

# RELOAD

**Reusable eLearning Object  
Authoring and Delivery**

RELOAD Editor  
Introductory Manual

November 2003

v1.1.2a

The RELOAD project is supported by:

**JISC**

The Joint Information Services Committee

Under the X4L Programme: <http://x4l.virtualsite.co.uk/>

## Table of Contents

<b>Table of Contents</b> .....	<b>2</b>
<b>1 Introduction</b> .....	<b>3</b>
IMS Specifications .....	3
Where to get RELOAD: .....	3
This Guide.....	3
<b>2 RELOAD Editor</b> .....	<b>4</b>
System Requirements (MS Windows).....	4
Installation (MS Windows) .....	4
Uninstalling (MS Windows).....	4
RELOAD Workspace .....	5
Main Toolbar .....	6
Resource Pool Toolbar .....	6
<b>3 Tutorial</b> .....	<b>7</b>
Exemplar Content.....	7
Opening an Existing Content Package .....	7
Collect Resources Together .....	7
Create Directory Structure .....	7
Open RELOAD and Set Up Workspace .....	8
Add a Reference to Metadata .....	9
Adding Organizations and Items .....	9
Import Content .....	10
Create an Organisation.....	10
Add Items (Content) .....	11
Drag and Drop.....	11
View Content Package .....	13
Tidy Content Package .....	14
Add Metadata .....	15
Multiple Entries .....	15
Save Content Package.....	16
<b>4 Metadata</b> .....	<b>17</b>
Creating a Standalone Metadata File .....	17
<b>5 Supporting Other Application Profiles and Vocabularies</b> .....	<b>18</b>
Metadata Profiles .....	18
Metadata for Individual Resources .....	19
Content Packaging Profiles .....	19
<b>6 SCORM Elements</b> .....	<b>20</b>
Types of SCORM Package .....	20
SCORM Metadata .....	21
SCORM Specific Elements .....	21
<b>7 Further Actions</b> .....	<b>23</b>
References and Dependency .....	23
Finding References.....	23
Adding Files and Dependencies .....	23
Manifests and Sub-Manifests .....	23
Aggregation .....	23
Disaggregation .....	25

## 1 Introduction

The RELOAD Editor is a Content Package and Metadata editor. With RELOAD, 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 such as those maintained by the JORUM+ project. (<http://www.jorum.ac.uk/>)

The RELOAD 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).

## IMS Specifications

The RELOAD 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 RELOAD Editor supports v1.1.3 of the IMS Content Packaging specification and v1.2.2 of the IMS Metadata specification. v1.1 of the RELOAD Editor also offers the option to edit SCORM 1.2 packages. Information on SCORM is available from ADLNet: <http://www.adlnet.org/>

## Where to get RELOAD:

RELOAD is free software (released under the MIT Open Source license) and is available for download from the project web site at <http://www.reload.ac.uk/>

RELOAD is under active development, but since version v1.1 (11 August 2003), RELOAD has been stable and fully functional. Subsequent versions will provide improvements in usability and customisability.

## This Guide

This document has seven sections:

- this **Introduction** section provides some background to the RELOAD Editor tool:
- the **Getting Started** 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 RELOAD,
- the **Supporting Other Application Profiles and Vocabularies** section describes how to customise RELOAD to your individual needs,
- the **SCORM Elements** section describes the SCORM functionality,
- the **Further Actions** section describes some more advanced features of RELOAD.

The screenshots used in this document are taken from the v1.1.2a release of RELOAD Editor.

We are always keen to improve RELOAD and the documentation that accompanies it. And we are also keen to know who is using the tools and for what purpose. Please get in touch with your ideas, comments, suggestions and testimonies by sending email to:

[colin.milligan@strath.ac.uk](mailto:colin.milligan@strath.ac.uk)

Thank you

## 2 RELOAD Editor

The RELOAD Editor is a Java application, and should run on any platform capable of running Java applications. From the RELOAD web site, installation packages are available for the Microsoft Windows and Macintosh (OSX only) platforms. The install file for Windows includes a copy of the Java Runtime Environment (JRE) which is installed under the RELOAD Editor installation directory. This JRE (v1.4.1) should not interfere with other Java Virtual Machines installed elsewhere on your system. The Install for Macintosh does not include a JVM, but utilises the system JVM.

