Brihaspati Editor User Manual

Jan 2008

Table of Contents

1 Introduction 3
IMS Specifications 3
Where to get EDITOR: 4
This Guide 4

2 Editor 5
System Requirements (MS Windows) 5
Installation (MS Windows) 5
Uninstalling (MS Windows) 5
EDITOR Workspace 6
Main Toolbar 8
Resource Pool Toolbar 8

3 Tutorial 9
Opening an Existing Content Package 9
Creating a Simple Content Package 9
Collect Resources Together 9
Create Folder Structure 9
Open EDITOR and Set Up Workspace 10
Add a Reference to Metadata 11
Import Content 12
Create an Organisation 13
Add Items (Content) 13
Drag and Drop 14
View Content Package 16
Tidy Content Package 17
Multiple Organizations 18
Save Content Package Preview 19
Add Metadata 20
Multiple Entries 20
Save Content Package 21
Save As Content Package 21

4 Metadata 22
Creating a Standalone Metadata File 22

5 Customising Editor 23
Metadata Profiles 23
Metadata for Individual Resources 25
Content Packaging Profiles 25
Internationalization 25

6 SCORM Elements 25
Types of SCORM Package 25
SCORM Metadata 26
SCORM Specific Elements 27

7 Further Actions 29
References and Dependency 29
Finding References 29
Adding Files and Dependencies 29
Manifests and Sub-Manifests 29
Aggregation 29
Disaggregation 31

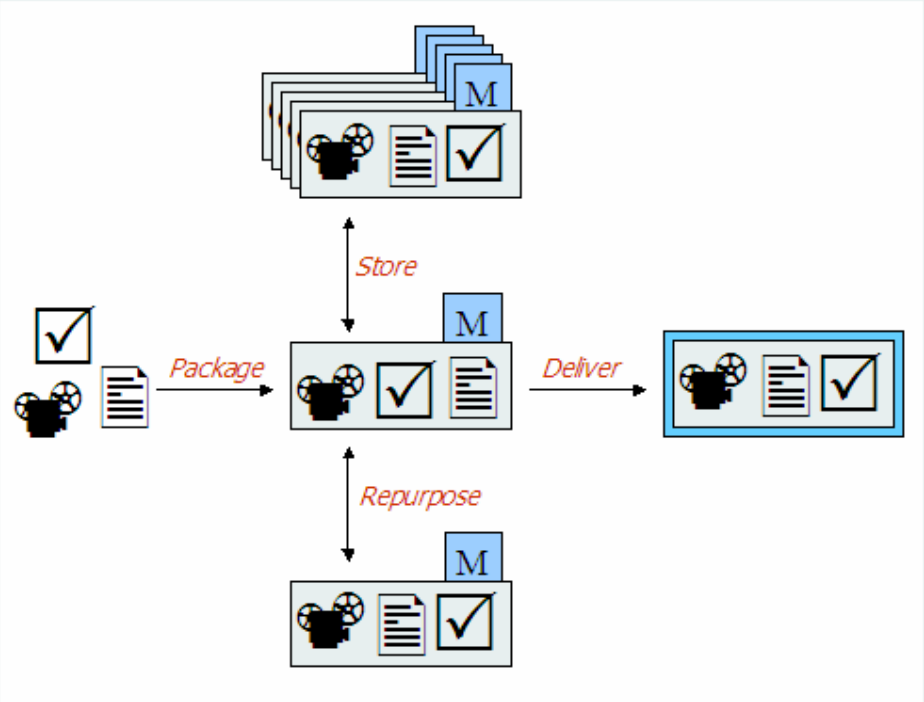
1 Introduction

The Brihaspati Editor is a Content Package and Metadata editor. With Editor, you can take your own electronic content (web pages, images, flash animations, Java applets etc.) and package and describe it ready for storage in content repositories.

The Brihaspati editor is of significant value to the UK Higher Education community and beyond, since it provides the crucial "missing link" which allows users to author and transfer learning objects, in specification compliant format, between authoring and design tools, local and distributed digital repositories, and Virtual Learning Environments (VLEs).

Editor provides the following functions:

- Packaging content created by other tools.
- Repurposing existing content through resentment and reorganization.
- Preparing content for storage in repositories such as JORUM.
- Delivery of content to end users using the 'Save Content Package Preview' facility.



IMS Specifications

The Brihaspati Editor implements the IMS Content Packaging and Metadata specifications. If you are not familiar with these specifications then it will be helpful to refer to the documentation available at the IMS web site:

CONTENT PACKAGING: http://www.imsglobal.org/content/packaging/index.cfm
METADATA: http://www.imsglobal.org/metadata/index.cfm

At present, the Brihaspati Editor supports v1.1.3 of the IMS Content Packaging specification and v1.2.2 of the IMS Metadata specification.

v1.3 of the Brihaspati Editor also offers the option to edit SCORM 1.2 packages. Information on SCORM is available from ADLNet.

SCORM: http://www.adlnet.org/ and click on the SCORM tab.

Editor has been stable and fully functional. Subsequent versions have sought to provide improvements in usability and customizability.

This Documents

This document has seven sections:

- this Introduction section provides some background to the Brihaspati Editor tool:
- the Brihaspati Editor section describes the installation process and the user workspace,
- the Tutorial section describes the whole process of creating a simple Content Package,
- the Metadata section provides further information on the Metadata features of Editor,
- the Supporting Other Application Profiles and Vocabularies section describes how to customise Editor to your individual needs,
- the SCORM Elements section describes the SCORM functionality,
- the Further Actions section describes some more advanced features of Editor.

2. Brihaspati Editor

The Brihaspati Editor is a Java applet application, and should run on any platform capable of running Java applications. This JRE (v1.4.2.04) should not interfere with other Java Virtual Machines installed elsewhere on your system.

System Requirements (MS Windows)

To run Brihaspati Editor you should have a PC with at least the following specification (for Linux and Mac (OSX), a similar/equivalent specification would be required):

- Intel Pentium 3 (or equivalent) Processor, 800Mhz,
- 256 Mb RAM,
- Microsoft Windows 95, 98, Me Windows NT4.0, Windows 2000 or Windows XP,
- A Web Browser for viewing Content Packages.

Editor space basically consists of three panes: **a resources pane (left), a manifest pane (right)**, and **an attribute pane (bottom)**. The manifest pane is the key area as this space represents the structure of the Content Package – with a Manifest containing Metadata, Organizations and Resources. The attribute pane includes a section with context sensitive information on the currently selected element as well as a table of values and boxes for editing attributes.

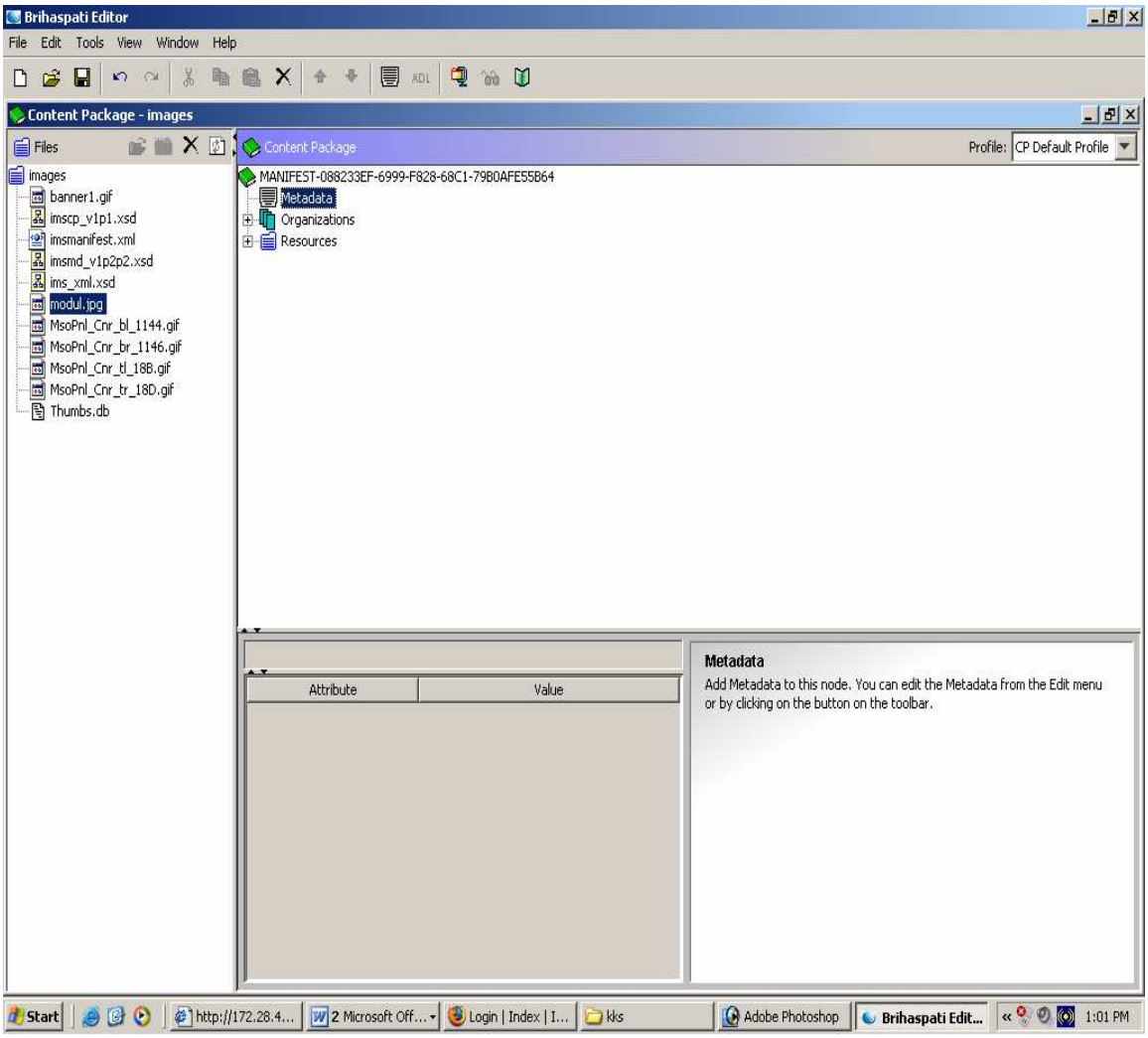

Brihaspati Editor
File Edit Tools View Window Help
Content Package - images
Files
images
banner1.gif
imscp_v1p1.xsd
imsmanifest.xml
imsmd_v1p2p2.xsd
ims_xml.xsd
modul.jpg
MsoPnl_Cnr_bl_1144.gif
MsoPnl_Cnr_br_1146.gif
MsoPnl_Cnr_tl_18B.gif
MsoPnl_Cnr_tr_18D.gif
Thumbs.db
Content Package
Profile: CP Default Profile
MANIFEST-088233EF-6999-F828-68C1-79B0AFE55B64
Metadata
Organizations
Resources
Attribute
Value
Metadata
Add Metadata to this node. You can edit the Metadata from the Edit menu or by clicking on the button on the toolbar.
Start
http://172.28.4...
2 Microsoft Off...
Login | Index | I...
kks
Adobe Photoshop
Brihaspati Edit...
1:01 PM

