. The form fields are now filled with the following data: Surname: Stiles; First name or Initial: Mark; Email address: m.j.stiles@staffs.ac.uk; Organisation: Learning Development Centre; Address line 1: Octagon Bldg; Address line 2: Staffordshire University; Address line 3: Beaconside; Town or City: Stafford; Country or State: Staffs; Country: UK; Postcode or Zip code: ST18 0AD; Department: ; School: ; University or College: Staffordshire University; Module or Award: . The 'Save' and 'Clear' buttons are visible at the bottom of the form. A yellow status bar at the bottom of the window displays the text 'Profile data saved to server'.

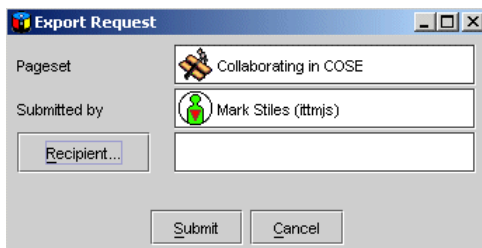
The procedure is described in detail below:

Author interface

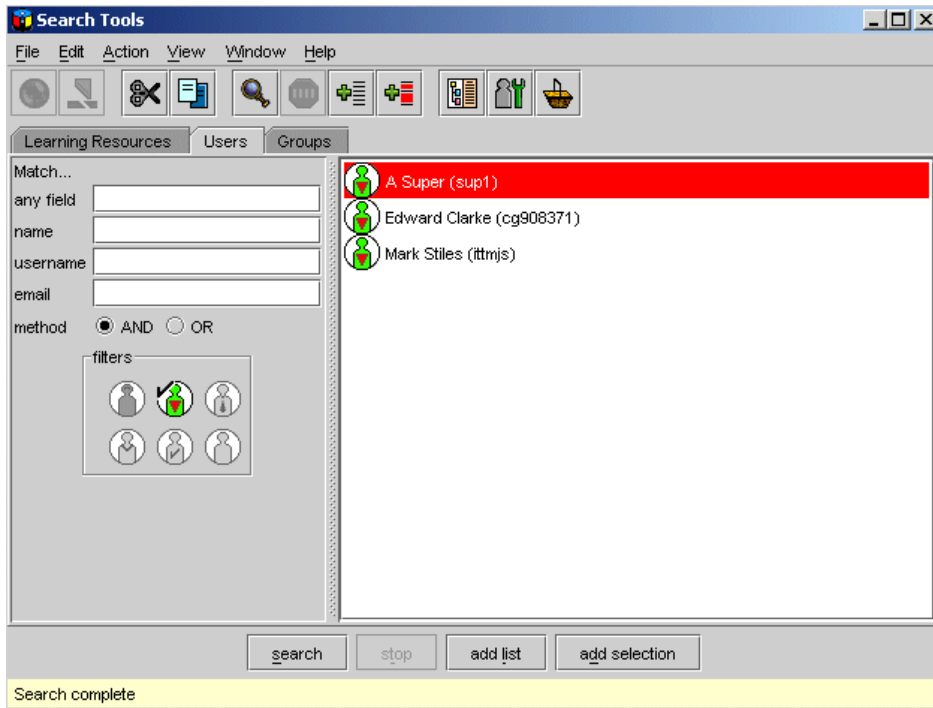
From the **Published Pagesets** tab an author submits a request for the export of a particular Pageset to a publisher. The next diagram shows the Published Pagesets tab as viewed by a Supertutor with the username ittmjs. The Pageset "Colaborating in COSE", authored by that user, has been selected.



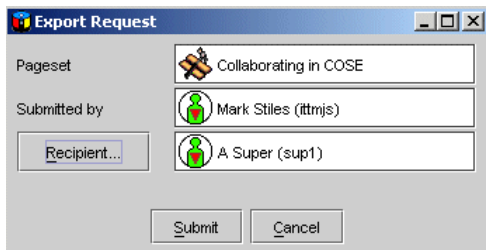
Selecting **Submit | for Export** from the **File** menu brings up a submission dialog frame, shown below, requiring the author to select a Supertutor as publisher, to whom they submit the export request.



Clicking the **Recipient ..** button brings up the Users tab of the Search Tools.



A Super (sup1) is selected as Publisher in this case, and inserted into the dialogue box by clicking the **add selection** button.