The information and instructions given here (and throughout this document) pertain to the Microsoft Windows version of the tool. Other versions should function in (much) the same way - though they may look different.

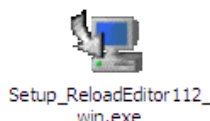
### System Requirements (MS Windows)

To run RELOAD Editor you should have a PC with at least the following specification:

- 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.

### Installation (MS Windows)

To install the RELOAD software, download the most recent install file from <http://www.reload.ac.uk/download/>. The download file is approximately 19Mb. This equates to a download time of around 80 minutes on a 56kbps modem (assumes sustained transfer rates of 4k/s). Obviously the download time will be much reduced if you use a broadband or an institutional connection.



Once download is complete, simply double click on the Installer and follow the on-screen instructions. The installation allows you to choose where to install the software. RELOAD Editor is installed to 'C:\Program Files\ReloadTools\ Editor\' by default, although you can of course change the location. Once installation is finished you can launch the RELOAD Editor by clicking on the Start Menu and finding its' Program Group:



### Uninstalling (MS Windows)

To remove RELOAD Editor from your system (and prior to upgrading to a new version of the software), click the 'Uninstall Reload Editor' Icon in the RELOAD program group.

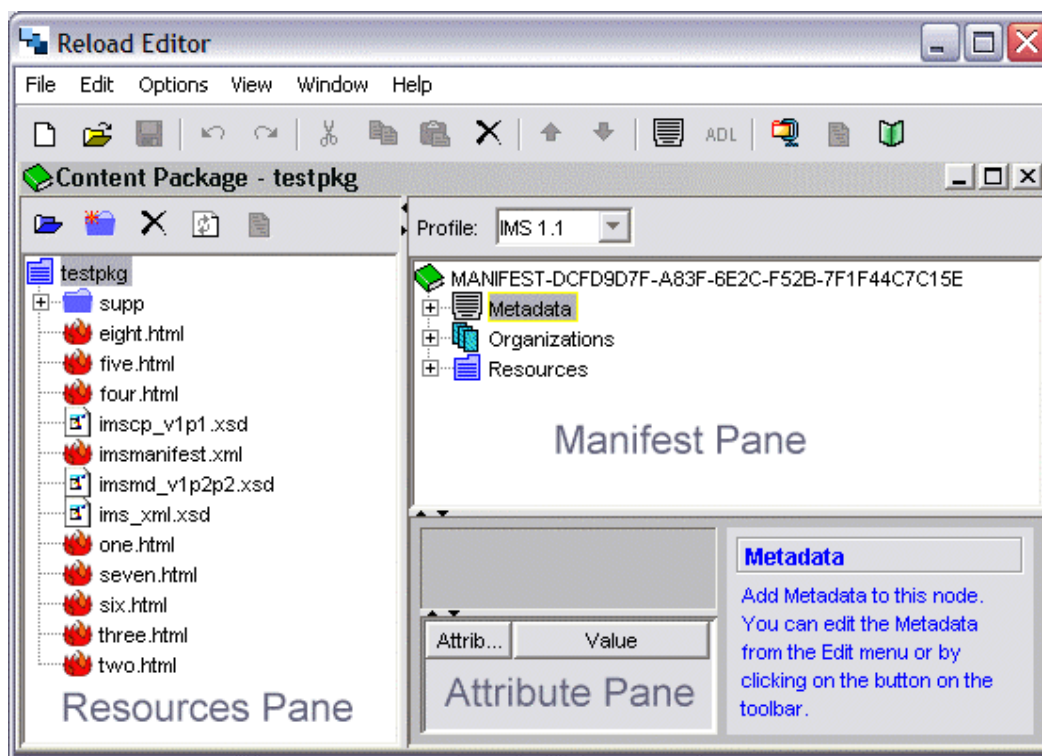


If you chose not to create a Program Group, you will find the relevant file at: C:\Program Files\ReloadTools\ Editor\UninstallerData\Uninstall Reload Editor.exe or wherever you installed to.