Figure 1

Each window represents a Content package and multiple windows may be open at any one time. If you wish to view the Metadata associated with a particular Content Package, then a new metadata window is launched in addition to the Content Package window.

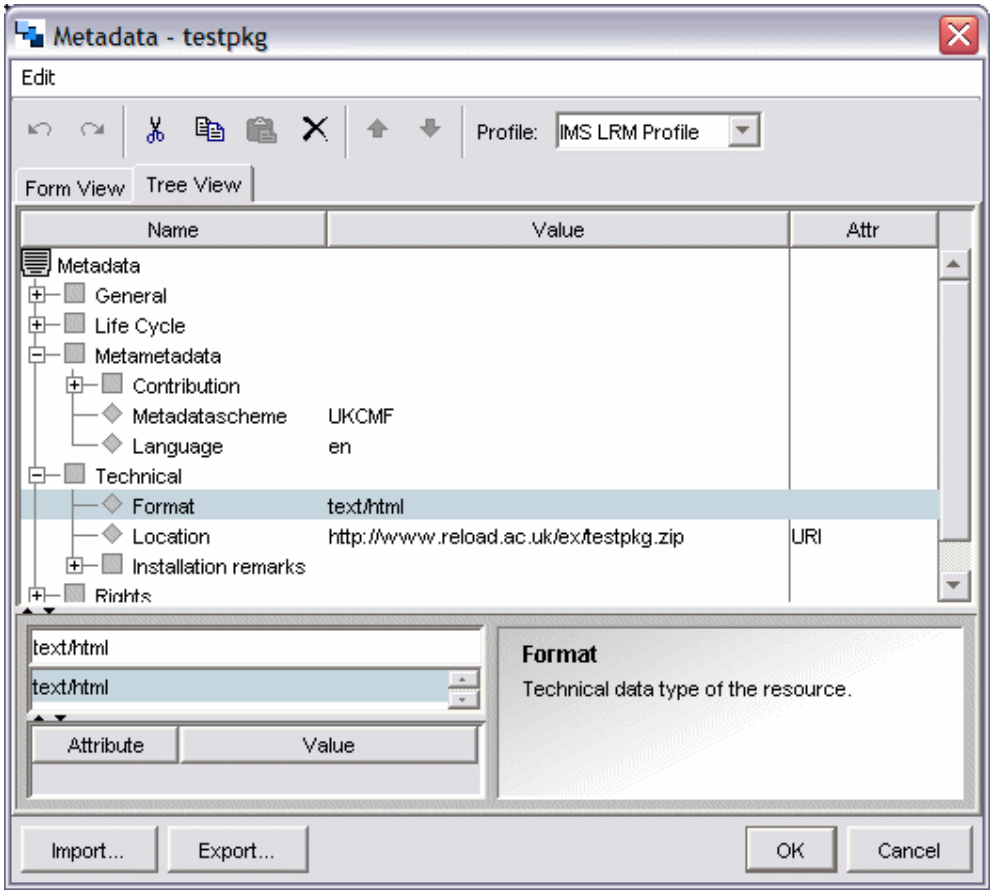


Figure 2

Toolbars

The Brihaspati Editor has two toolbars. The Main Toolbar, with general options, and the Resource Pool Toolbar with access to actions specific to the Resource Pool.

Main Toolbar

From left to right, the icons represent (with equivalent menu actions and/or keyboard shortcuts in brackets):

New (File, New) creates a new Content Package (SCORM or IMS) or Metadata Record.

Open (File, Open) opens an existing Content Package (CP) or Metadata Record.

Save (File, Save Ctrl+S) to save the current CP or Metadata Record.

Undo (Edit, Undo Ctrl+Z) to undo the last action.

Redo (Edit, Redo Ctrl+Shift+Z) to redo the last action.

Cut (Edit, Cut Ctrl+X) to cut the current selection.

Copy (Edit, Copy Ctrl+C) to copy the current selection.

Paste (Edit, Paste Ctrl+V) to paste the contents of the clipboard.

Delete (Edit, Delete del) to delete the current selection.

Move Up (Alt+Up) to move the selected file within the CP.

Move Down (Alt+Down)to move the selected file within the CP.

Edit Metadata (Edit, Edit Metadata) to edit metadata for the selected manifest.


Edit SCORM (Edit, Edit SCORM) properties for a given Item.


Make Content Package (File, Zip Content Package) to create a .zip CP.


View File (View, View File)to view an individual file (via browser).


Preview Content Package (View, Preview Content Package) previews the CP.

Resource Pool Toolbar

From left to right, the icons represent (with equivalent menu actions in brackets):


Import to Resource Pool (File, Import Resources ...)


Create New Folder


Delete from Resource Pool (del)


Refresh, refreshes the file list for the Resource Pool.

You will also come across other icons in system menus. These will be explained as we encounter them.

---

3 Tutorial(How to Create Content Package):

Opening an Existing Content Package

If you wish to open an existing Content Package choose File, Open. This displays a dialogue box prompting you to choose either a Zip file or .xml manifest. If you choose a Zip file, you will be asked to choose a folder to unzip to, and this will subsequently become the working folder for the Content Package. Choose the folder name wisely, it will be utilised later. Dragging a Zip file (or imsmanifest.xml) onto the EDITOR workspace can be used to quickly open packages.

Creating a Simple Content Package

To understand fully the facilities offered by Editor, you must create and work with Content Packages. The following pages lead you through all the stages in creating a simple CP.

Collect Resources Together

A Content Package is a means of transferring a set of resources from one location to another whilst retaining their structure and inter-relationships. When creating the Content Package we create a space in which all local files are kept - a base folder. As we build the Content Package, all the files we use will be stored under this working folder, but their original location can be anywhere on the local machine (indeed external resources can be referenced by hyperlink, but we will only deal with local files here). For simplicity, we will start out with all the files that we envisage using within a single folder structure. If you wish to work through this tutorial yourself, you will probably set up a similar structure.

Create Folder Structure

Define a single folder for Content Packages on your computer (for ws e.g. c:\cp) or (linux e.g.user.home/mydir) Within this folder, make a new folder called test-src (c:\cp\test-src) or (linux e.g user.home/mydir/test-src).Within that folder, you will place all the files you need for your Content Package.

Download the source zip file http://www.Editor.ac.uk/ex/test-src.zi (131kb) and extract to c:\cp\test-src. The folder should now hold eight .html files, as well as a sub-folder (supp)

containing nine images (.jpg and .gif) and a stylesheet (.css). Some unzip utilities may have trouble with recreating the folder structure, if this occurs, you will need to recreate it yourself.

Create another folder testpkg at the same level as your test-src folder (c:\cp\testpkg). This folder will be used as the working folder for our Content Package. Your folder structure should resemble that shown below.

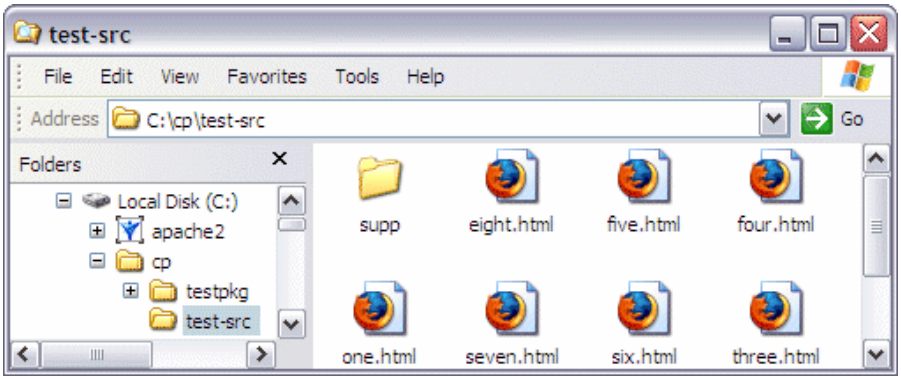


Figure 3

Open EDITOR and Set Up Workspace

Launch editor (By clicking the Packager link). The Editor opens up with no files open. We want to create a new Content Package, so:

Click File, New, > IMS Content Package. A dialog box appears and you are asked to select a folder for the Content Package. Navigate to c:\cp\testpkg, select this and click Select.