Clicking the **Submit** button now opens the **Metadata for Pageset** frame. The author is expected to fill in the entries for a number of different categories of metadata as shown under the different tabs below. Some of these will already be filled in with default values. For example, in the first diagram, the title is taken from the Title given to the Pageset by its author. The second diagram shows Lifecycle metadata which takes values submitted to the system by means of the Personal Metadata tool. The meaning of the various categories of metadata and any particular recommended vocabularies used in entering these metadata fields are described in Appendix F of the COSE manual - Metadata used in COSE.

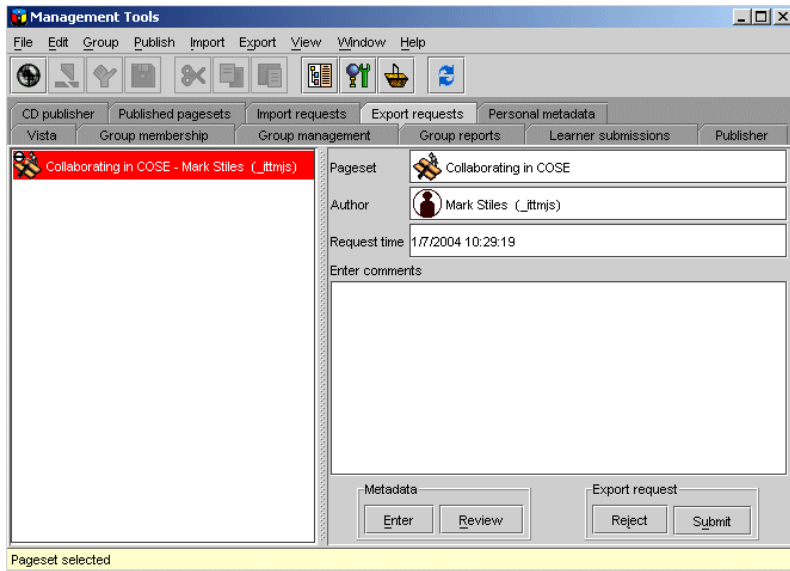
The figure displays four screenshots of the 'Metadata for PageSet' dialog box, arranged in a 2x2 grid. Each window has a title bar with the text 'Metadata for PageSet' and a close button (X). The dialog box is divided into four tabs: 'General', 'Lifecycle', 'Relation', and 'Classification'.

- Top-left window (General tab):** Describes certain general properties of pageset. Fields include Title: 'Collaborating in COSE' and Description: 'group working in COSE'.
- Top-right window (Author Information tab):** Describes yourself, as Author. Fields include Surname: 'Stiles', Forenames: 'Mark', Email: 'm.j.stiles@staffs.ac.uk', Address 1: 'Octagon Bldg', Address 2: 'Staffordshire University', Address 3: 'Beaconside', Town / City: 'Stafford', County / State: 'Staffordshire', Country: 'UK', and Postcode: 'ST18 0AD'.
- Bottom-left window (Relationship tab):** Describes how pageset is typically a component of a module/course. Fields include Relationship: 'isPartOf', Module or course code, Department: 'Learning Development', School, and University or college: 'Staffordshire University'.
- Bottom-right window (Classification tab):** Describes pageset according to some (formal) classification scheme. Fields include Subject: 'Virtual Learning Environments' and Keywords: 'Sharing Unpublished Content; Collaborating; Peer Groups; Email; Chat; Margin Notes'.

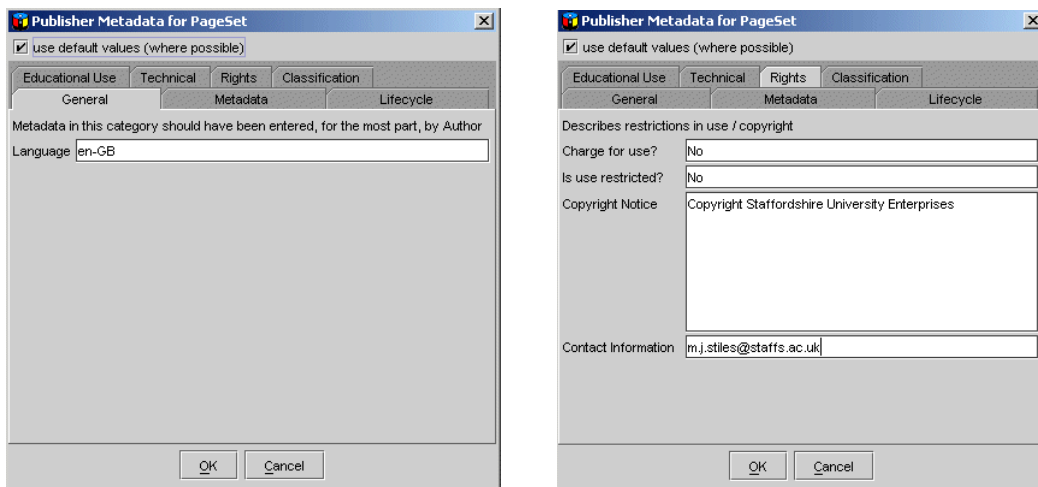
Clicking the OK button closes the metadata and submissions windows. A message is displayed on the status bar of the Management Tools to say that the Export Request was submitted.