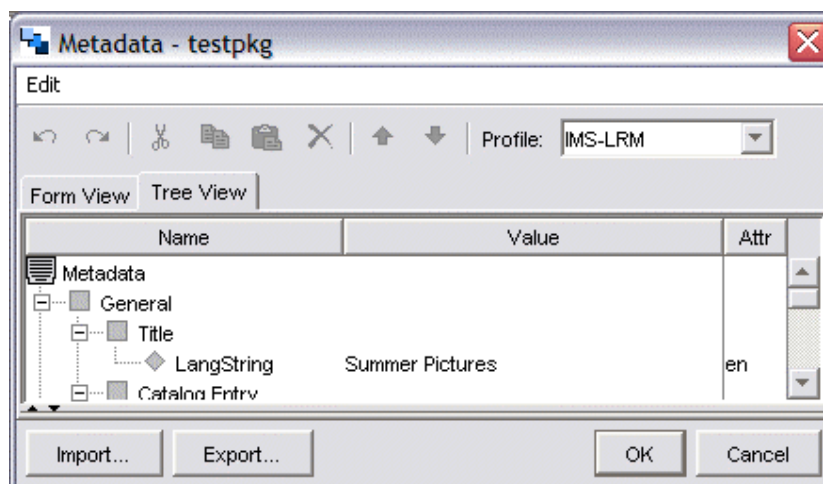
On uninstall, files and folders you have created in the RELOAD Installation directory will not be removed.

## RELOAD Workspace

The RELOAD Workspace basically consists of three panes: a resources pane (left), a manifest view pane (right), and an attribute view pane (bottom). The manifest pane is the key area as this space represents the structure of our 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 a box for data entry.



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.



















## Toolbars

The RELOAD 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 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
-  Create New Folder
-  Delete from Resource Pool
-  Refresh, refreshes the file list for the Resource Pool.
-  View File, launches a browser window to allow viewing of an individual file.

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

### 3 Tutorial

This section takes you through the creation of a simple Content Package.

#### Exemplar Content

To understand fully the facilities offered by RELOAD Editor, you must create and work with content packages. The pages which follow lead you through all the stages in creating a simple content package. If you wish to have a look at the content package that is created by the Content Package tutorial you can find the final file here:

<http://www.reload.ac.uk/ex/testpkg.zip> (140k). Either unzip the file and look at the individual files in your file browser, or use the RELOAD Editor to open this content package.

#### Opening an Existing Content Package

If you wish to open an existing content package choose File, Open from the File menu. This displays a dialogue box which prompts 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 directory for the Content Package. Dragging a Zip file (or imsmanifest.xml) onto the RELOAD workspace can be used to quickly open packages.