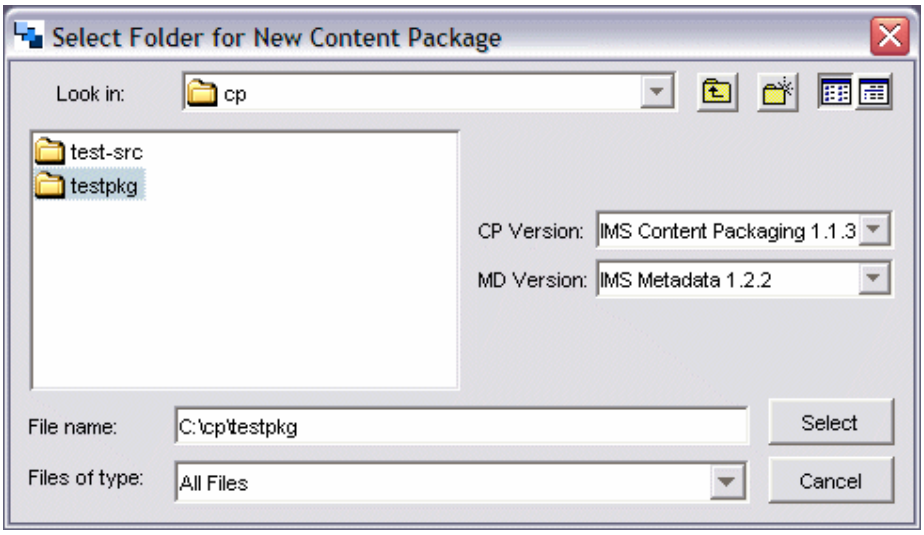


Figure 4

The 'Select Folder' window also allows you to change the default IMS MD and CP schemas to use. Unless you have a specific reason (creating content for a legacy learning environment) then you should choose the most up to date schemas: IMS CP 1.1.3 and IMS MD 1.2.2

A new window appears with the title Content Package - testpkg (taken from the folder name) and three frames showing:

- the tree structure of files and folders (the resources view),
- a representation of the Content Package (the manifest view),
- information about individual elements (the attributes view) .

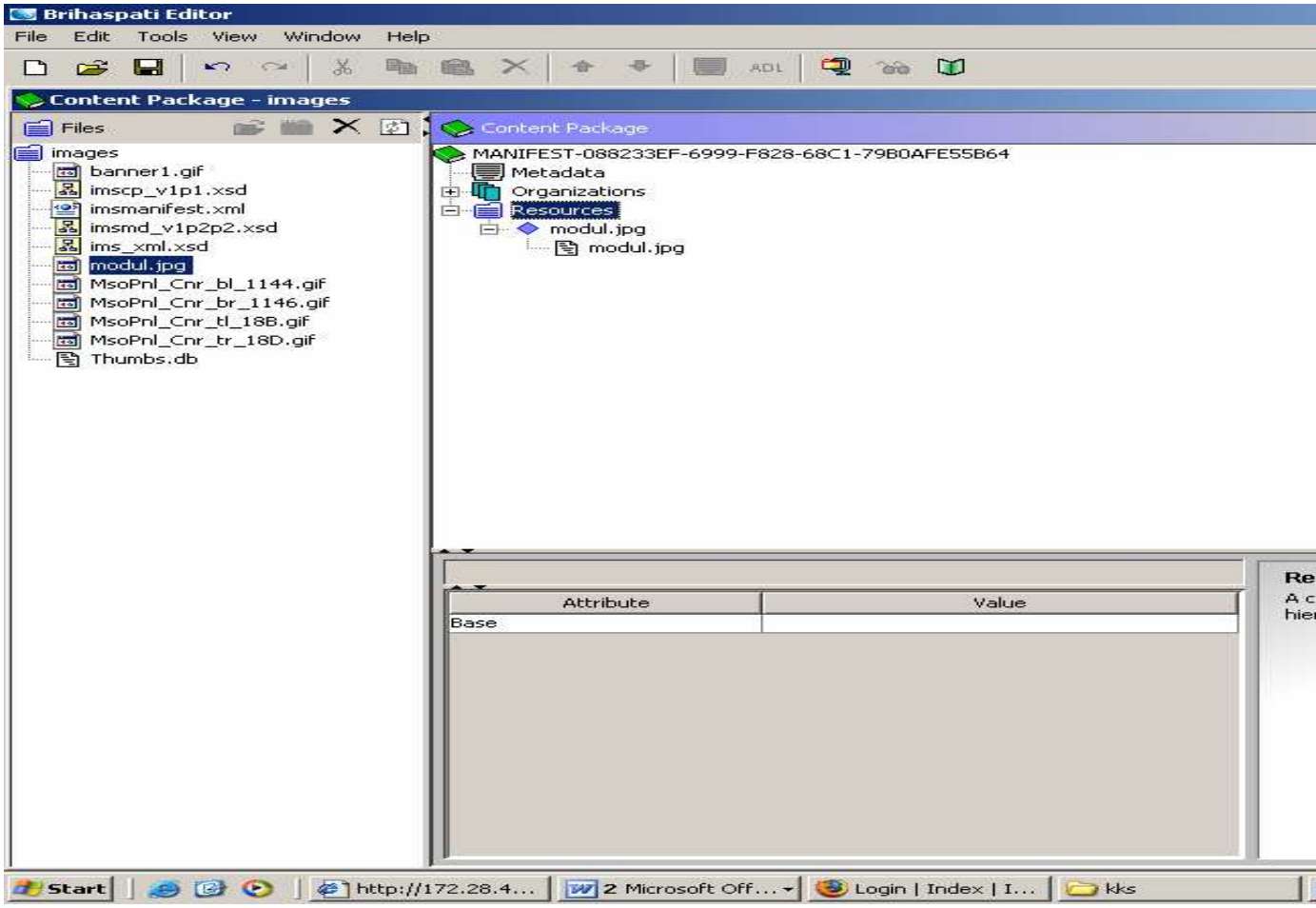


Figure 5

On creating the Content Package, the Brihaspati Editor automatically creates the main data: file:

- **imsmanifest.xml** - the manifest for this Content Package, where all the information will be stored. (the name imsmanifest.xml is mandatory, and this file must appear in the root of any valid Content Package).

Three other files are created, each referenced by the manifest (inclusion of local copies of schema files for offline validation is recommended in the IMS CP Best Practice Guide).

- imscp_v1p1.xsd - a local copy of the content packaging XML Schema Document.
- imsmd_v1p2p2.xsd - a local copy of the metadata XML Schema Document.
- ims_xml.xsd - a local copy of the XML Schema Document.

Add a Reference to Metadata

At the moment the Content Package contains no content, before we add any content, we should add some metadata, or rather, a placeholder - we will add the metadata later.

Right-click on the Manifest node in the manifest pane and choose Add Metadata.

Right-click on the Metadata node that is inserted and click Add Schema.

Select the schema and type IMS Content into the value field.

Right-click on the Metadata node again and this time click Add Schema Version.

Select the schema version and type 1.2.2 into the value field at the foot of the window. Although we have yet to add any metadata, we have defined the format of any metadata we will add as conforming to the IMS Metadata specification v.1.2.2

The Metadata node in the manifest pane should now look similar to the one shown below.

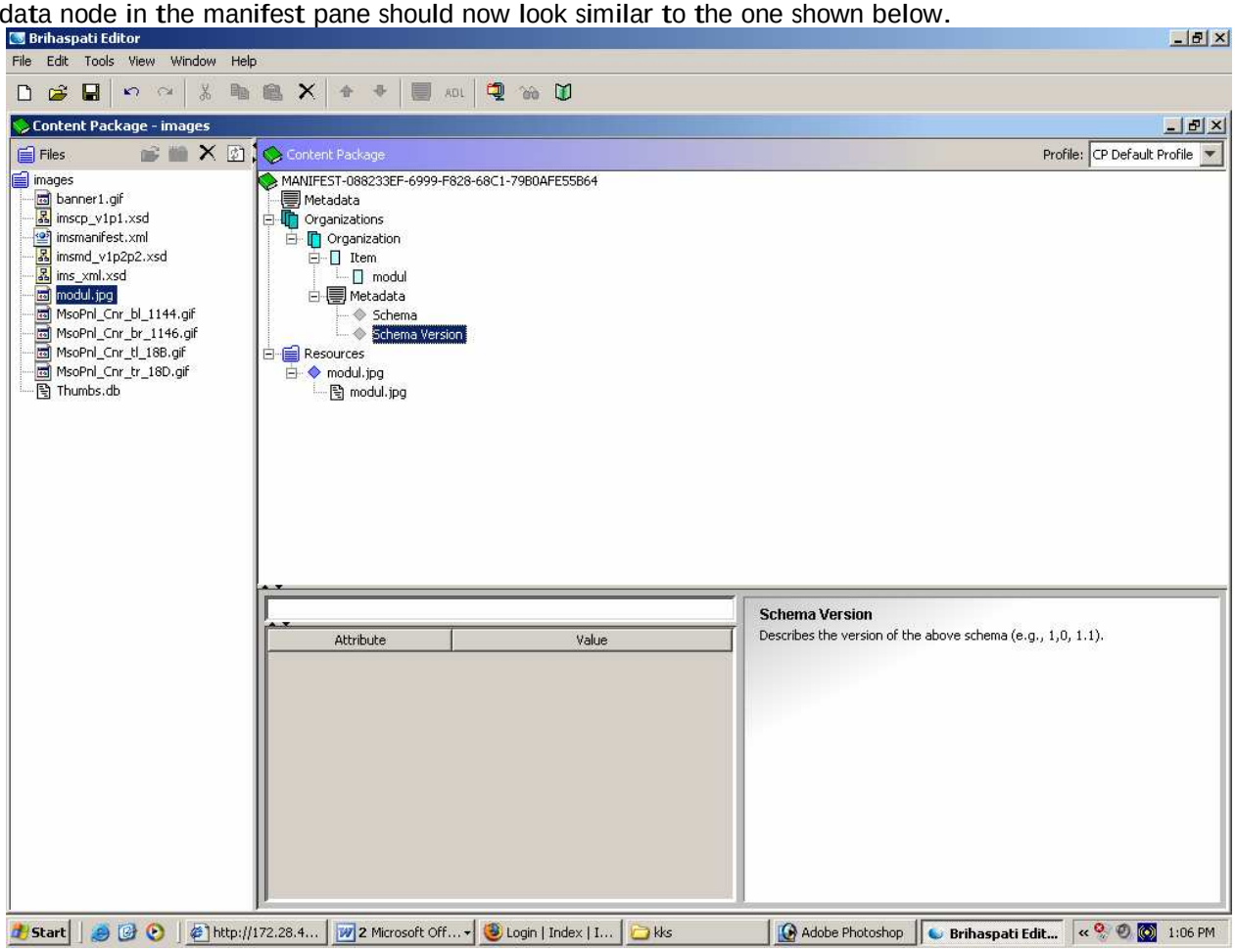


Figure 6

At any time you can view the imsmanifest.xml file by selecting it and clicking on the view file icon in the Main toolbar. This will show you the current structure of the manifest file. Click the Save Icon first to update its contents.