Publisher interface

The image below shows the Export Requests tab of the publisher's interface, the publisher in this case being A Super. The pane on the left lists requests for Pagesets to be exported. As A Super was the recipient nominated in the previous section, the request made by user itmjs for the export of the Pageset "Collaborating in COSE" is listed. Selecting the entry lists details of this submission request on the right. There is also a area for comments which will be sent to the author by email when the request is forwarded or rejected. This may specify, for example, that either one, or both, of the packaging options should be used in processing the request.



Clicking the **Enter** button opens the Publisher Metadata for Pageset frame. As shown, a number of the tabs in the frame correspond to the same categories as the author metadata, while others are different. Again, the meaning of the various categories of metadata and any particular recommended vocabularies used in entering these metadata fields are described in Appendix F - Metadata used in COSE.



The division of labour in supplying metadata is designed to make the process not too onerous for any individual. The publisher inputs the metadata for which they are responsible, using default values as far as they can be supplied. By ticking the box **use default values**, many of the fields in each tabbed frame are filled with values. Lifecycle metadata takes values submitted to the system via the publisher's Personal Metadata tool. Clicking the **OK** button saves the metadata to system files and closes the frame. By

clicking the **Review** button, all metadata associated with that Pageset is displayed, as shown next

The figure displays four screenshots of the 'Metadata for PageSet' dialog box, showing different tabs and their content:

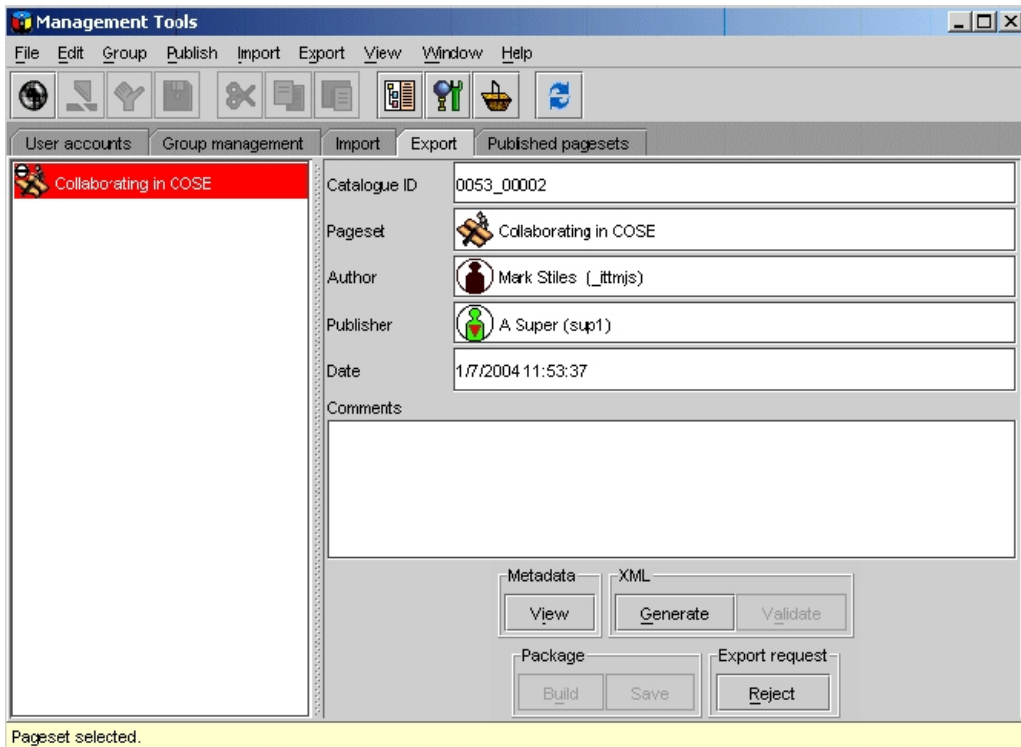
- General Tab:** Shows fields for Language (en-GB), Title (Collaborating in COSE), and Description (group working in COSE).
- Publisher details Tab:** Shows fields for Surname (Super), Forenames (A), Email (e.a.clarke@staffs.ac.uk), and Organisation (Learning Development Centre).
- Author details Tab:** Shows fields for Surname (Stiles), Forenames (Mark), Email (m.j.stiles@staffs.ac.uk), and Address (Octagon Bldg, Beaconside Campus, Staffordshire University, Stafford, Staffs, UK, ST18 0AD).
- Classification Tab:** Shows fields for Relationship (isPartOf), Module or course code, Department (Learning Development), School, University or college (Staffordshire University), Subject (Virtual Learning Environments), Keywords (Sharing Unpublished Content, Collaborating, Peer Groups; Email, Chat, Margin Notes), Scheme (Dewey Decimal Classification), ID, and Entry.