#### Collect Resources Together

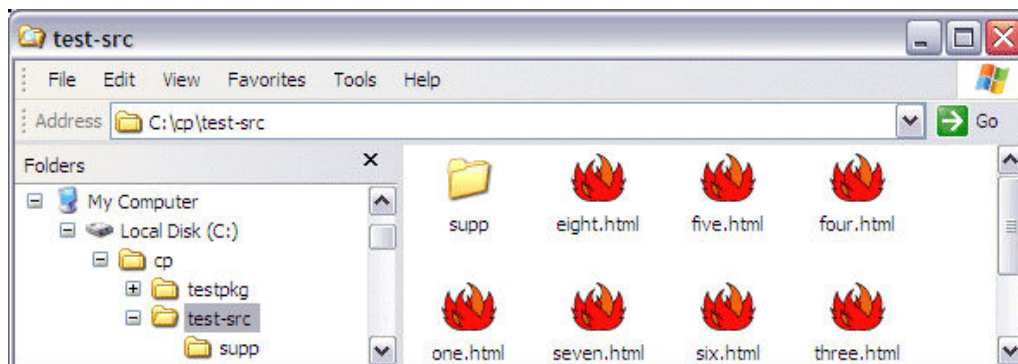
A Content Package is a means of transferring a set of resources from one location to another whilst retaining their structure and inter-relationships. When creating the content package we create a space in which all local files are kept - a root. As we build the content package, all the files we use will be stored under this working directory, 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 directory structure. If you wish to work through this tutorial yourself, you will probably set up something similar.

#### Create Directory Structure

Define a single directory for content packages on your computer (e.g. `c:\cp`). Within this directory, make a new directory called `test-src` (`c:\cp\test-src`). Within that directory, you will place all the files you need for your content package.

Download the source zip file <http://www.reload.ac.uk/ex/test-src.zip> (131kb) and extract to `c:\cp\test-src`. The directory should now hold 8 .html files, as well as a sub-directory (supp) containing 9 images and a stylesheet. Some UnZip utilities may have trouble with recreating the directory structure, if this occurs, you will need to recreate it yourself.

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

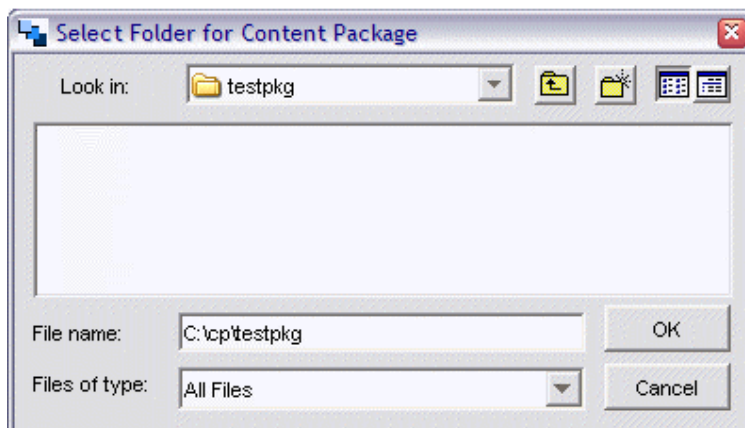




## Open RELOAD and Set Up Workspace

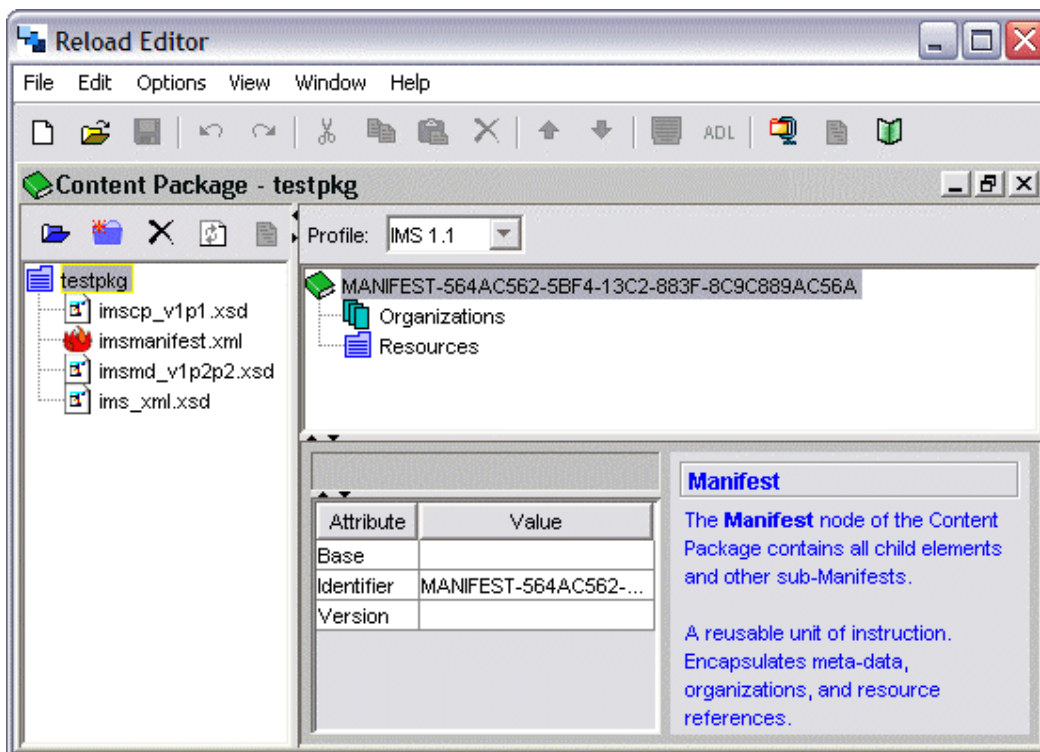
Launch RELOAD editor (either from the start menu or desktop shortcut). By default, the Editor opens up with no files open. we want to create a new Content Package, so:

Click File, New, > IMS 1.1.3 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 Open.



A new window appears with the title **Content Package - testpkg** and three frames showing:

- the tree structure of files and folders (the resources view),
- the main content package (the manifest view),
- information about individual elements (the attributes view) .



On creating the content package, the RELOAD 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.

- **imscp\_v1p1.xsd** - a local copy of the the content packaging XML Schema Document
- **imsmd\_v1p1.xsd** - a local copy of the 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 icon in the manifest pane and choose **Add Metadata**.

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

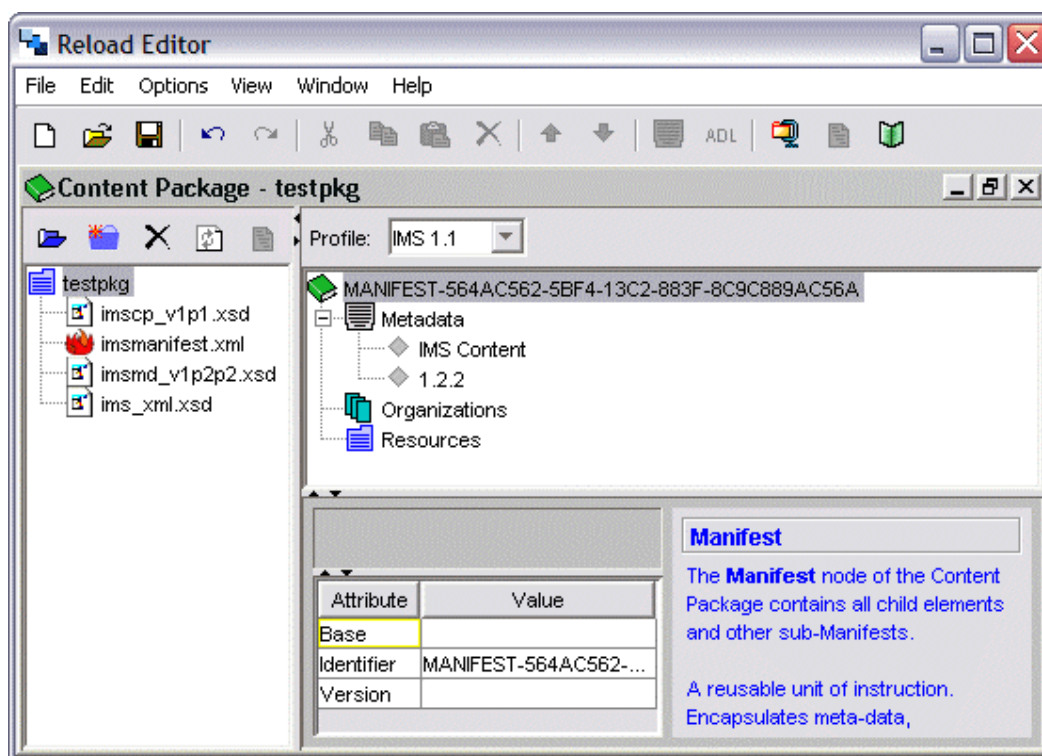
Select the schema and type `IMS Content` into the value field.