Import Content

To create our Content Package, we must first import some content.

To ensure that resources are imported to the correct place, click on the root node (testpkg) in the Resources Pane. From the File Menu, select Import, Import to Resources Pool ... This brings up the Import Resources File Dialogue. Alternatively, click the Icon in the Resource pool toolbar.

Navigate to the source directory you made earlier (c:\cp\test-src\).

Select all the files, including the subdirectory.

Check the 'Include Dependent Files' Option. The **Brihaspati Editor** will parse html files etc. to find dependent images, stylesheets

Click Open. Click Yes if asked to overwrite files (e.g. stylesheets and logo's which may have been referenced in more than one file).

The files you imported now appear in the tree view on the left hand side of the Editor workspace. You should see the eight .html files in the root space, as well as the supp folder containing the images and stylesheet.

Your workspace should resemble that shown below:

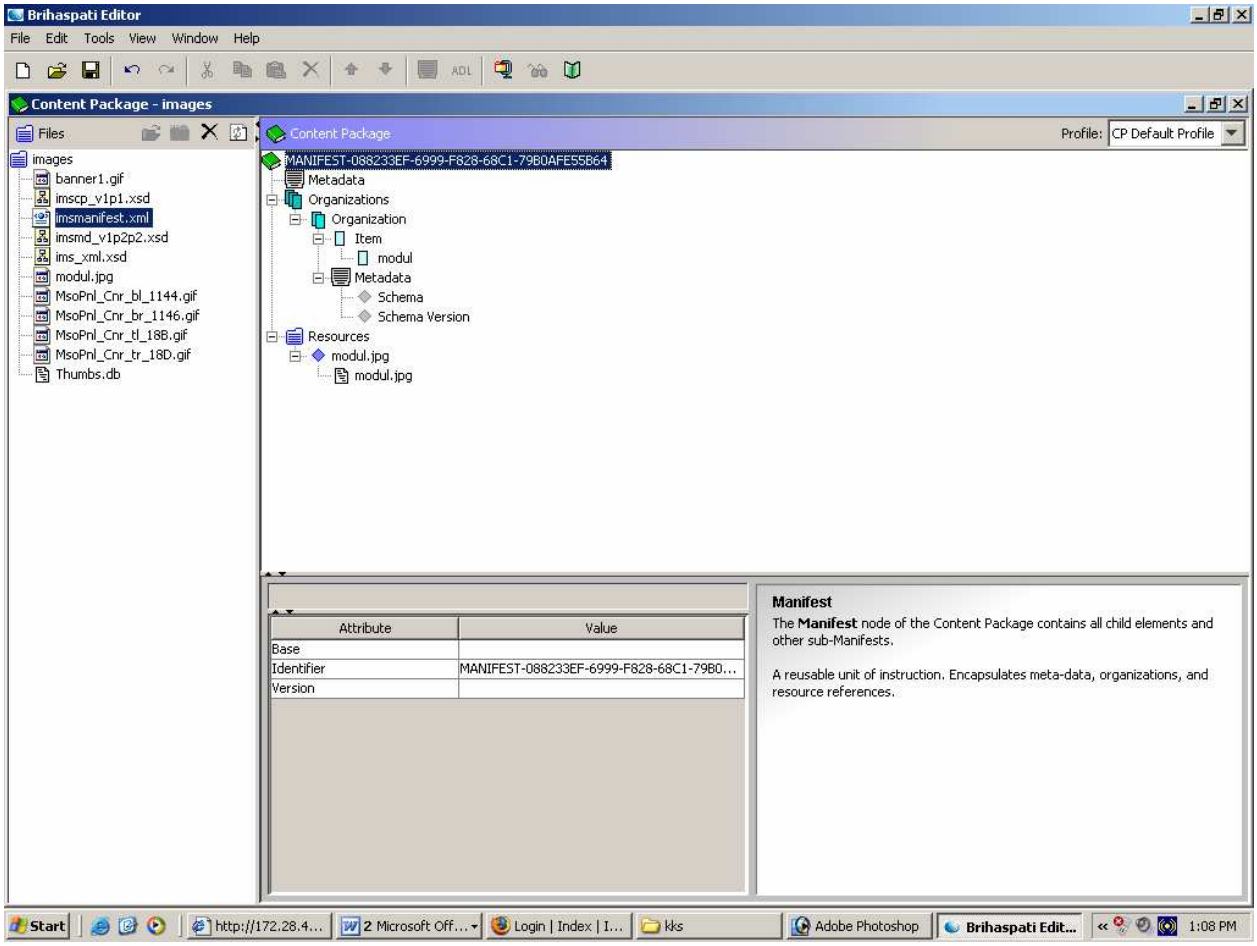


Figure 7

Create an Organisation

A Content Package consists of one or more organisations of content. By default the CP we create has no organisations, so we have to add one.

Select the Organizations node  in the manifest view and right-click.

From the context sensitive menu, click 'Add Organization'

Select the new Organization node  and, in the attribute pane at the bottom of the workspace, enter a value in the box. You can choose any title, in the oackage below, we have used the title 'Main' for our single Organization.

Your workspace should resemble that shown below.

Add Items (Content)

Finally, we can add some content to our Content Package. Basically, we create the structure for our Content Package by adding Items to an Organisation. Adding Items to the organisation can be carried out using menus (Edit, Add Item, or right-click, Add Item) but it is far easier to just 'drag and drop' content from the tree pane to the manifest pane.

Drag and Drop

Moving content around between the three panes and within panes is easy if you use drag and drop. You can easily:

- drag and drop Files or Folders from the resources pane to become Resources in the manifest pane
- drag and drop Files or Folders from the resources pane to become Items in the manifest pane
- drag and drop Folders from the resources pane to become Organizations in the manifest pane
- drag and drop Resources in the manifest pane to become Items in the manifest pane
- drag and drop from the Desktop to a folder in the resources pane

NB When selecting multiple files you must hold down shift/ctrl as you drag.

Select the resource you want to add from the resources node in the manifest pane. You can start with one.html (though you don't have to).

Drag the file to the ''Main' Organisations node of the manifest file. A new Item appears, with the title One. When the editor adds content to an organization it will attempt to provide a sensible name by parsing the html to read the contents of the file's <title> tag.

Notice that as you add content to the 'Main' Organizations node as Items, the file (and its' dependents) are added to the Resources node.

Continue adding content to the 'Main' Organizations node. Don't worry about adding the files in order, if you want to rearrange them, you can always select individual Items and use the 'Move Up' and 'Move Dn' buttons to reorganise. If your content exists entirely within a folder (not the case here) then you can drag the whole folder across to the Organisations node from the Tree Pane and the whole adding organizations and item process is carried out automatically (you may have to tidy up a little).

You can also delete content from the organization - select the content and click the delete button. This won't delete files, only the reference to them within the Content Package.

Once you have finished adding content, your workspace should resemble that shown below.

View Content Package

Once all content has been added to the main organization, you are ready to preview the Content Package in a web browser.

Click the 'Preview Content Package' icon on the main toolbar, or click the same item on the View menu. A web browser should open on your computer.

In the browser, a three-framed page will load, the banner frame provides some navigation, the left frame will correspond to the structure of Content Package you have defined, the right pane will normally contain the first item of the default Organization.

In the banner frame, the name of the currently loaded Item is displayed. To the right hand side of this banner, the arrow icon allows the user to show or hide the package structure. 'prev' and 'next' buttons allow the user to navigate through the content sequentially. In the left pane, the active Organization is shown, with its' title in black and the content in that Organization in blue. Clicking on any of the blue links will load the corresponding content into the main frame. An extra drop down menu will appear if there are multiple Organization elements. See the later section for an illustration.

Your previewed Content Package should resemble that shown below.