Once the publisher is satisfied that the appropriate metadata has been entered, and the Pageset is suitable for export, the request is submitted to an administrator by clicking the **Submit** button. Again the submission is made via a submission dialogue which uses the User finder tab of the Search Tools to select an administrator. On the other hand, for example, if the Pageset is not suitable for export for some reason, or if he finds that the author metadata which has been entered is inappropriate in some way, a message can be typed into the comments area and the request rejected by clicking the **Reject** button. In either case, an email is sent to the author informing them of the outcome of the request, plus comments entered.

Administrator's interface

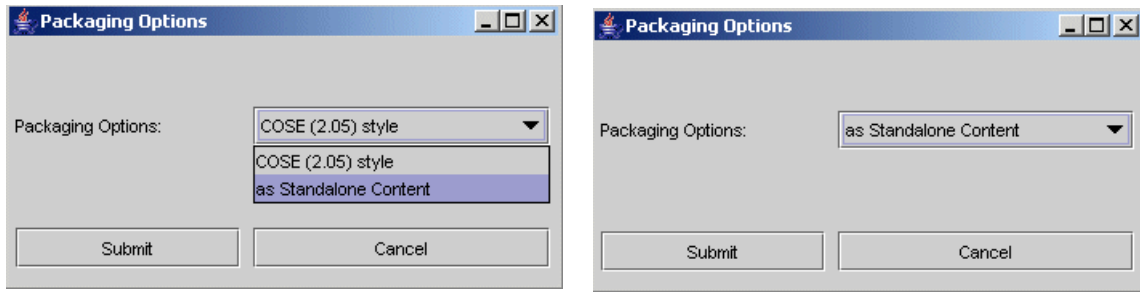
The diagram below shows the Export tab of the Administrator's interface. As in the previous section, there is a pane listing requests for Pagesets to be exported. The metadata associated with the pageset can be viewed and checked as before using the

View metadata button, and requests can be rejected with the **Reject** button, as described previously, but the other buttons have a function specific to the administrator.



Note that a Catalogue ID has been assigned to the Pageset. The first part of this digit string corresponds to the System Identifier assigned to your COSE system; you should not export Pagesets without first having been assigned a System Identifier. The second part of this digit string corresponds to the System Entry number, a counter maintained by your system. This Catalogue ID forms the base for the name given to the packaged content file. When the Pageset is packaged for export with its associated metadata using the native (COSE style) packaging option, it is packaged as a zip file, and given the name *catalogue_id.zip*, which is unique and simple to reference. If the standalone packaging option is used, the pack name is suffixed with an 's': thus, *catalogue_ids.zip*

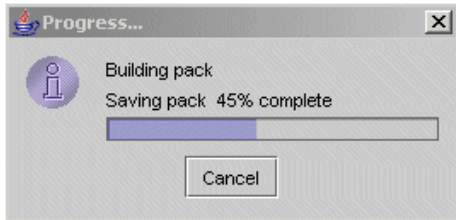
Content packages can be exchanged between different brands of VLEs systems is facilitated using standards for packaging and metadata, commonly expressed in XML. When the administrator is ready to begin exporting, having selected a Pageset request from the list, the first step is to generate an .xml system file relating to the Pageset, using the metadata previously entered.



On clicking the **Generate (XML)** button, the Administrator is presented with packaging options. For maximum interoperability and flexibility in using the content, the Standalone option will likely be preferred, but the original COSE-style option, available since v. 2.05 for content exchange between COSE systems is also available. Selecting the appropriate option and clicking **Submit**, the message displayed in the status bar changes to indicate that an .xml file is being generated. This can take some time, depending on the size of the pageset being packaged for export, but for modest sized pagesets with dozens rather than several hundred components, this step is completed in a couple of minutes. Assuming the xml file is generated without any problem, the status bar is updated to indicate that this is the case. Again, a `_s.xml` file is generated for standalone content, distinct from the `_.`xml file of the native routine. (The standalone routine has rather more work to do in the background than just to generate an `_s.xml` file here, and takes a little longer than the native routine).

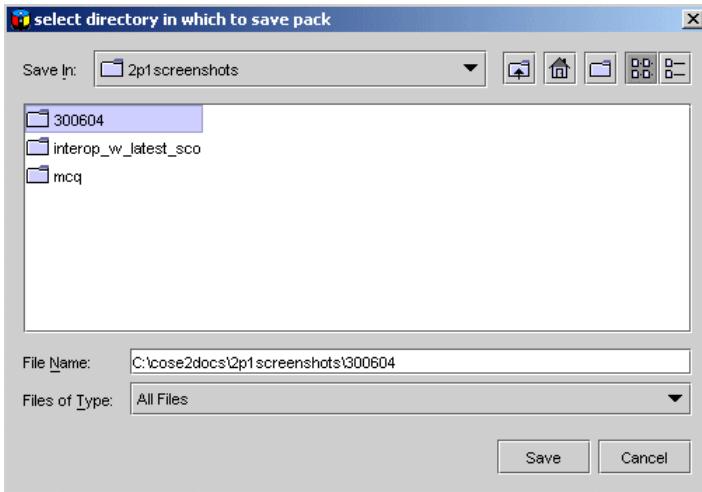
In order for the package to be exported (and subsequently imported) correctly, the .xml file generated must be valid. On clicking the **Validate (XML)** button, the message displayed on the status bar changes to indicate that the .xml (or s.xml) file is being validated. Again, for large pagesets this can take some time, but usually takes no more than a minute or two. When the file is checked as being valid, the message on the status bar is updated to indicate this.

If the .xml (or s.xml) is valid, the Pageset can be packaged. Depending on the size of the pageset, i.e. the number and nature of its components, packaging can take from a few minutes to more than an hour. When the **Build (Package)** button is clicked, the message in the status bar indicates that a pack is being built, and a progress bar appears, giving an indication of how far progressed the process is. The pack is built, copying and saving the required components, then packaged, as indicated by the progress bar.



When packaging is complete, the message displayed on the status bar is updated to read, 'Build complete. Press the Save button to copy the pack from server to local drive'.

If the administrator has direct access to the COSE server, he may choose not to save a copy of the pack to the local drive. Otherwise, on clicking 'Save Package', a file dialogue appears requesting the user to select a directory in which to save the pack, shown below.



Assuming the pack is saved successfully, the message on the status bar is updated to indicate that. Having saved the pack, or having copied it directly from the server, this is then sent to the author as an email attachment.

If for any reason the packaging is unsuccessful, or the Pageset and its associated metadata are deemed unsuitable for export, the request can be rejected and a message sent to the publisher with appropriate comments. Whether the request is processed or rejected, an email is automatically sent to the author informing of the status of the request with a default message, including any comments entered in the comments box.

e.g.