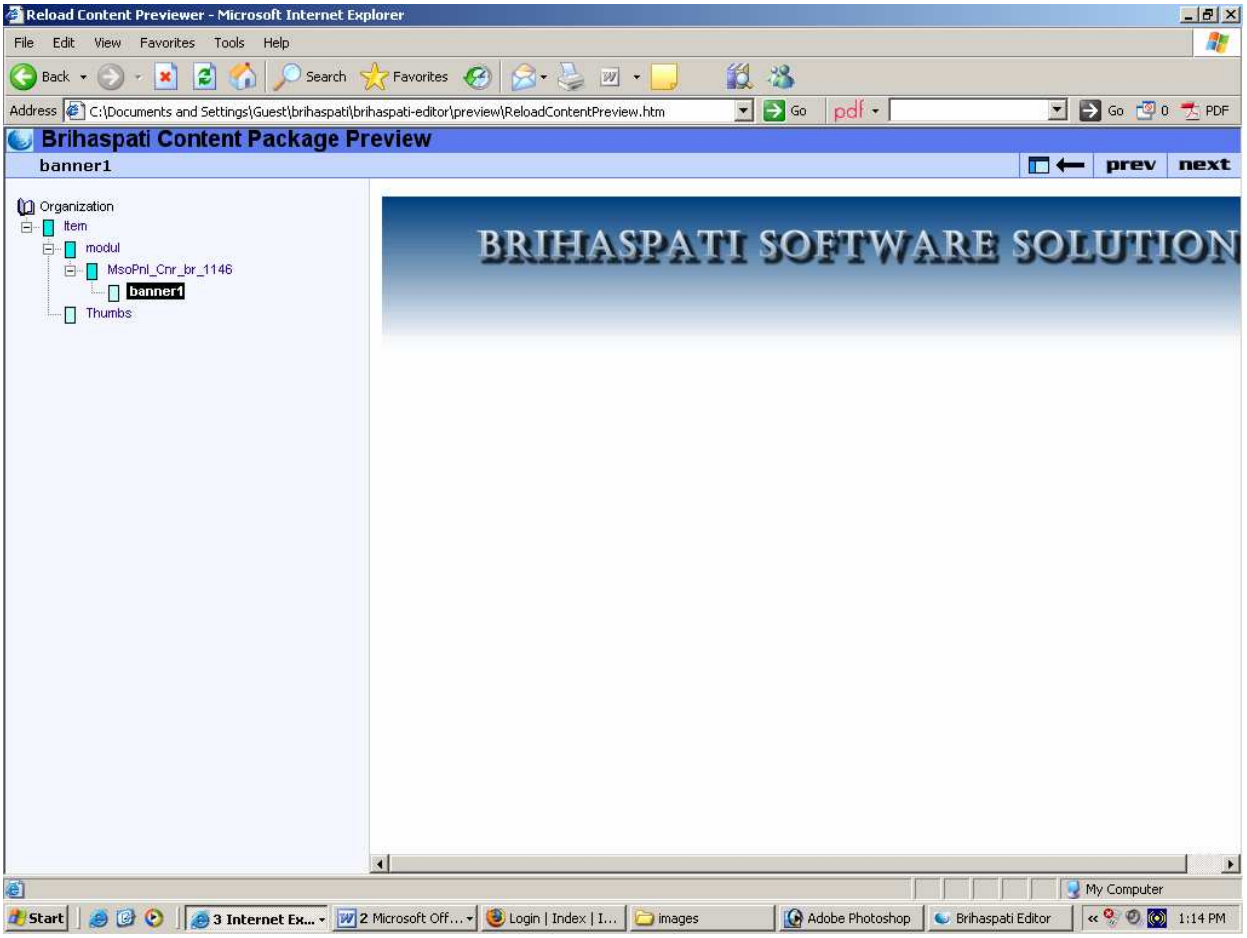


Figure 10

Tidy Content Package

The Content Package is now fully functional, but we may wish to tidy up the package before final export. For instance, we could:

- rename the Main Organization to 'Summer Pictures' (Select the Organization and edit the entry in the attribute window at the bottom.)
- rearrange the content to follow a different structure - reflecting the subject of the individual photographs. (Use 'Move Up' and 'Move Dn' buttons to rearrange the content,
- rename the Items so that their names are more meaningful than One, Two Three etc.)

When you preview again (close and relaunch the browser to flush the old preview files from its cache), the Content Package previewer will pick up the new name for the files.

If you wish, you may now add further Organizations (you can reuse the same content) and observe how the structure of the previewed Content Package corresponds to the Organization/Item structure you have created.

drop-down box controls which Organization is displayed at any time. Other software displaying the Content Package may choose to render multiple Organizations in different ways. The screenshot below shows a second Organization – 'Parliament' as well as the initial 'Summer Pictures' Organization.

Multiple Organizations

So far, we have created relatively simple structures of content. One way to create more complex structures is to use multiple Organizations. Multiple Organizations can be utilised to specify alternative (parallel) structures to the same material. Multiple Organizations may also be used to provide alternative content for the same learning outcome.

Adding a second Organization is simple. Right click on the Organizations element and click 'Add Organization'. Type a new name for the new Organization in the box in the attribute pane. Add Content to the new Organization by dragging it over from the resources pane as before. You may have to rename Items (as before) to provide meaningful titles.

When there are multiple Organization elements present, a default should be specified. If no default is specified then the first Organization in the manifest is presumed to be the default. To specify a default Organization, you should select the Organizations element, then select the appropriate Organization from the drop-down list in the Attribute pane.

When there are multiple Organizations in a Content Package, the Editor Content Previewer will display a drop-down menu in the left navigation frame. This

Save Content Package Preview

A new feature in Editor v1.2 allows you to save not just the Content Package, but a copy of the resource you have created for preview - ready for presentation in a Web Browser. Normally, Content Packages are imported into software (Learning Management Systems or Repositories) which interpret information held within the manifest and create a navigational structure for the content. Sometimes, you may wish to utilize the content you have organised outwith an LMS or repository, perhaps as a simple web site. For this, it is useful to have the navigational structure generated by Editor for Web Preview included.

To save a copy of the Web Preview, click File, Save Content Package Preview ...

Choose an appropriate folder in the dialogue box which appears (you may wish to create a new folder at this point). When you click 'Select', all the files necessary for preview are copied into this folder. Using Windows Explorer, have a look at the content. Clicking on the file 'Editor ContentPreview.htm' will launch it in a web browser. All the files in th efolder you created are required to display the content correctly with navigation.

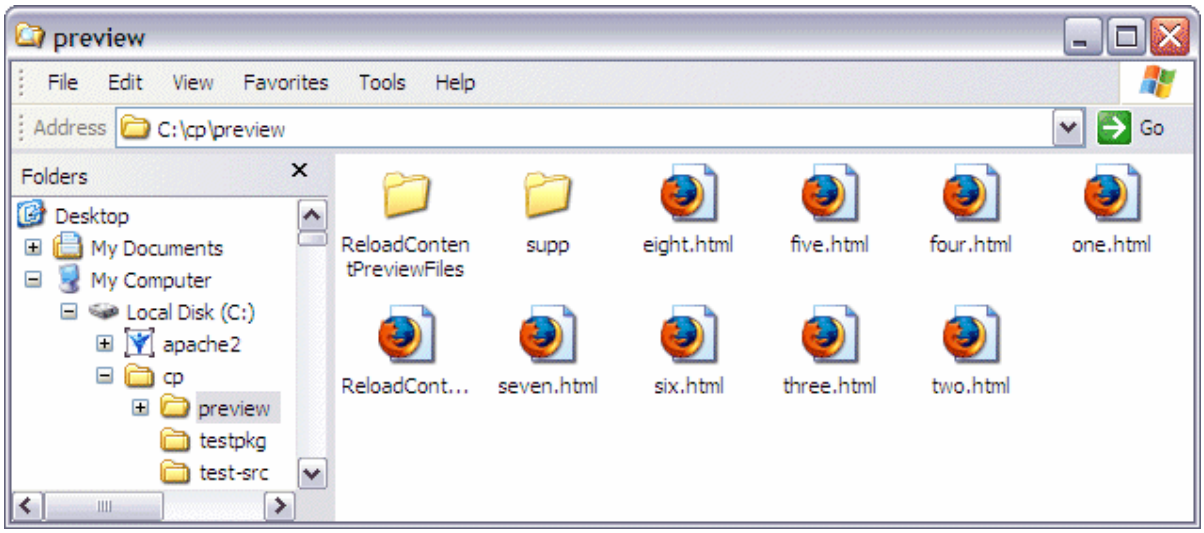


Figure Error! No text of specified style in document.**-1**
Figure 13

It is important to stress that the Content Package Preview is not in itself a 'Content Package'. This is because it doesn't include the necessary imsmanifest.xml. In fact it is no more a 'Content Package' than any collection of web pages.

A preview of the content is very useful however and it is possible to store the preview within the Content Package itself. If you choose the Content Package folder after 'Save Content Package Preview Q' then the Preview files are saved inside the Content Package folder. Now, when you zip up the Content Package for storage, the Preview files are also included (though they DO NOT appear in the manifest). By including the extra Preview files, the Content Package can be quickly previewed by a third party who does not have access to a Package viewer like MS LRN or Editor itself.

It is permissible for files to appear in the zipped Content Package which are not referenced from within the manifest, however, other software which reads and imports Content Packages may object to the extra files as it won't know what to do with them.

Add Metadata

The eventual destination of a Content Package will probably be within a Learning Management System (LMS) but it is also likely that the Content Package will be placed in a repository to facilitate re-discovery and re-use. Adding Metadata to the Content Package will create information that can be searched by users of the database.

The IMS Metadata specification (and the IEEE LOM, the standard influenced by IMS Metadata) provide a starting point for adding metadata, but ultimately, guidelines for implementing metadata at a local level are needed - to ensure that vocabularies are compatible, and that a common core of metadata is stored for all content in your locale.

Within the UK Education community, a second draft Metadata Framework (UK Learning Object Metadata Framework: http://www.cetis.ac.uk/profiles/uklomcore/) has been produced (previously referred to as the UKCMF: UK Common Metadata Framework). It would be useful to look at the UK LOM Framework document if you wish to add metadata to your content as it provides useful guidance.

If you recall, when we initially set up the Content Package, we decided that we should conform to IMS Metadata v1.2.2. but we didn't actually add any metadata. Now we can add Metadata. Select the Metadata node  in the Manifest view.

Right-click and select 'Add Metadata' from the context-sensitive menu.

A new window appears, showing various metadata fields. There are two views, Form View and Tree View. Form View is the view we will use, it lets you add metadata into specific fields. Tree View lets you see the entire metadata record and lets you create more sophisticated metadata.

The Form View is really just a simpler representation of the full metadata Editor, and it is possible to add information in the Tree View which is not shown in the Form View, even though the data is retained. For our purposes, Form View will be ideal.