Subject: COSE export request answered

This message has been automatically generated by the COSE system

Request for export of packaged resource has been answered

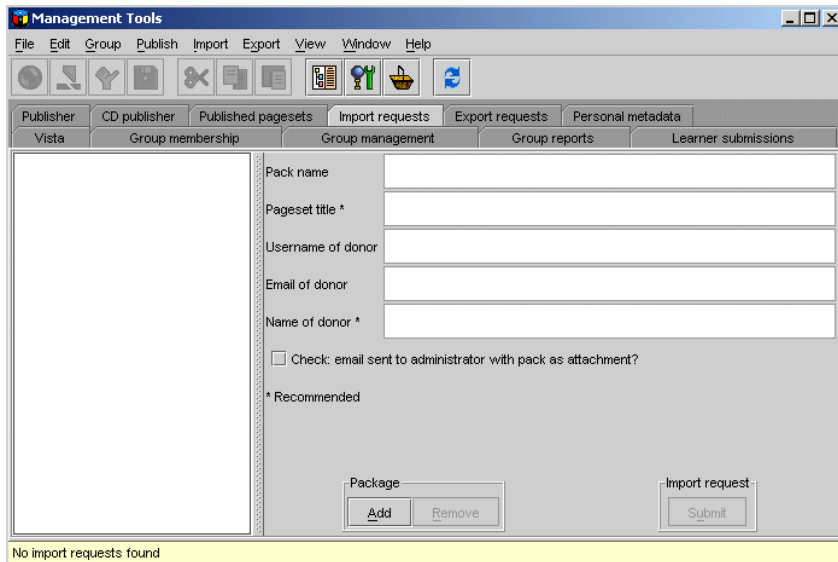
Catalogue ID: 0053_00002. Package generated as: 0053_00002s.zip.
 Pageset title: Collaborating in COSE.
 Pageset filename: cose:/misc/collaborating_in_col.act.
 Author: Mark Stiles (_ittmjs)
 Requested by: A Super (sup1).
 Answered by: admin.
 Comments: packaged as standalone content

As the pack is saved, the request is deemed as having been answered and refreshing the view will remove the export request from the list of those to be processed. Before refreshing the view, it is possible to package the content using another packaging option without having to submit another export request; this is achieved by clicking on the currently selected export submission, then running through the process again.

Importing Pagesets

The Importing interfaces complete the tools which allow the transfer of COSE content between COSE systems. The Import procedure involves two types of COSE user, supertutors and administrators, as illustrated below.

Supertutor Interface



A Supertutor submits requests for the import of COSE resources, known as packs, to an administrator using the 'Import Requests' interface illustrated below. This assumes that the Supertutor has identified a resource which they deem suitable for import, and that this content, packaged COSE style, and taking the form of a zip file archive, has been sent to him by some appropriate authority (the author, or some agency acting on their behalf).

The resource or pack name is derived as a unique identifier based on a code for the source system i.e. where the content pack was generated. The pack name includes the “.zip”

extension. As well as possessing the .zip file containing the pack, the Supertutor ideally needs to know the original title of the packaged Pageset. The username of the donor (author / agent) on the source system is also required, so that rights issues can be checked, to allow tracking of resource usage and to further aid identification. The email of the donor is also required so that they can be contacted if needed. The actual name of the donor can also be provided to facilitate communications, but is not required. These details should be requested of the donor when acquiring the resource.

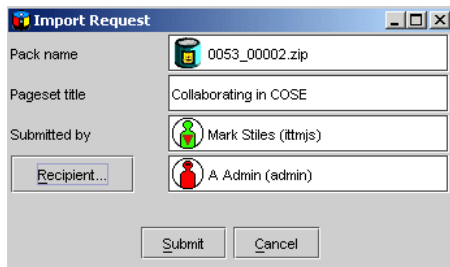
Clicking the **Add (Package)** button opens the frame below. The Supertutor is expected to fill in the fields shown. Clicking the **Add to List** button adds the request to the user's system files, closes the frame, and reloads the user's request list.

On reloading, the new request, represented by the pack name and its icon, is added to the list on the left. Selecting items from the list pane displays the request details.

Requests can be removed from the list by selecting the item in the list pane, and clicking on the **Remove (Package)** button.

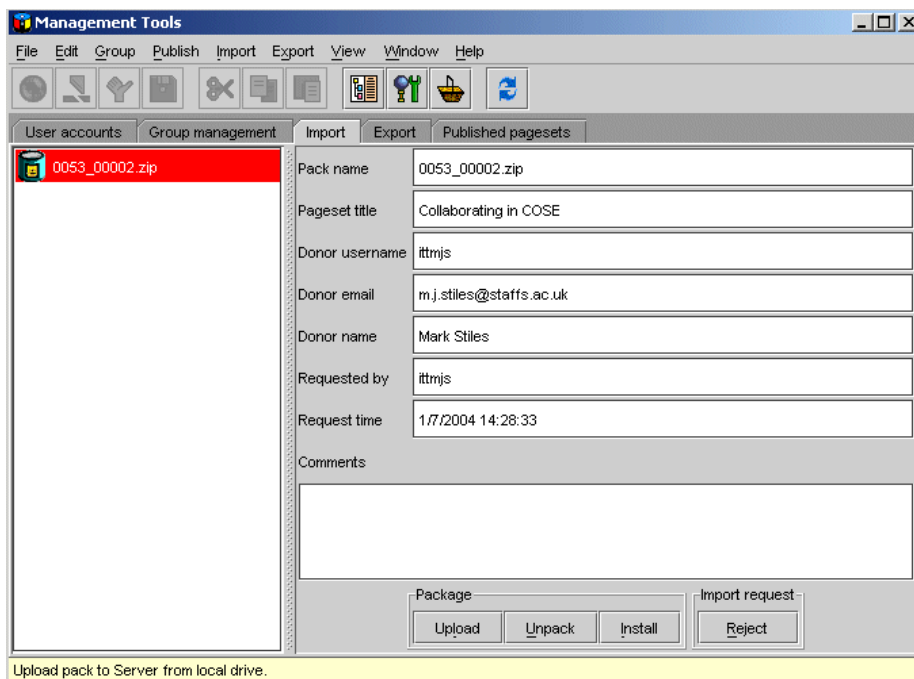
In making a request of an administrator for import of a pack, the supertutor has to supply the administrator with the zip file, usually by email. The check box is meant to remind the supertutor that this is a basic requirement of the procedure, and the tick box must be checked in order to submit a request. If the browser in which COSE runs is configured for sending email, this makes a convenient way to send the pack / zip file to the recipient as an attachment.

Clicking the **Submit (Import Request)** button opens the Import Request dialogue. The values in the frame's fields are filled according to the item selected from the request list. The user is required to select a recipient to whom the request will be submitted. Clicking the **Recipient** button, opens the Search Tool whereby the user selects a recipient from the list of Administrators, the Recipient field is filled, and the submission can be made.



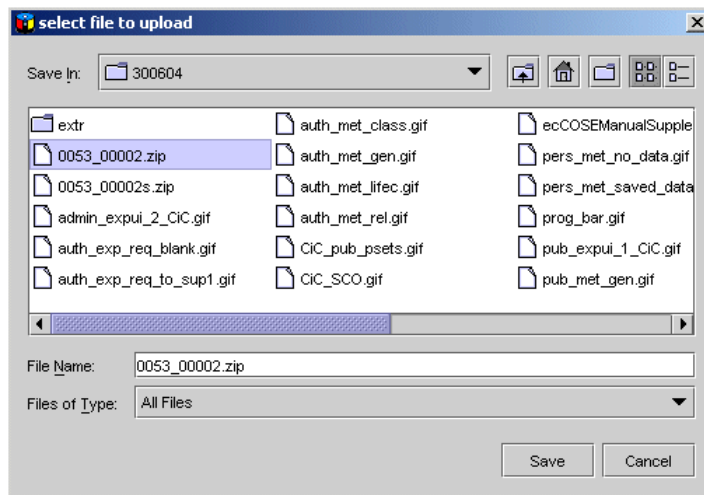
As the submission is made, the request list is updated: the request being submitted is removed from the list and the fields on the right hand side of the Importing interface are cleared. At the same time the request is written to the system files of the administrator.

Administrator Interface



The next figure above shows the Administrator interface with the pack request shown in the list. As items are selected from the list the fields on the right are filled. As the Administrator can receive requests from any Supertutor, details relating to the person making the request and when they made it are shown alongside details of the pack request. Note again that the Administrator should have received the zip file at this stage by email, or has been given access to it in some other way. The status bar displays a message instructing the administrator to upload the file to the server from the local drive.

Having selected the request, clicking the **Upload (Package)** button displays the upload dialogue.

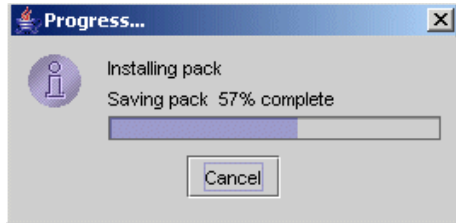


The pack should be located and **Save** clicked.

Assuming the upload is successful, this is indicated by a message in the status bar. Clicking **Unpack (Package)** will unpack the resource from its .zip file, copying the files contained therein to the appropriate system directory. Should the administrator attempt to unpack a resource without first uploading the .zip file to the appropriate location, (assuming they have received it from the person making request), an error message will be displayed in the status bar to indicate that 'Unpack' failed. Other error messages might be generated to indicate that the pack, which might have been subject of a requested made by some other user to some other administrator, had already been unpacked or installed. When the file is unpacked successfully, a message in the status bar is shown to indicate this.