From the Profile combo box, select UKCMF Profile - as you do so, you will see that the choice of fields in the form changes slightly, reflecting the UK LOM Framework (UKCMF) Profile.

Using the UK LOM Framework document as a guide (available from http://www.cetis.ac.uk/profiles/uklomcore/), go through the form adding (valid) metadata to the form. When you have finished, click OK.

Multiple Entries

If you wish to add more than one entry for a given element (for instance if you want to refer to content containing multiple mime types (e.g. a web page with a flash animation), you need to use the tree View rather than the Form View.

- First change from Form to Tree View in the metadata editor.
- Next select the appropriate element, e.g. Technical/Format, right-click choose Copy.
- Select the technical element and again right-click - this time choosing Paste.
- Use the scrolling list to choose a new mime-type.

If you go back to the Form View, only the first element will be shown, but the information is stored and will be retained on Save. This procedure is the same for any element which allows multiple entries.

Once you finish editing, your completed Metadata form (Tree View) may look like the one shown below.

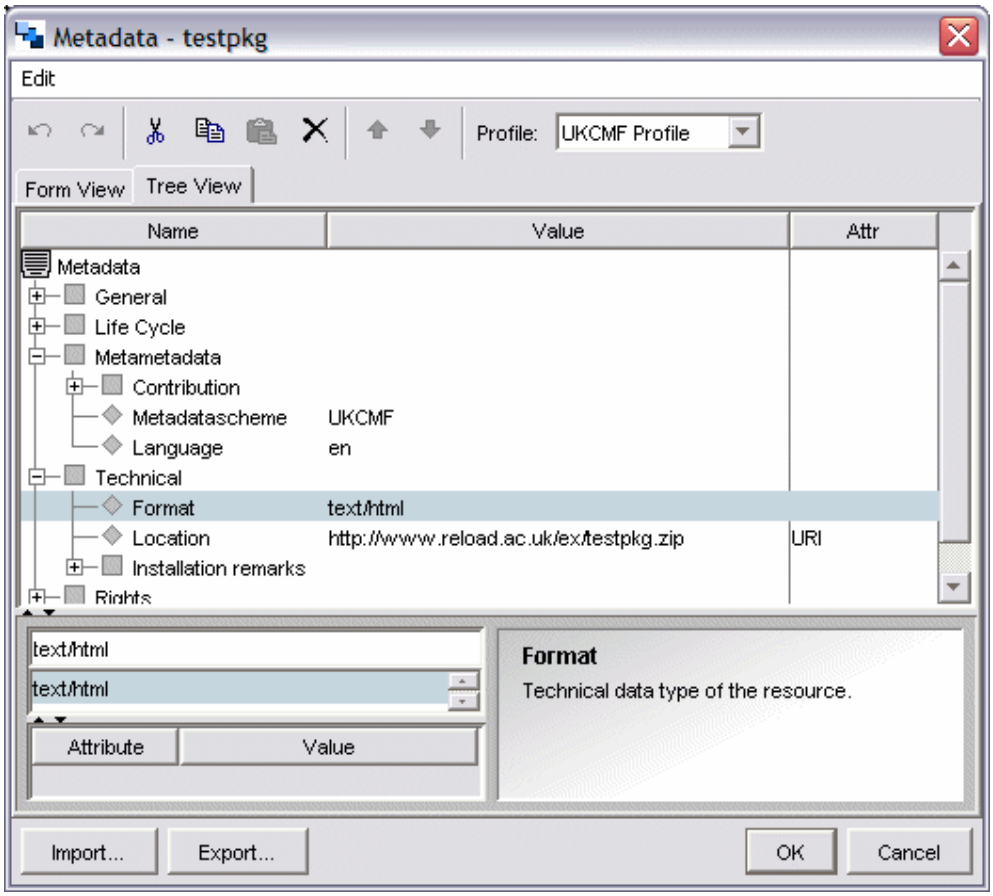


Figure Error! No text of specified style in document.**-2**
Figure 14

Save Content Package

Although the Content Package (as held in c:\cp\testpkg\) is now complete and valid, we still need to save it and store as a zip.

Saving the Content Package is simple - press the Save Icon, or select Save from the File menu. You can view the manifest again now: select the imsmanifest.xml file in the Resources Pane, and click on the view file icon in the Main toolbar. You will see the same structure that you can see in the resources pane, along with all the metadata.

Saving the Zipped Content Package is simple too. Click the 'Zip Content package' Icon (or select 'Zip Content Package' from the File Menu. in the dialogue that appears, type a filename (including extension - .zip) and choose a directory - c:\cp (ie the level above the content) is convenient.

Use Windows Explorer to navigate to the c:\cp directory. Find the file and open it with WinZip. You should see all the files you need have been included.

Save As Content Package

You can save a duplicate of your content package by using the 'Save As ...' option in the file menu. The main reason you might want to do this is if you want to create a package with the same structure as your current one, but wish to have different Unique Identifiers for 'identifier' attributes.

4 Metadata

We have already seen some of the metadata capabilities of the Editor in passing, but now we shall look more closely at some of the Metadata functionality.

Creating a Standalone Metadata File

Although primarily a Content Package editor, the Editor incorporates a fully fledged metadata editor and can be used to create and view metadata files which are independent of content.

To create a new Metadata file, Click File, New > IMS Metadata File. You will be asked which schema version to utilise (the default is 1.2.2, though you could choose v1.2.1 or v1.1) and whether you wish to be asked in future. Once chosen, a new blank metadata file is created and you can choose the desired application profile by clicking on the drop-down list. The editor automatically removes invalid combinations of schema and helper files.

Add Metadata as before using the form view or tree view (or a combination) until all necessary entries have been completed. Then click File, Save As … to open the Save dialog box. Enter a filename, and retain the .xml suffix. Save the file somewhere convenient. You can view the file in a web browser, where you will see that it is a valid IMS Metadata v1.2.2 record.

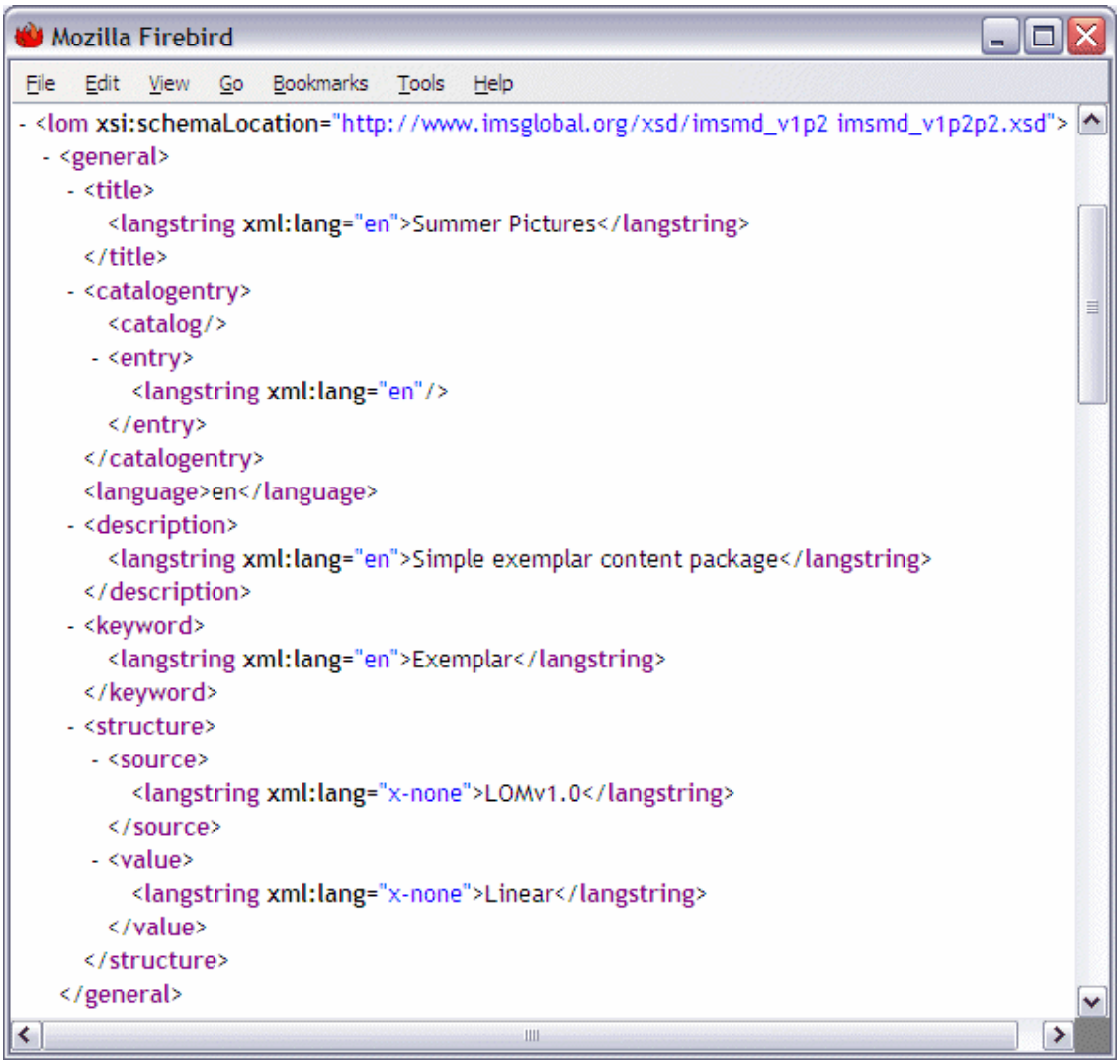


Figure Error! No text of specified style in document.**-3**
Figure 15

5 Customising Editor

By default, EDITOR supports the IMS Metadata v1.2.2 and Content Packaging v1.1.3 specifications (as well as SCORM v1.2). However in practise, individual organisations will subtly adapt these specifications to meet their own requirements - either by translating them into another language, or by customising the way they represent the specifications. For instance, when creating a metadata record for a content package, the core IMS specification is not a great deal of use as all the elements are optional - so a valid metadata record could consist of nothing more than an empty metadata element tag: <metadata /> If the eventual goal of adding a metadata record is interoperability (e.g. storage in a repository where your resource can be placed alongside other resources and retrieved using appropriate search criteria) then some rules (common practice) must be established governing:

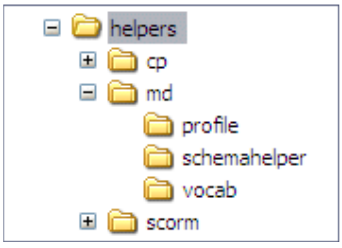
- which elements of the metadata specification are mandatory
  for instance it may be that you wish to ensure that every metadata record contains a description of the technical requirements for a resource.
- a vocabulary to use to describe content
  for instance for the element <coverage> it may be decided to restrict responses to a list of (United Kingdom, England, Scotland, Wales, Northern Ireland) from which anyone preparing content to your criteria must choose.

Such rules constitute an 'Application Profiles' and are normally set down by a community to ensure that resources created and used by that community (and shared, through object repositories) are interoperable.

In EDITOR, a standard profile is loaded by default. When you create a new Content Package or Metadata record, you can choose to load one of the supplied alternative profiles by clicking on the menu at the top left corner of each document and choosing from the drop down list. Note that if you are creating an IMS MD v1.1 record, only the default profile is offered (IMS MD 1.1 Profile). In addition to the default profiles, you can provide your own application profile, as described below (in detail fro Metadata, the same process would be used for IMS CP and SCORM Content Packages, though it is more common to customise the Metadata profile).

Metadata Profiles

The Editor itself doesn't know anything about individual application profiles and vocabularies, but will tailor the metadata form it displays based on any profile that it is supplied with. When you create a metadata file, a drop down box appears at the top left of the metadata form which allows you to specify a specific application profile. At present (v.1.2) three application profiles are included: IMS-LRM (the basic IMS Learning Resource Metadata profile), UKCMF v1.0 (UK Learning Object Metadata Common Framework) and LTSN v1.0, prepared by the Learning and Teaching Support Network - see the box below). The drop down list is populated automatically by the editor which reads the contents of a folder of helper files situated at: [USER HOME]\Editor\Editor-editor\helpers\md\profile\ (this is for a windows install. Typically, the path to USER_HOME will be C:\Documents and Settings\your_username\ ). You can add your own application profile here.



The metadata helper directory structure is as follows (and the content packaging and scorm helper directory structures are the same): There are three directories, profile, schemahelper and vocab. The profile directory holds the profiles themselves, and these refer to a schemahelper file (which provides help text etc.) and a vocabulary file (which provides the correct vocabulary for a given profile).

To add a new profile, you need to provide three XML files, the profile itself, a schemahelper, and (optionally - you could use an existing one) a vocabulary. You would normally base your profile on an existing profile - so the easiest way to create a new profile is copy the relevant files, rename them and edit them.