After unpacking, the pack is ready to be installed by clicking the **Install (Package)** button. Installation involves parsing a manifest file unpacked from the zip file. The manifest file is an XML document which is checked as valid before installation; this can take some time, c. 20 minutes, for packaged pagesets which are particularly large. A large pageset may consist of a thousand files or more compressed in the zip archive file. Messages are displayed in the status bar to indicate the progress of the installation.

Assuming the installation is proceeding normally, and the pack manifest is valid, the message “Pack manifest is valid – installing pack, please wait ..” appears. A progress bar will then appear to indicate further progress in copying files in the system, something analogous to the publication procedure.

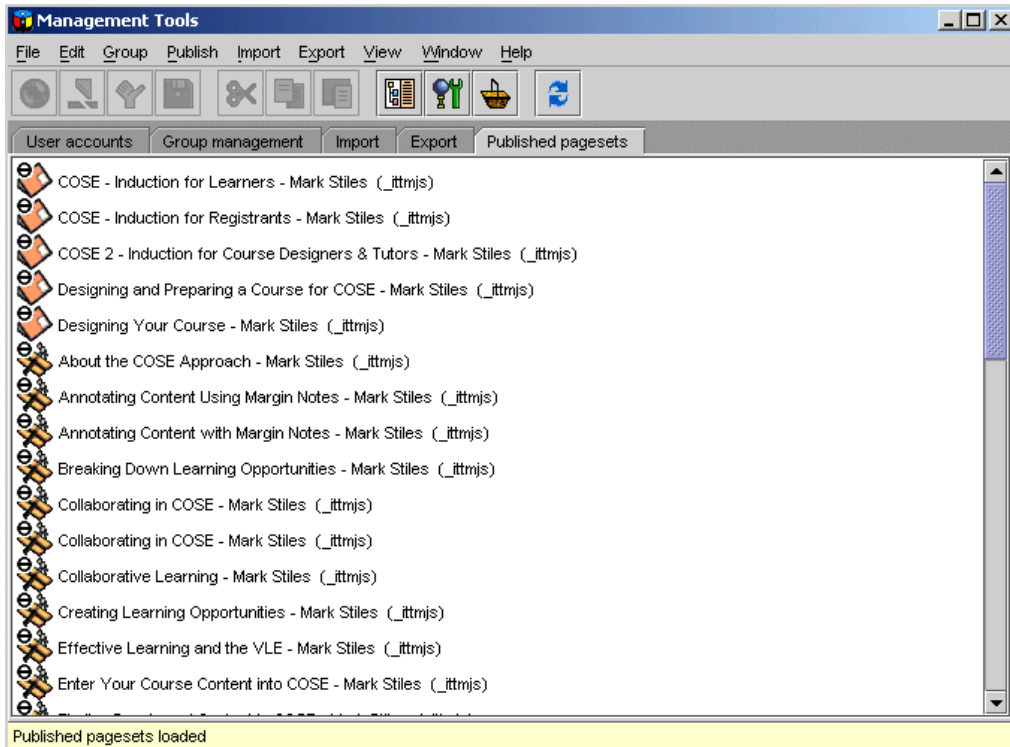


Finally a message appears, “Updating datafiles, please wait ..”. As with validation, installation may take quite some time depending on the size of the resource being imported.

On completion of the installation, the status bar is updated with the message, ‘Request completed. Refresh view for update of request list’. Selecting View | Refresh, the request is removed from the administrator’s list. An email acknowledgement sent to the requestor to say that the request was answered, including any comments the administrator might have added in the text area.

If the request is rejected for some reason, the request is removed from the list (refresh to view this) and an email sent to the requestor to say that the request was rejected, plus any comments from the administrator.

Once installed, the pack appears in the Published List and can be selected for browsing. In the figure below, there are two copies of the pageset entitled “Collaborating in COSE”, one of which has been imported according to the procedure we have described. These pagesets are identical to the viewer. Normally, of course, you would not import a duplicate pageset, but the example given here is simply to describe the process [and it is easy to verify that the procedure works by direct comparison of the two pagesets].



The final figure shows the imported pageset displayed in the COSE browser.