The text snippet below shows the first few lines of the IMS-LRM profile XML file and shows how it refers to the schemahelper and vocab files:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- EDITOR Metadata Profile -->
<!-- This is the default Profile - do not edit or delete! -->
<profile name="IMS-LRM" vocabfile="ims-lrm_en_vocab.xml"
        schemahelperfile="imsmd_en_helper.xml">
```

Once the files have been copied and renamed, edit the fourth line of your new profile file to reflect the 'profile name' - whatever you want to appear in the drop down list, and the names (if they have changed) of the 'vocab' and 'schema helper' files

Editing is a relatively straightforward process of going through the elements and sub-elements until you find the ones that are of relevance to you, then making the necessary change. The changes you might introduce could be to items listed in menus (vocabulary) or the language in which menu items are given. You may also wish to augment or replace the helper texts given - these may be used to give local guidance such as 'we recommend that for this item no more than two keyword are defined' or translated into a language other than English. The order in which elements appear can also be changed as the order of elements in the profile XML file defines the order on the metadata form: normally General, Lifecycle, Metametadata etc.

In the example below, a fourth profile 'Customised-LRM' has been included.

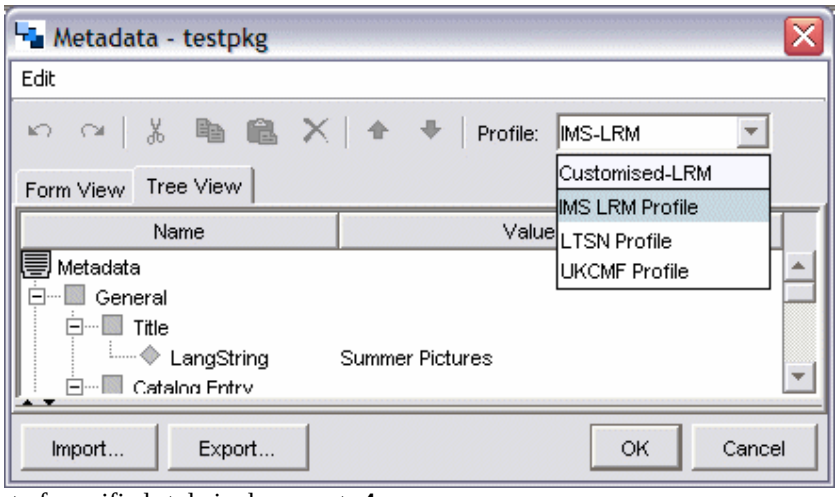


Figure Error! No text of specified style in document.**-4**
Figure 16

Once you have created a customised profile, you can set it to be the default profile by choosing View, Preferences ... and choosing the Metadata Preferences tab.

Pre-filling metadata fields is not currently possible in the Editor. If you want to use the same metadata every time a record is created then make a standalone metadata file with this information, save it to the helpers directory somewhere and import it every time a new metadata record is created (watch out - it will override existing metadata).

For the UK education community, an emerging application profile is being prepared under the auspices of CETIS and is entitled the UK Learning Object Metadata Framework (previously referred to as the UK Common Metadata Framework) For more information, see: http://www.cetis.ac.uk/profiles/uklomcor and in particular the document entitled: UK LOM Core v0.2 which effectively goes through the whole Learning Object Metadata specification and explains how the UK community interprets the various fields.

Metadata for Individual Resources

So far we have only discussed adding metadata to the Content Package as a whole. Metadata can in fact be added to individual items in the package in exactly the same way – just right click on the item you want to annotate.

Content Packaging Profiles

Profiling for IMS and SCORM Content Packages is achieved by editing the corresponding files in the cp and scorm directories under the helper tree.

Internationalization

Version 1.3 of Brihaspati Editor introduces the option of providing different language versions of Editor for different locales. You can store translations of the Editor Program messages, tooltips and Menu items in text helper files stored in the USER_HOME directory. The Preferences/Appearances tab allows you to choose the default locale for your system. If this is set to something other than the default (English) then Editor will look for the translated files and use these in preference to the inbuilt English messages and item names.

We will try and provide additional translation helper files for Editor through our web site. Further details on how to make your own files can be found in the help files under 'Internationalisation'.

6 SCORM Elements

A new feature in v1.1 of the Brihaspati Editor is support for editing SCORM 1.2 packages. SCORM (the Shareable Content Object Reference Model) builds heavily on the IMS Content Packaging specification and so IMS CP and SCORM v1.2 packages are very similar, but there a few important differences:

- SCORM packages can be of two types – a resource package and a content aggregation package.
- Metadata can be kept in an external file and referenced, rather than being kept within the Content Package as part of the main manifest.
- SCORM packages utilise 5 extra elements which provide information on how the package behaves. These five elements are:
    - Prerequisites
    - Max time allowed
    - Time Limit Action
    - Data from LMS
    - Mastery Score

This section will describe the SCORM v1.2 specific features.

Types of SCORM Package

SCORM Packages can be used for two purposes – either to store and transport content as 'assets' for future use, or as organisations of content for delivery via a Learner Management System LMS. SCORM Resource Packages are the simpler of the two package types – the manifest provides content metadata and describes the files which make up the resource, but the organization element is empty and no information about how the content is structured is kept as can be seen in the following screenshot of a valid SCORM resource package.

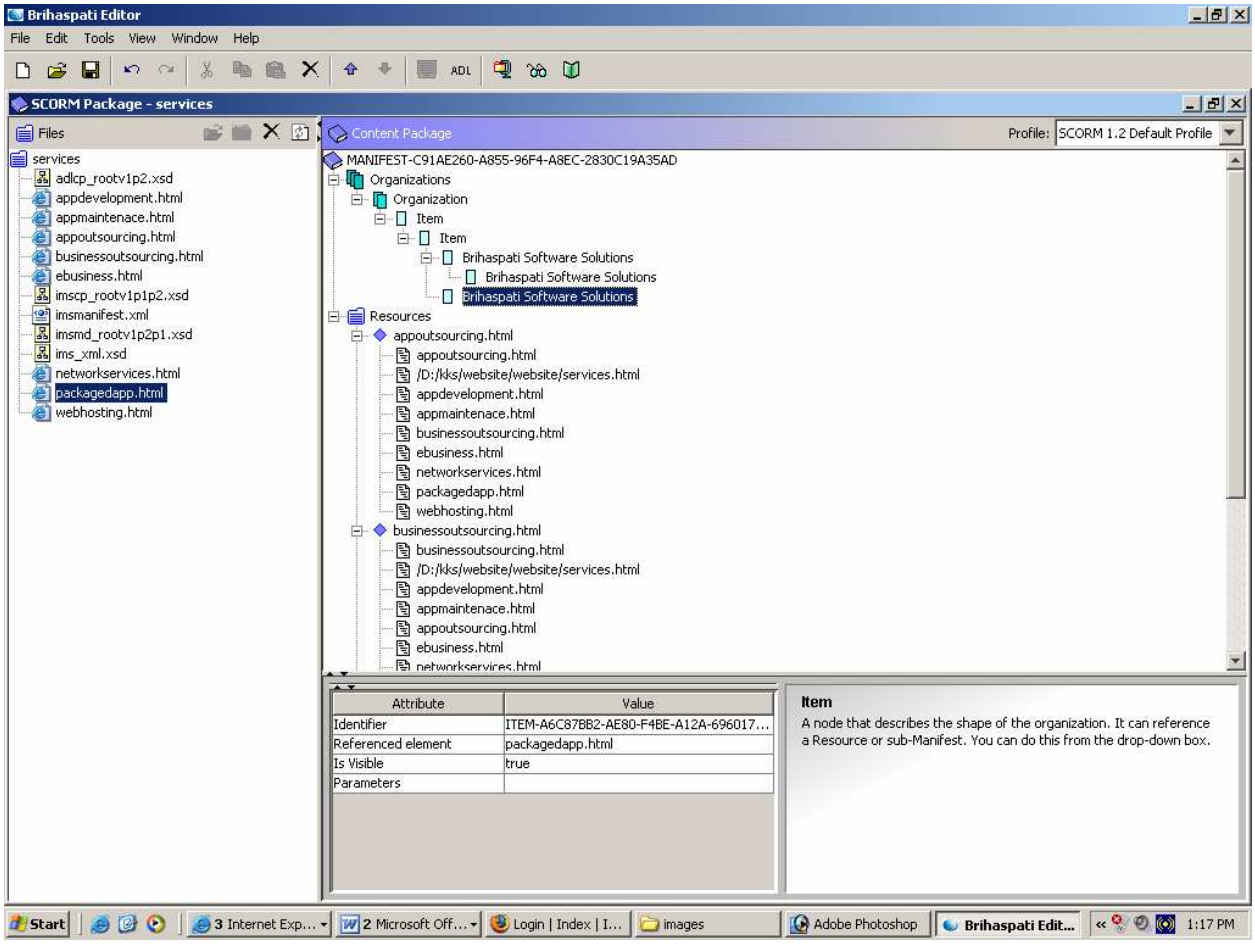


Figure Error! No text of specified style in document.**-5**

Figure 17

SCORM Content Aggregation Packages are more complex, and in this case at least one organization must be available to describe how the content is structured. SCORM Content Aggregation Packages are similar to IMS Content Packages, but can make use of the extra elements defined by the SCORM v1.2 specification (and see below).

SCORM Metadata

SCORM Metadata is based extensively on the IMS LRM specification, and ADL (Advanced Distributed Learning, who manage the SCORM initiative) do not impose a specific application profile. The difference between SCORM Metadata and IMS Metadata is that SCORM Metadata can be kept in a file external to the Content Package.

If you are creating a SCORM v1.2 package and wish to make use of externally held metadata, then you must use the ADL:Location element.

- Right click the Manifest as below and click Add Metadata
- Right click the Metadata element and click Add Schema – the node will appear pre-filled with ADL SCORM
- Right click the Metadata element and click Add Schema Version – the node will appear pre-filled with 1.2
- Right click the Metadata element and click Add Location – You can then type a URI for the metadata file in the box provided. You should of course make sure that the metadata file is valid XML.

SCORM Specific Elements

The main area where SCORM v1.2 differs from IMS CP v1.1.3 is in the support it provides for interaction with Learner Management Systems (LMSs). An extra five elements are defined which describe these interactions, and the LMS can use these elements to define a set of rules which control a learners progression through content to be defined. Each of these elements is optional.

prerequisites
This element defines any other Item which the learner must have completed satisfactorily before they are allowed to access the current Item. This can be used to define multiple routes through learning materials.

maxtimeallowed
This is the time (in seconds) that a LMS would permit access to the item for. For instance you may wish to restrict access to a simple assessment, forcing all learners to complete within 15 minutes.

timelimitaction
This is the action specified if the 'max time allowed' is exceeded. The user is provided with a choice of four built in actions:

- Exit, no message,
- Exit, message,
- Continue, no message,
- Continue, message.

datafromlms
This element allows initialisation data to be sent from the LMS to the Item on launch.

masteryscore
This is a normalised score (between 0 and 100) which must be achieved to indicate satisfactory completion of an item.

These elements are accessed by clicking the ADL icon on the toolbar. The editor pane for these elements is shown below

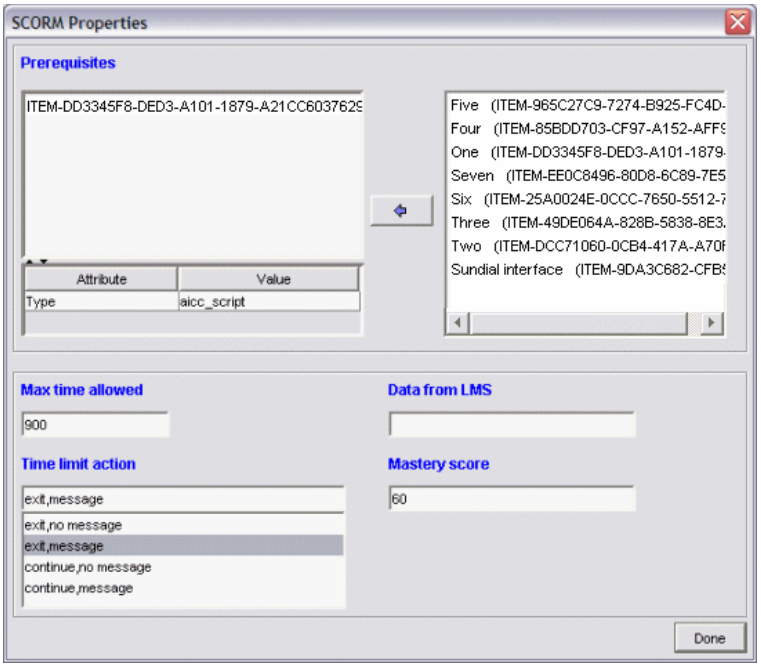


Figure 18

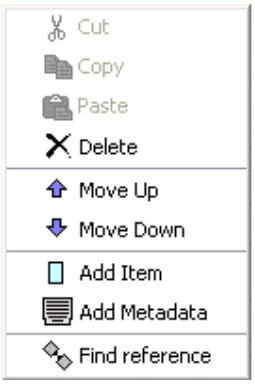
Once an Item carries SCORM information, its Icon changes from a simple rectangle to one containing a central spot. In addition a plus sign appears to the left side of the Item and when expanded, a list of the SCORM elements utilised appears. An expanded Item node is shown below:

7 Further Actions

References and Dependency

The Brihaspati Editor is essentially a tool for organising and reorganising content and the tools possess many simple features which assist the user in their task.

Finding References



In complex Content Packages, it can often be hard to match up Items and the resources they reflect. By selecting a given item and right-clicking, a context sensitive menu appears with the option Find Reference at the bottom (see screenshot). Clicking this option will move focus to the relevant resource in the resources tree. When an Item is selected the name of the associated resource also appears in the Attribute Pane. If you wish to reassign a resource to the item, simply click on the Referenced Element attribute and select a new resource from the drop down box which lists all current resources.

Adding Files and Dependencies

Although the Brihaspati Editor parses HTML files when adding them to the resources tree, it cannot exhaustively add all dependencies for all resource types. This is especially the case for pages which reference flash movies and java applets where the movie or applet may itself refer to content within compiled code.

In such cases, additional files can be added manually. By selecting a given resource and right-clicking, a context sensitive menu appears with the option Add File towards the bottom (see screenshot). Once selected, a new file icon appears at the appropriate point in the resources tree and the relevant value (a path and filename) can be added in the Attribute Pane below. In addition to single files, a given resource may well depend upon another resource. By selecting a given resource and right-clicking, a context sensitive menu appears with the option Add Dependency at the bottom (see screenshot). Once selected, a new dependency icon appears in the resources tree and the relevant resource can be selected from the drop down list in the Attribute Pane. The Find Reference option can be used to re-find the relevant resource for a given dependency as above.

Manifests and Sub-Manifests

The core element of a Content Package is the manifest, this is the organisational level which includes metadata and all the content within a given Content Package. As well as the Metadata, Organizations and Resources structure which we have already come across, a manifest can also include sub-manifests – effectively whole Content Packages, stored within itself. Sub-manifests can contain their own metadata and will of course contain organization(s) and resources. They can also contain further sub-manifests, allowing very complex file structures to be created. We use the terms aggregation and disaggregation to cover the processes of adding and removing sub-manifests from a Content Package.

Aggregation

Aggregation is the process of combining more than one Content Package manifest into a single Content Package. Right clicking on any manifest will display a content sensitive menu with Aggregate … as the final menu item. Choosing this option will launch a dialogue box which will ask you to choose the manifest (or zipped Content Package) to aggregate.

Although both these options are equally valid it is probably better practice (if more work) to aggregate the two sub-manifests side by side. Aggregating Content Packages can be a very

complex business and there is always a risk of files in individual sub-manifests having the same name (one overwrites the other) or non-unique identifiers (if you let Editor assign your unique identifiers it will always make sure they are unique). Ultimately however, the structure should be determined by the learning material.

Of course working with sub-manifests is more complex than working with simple Content Packages and many Managed Learning Environments and Repositories which are not yet fully IMS CP compliant will have problems with aggregated content. So Beware.

Disaggregation

Of course anything which has been aggregated can also be disaggregated. Right-clicking on any sub-manifest will display a content sensitive menu and the last option in the menu will be Disaggregate … Clicking Disaggregate … will launch a dialog box which will ask you to choose which directory to save the disaggregated content to. Once chosen, all the relevant files are copied across, including the manifest file. Within Editor, you can then Open this new Content Package as usual.