Right-click on the Metadata Icon 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 view should now look similar to the one shown below.



At any time you can view the `imsmanifest.xml` file by selecting it and clicking on the view file icon in the Resource Pool toolbar. This will show you the current structure if the manifest file.

## Adding Organizations and Items

Now we need to add some content. For our simple content package we have all our content stored in one place, but we need to Import the Resources into our resource pool.

## 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\).

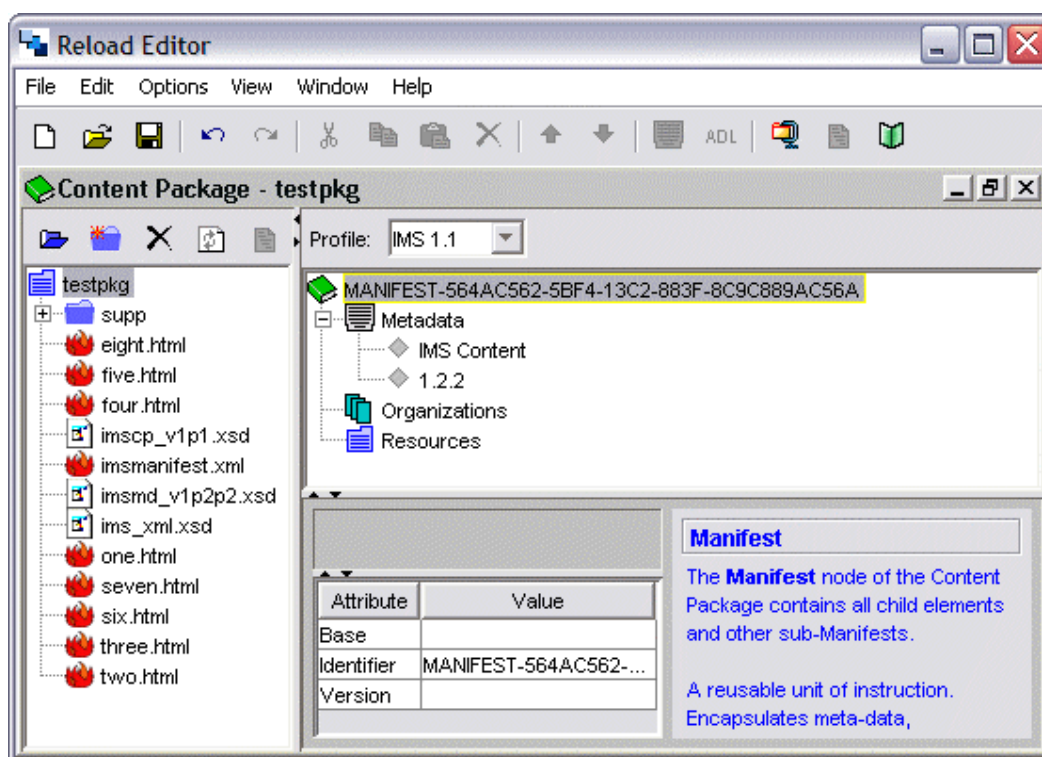
Select all the files, including the subdirectory.

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

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 RELOAD workspace. You should see the eight .html files in the root space, as well as the supp folder containing the images and stylesheets.

Your workspace should resemble that shown below:



## Create an Organisation

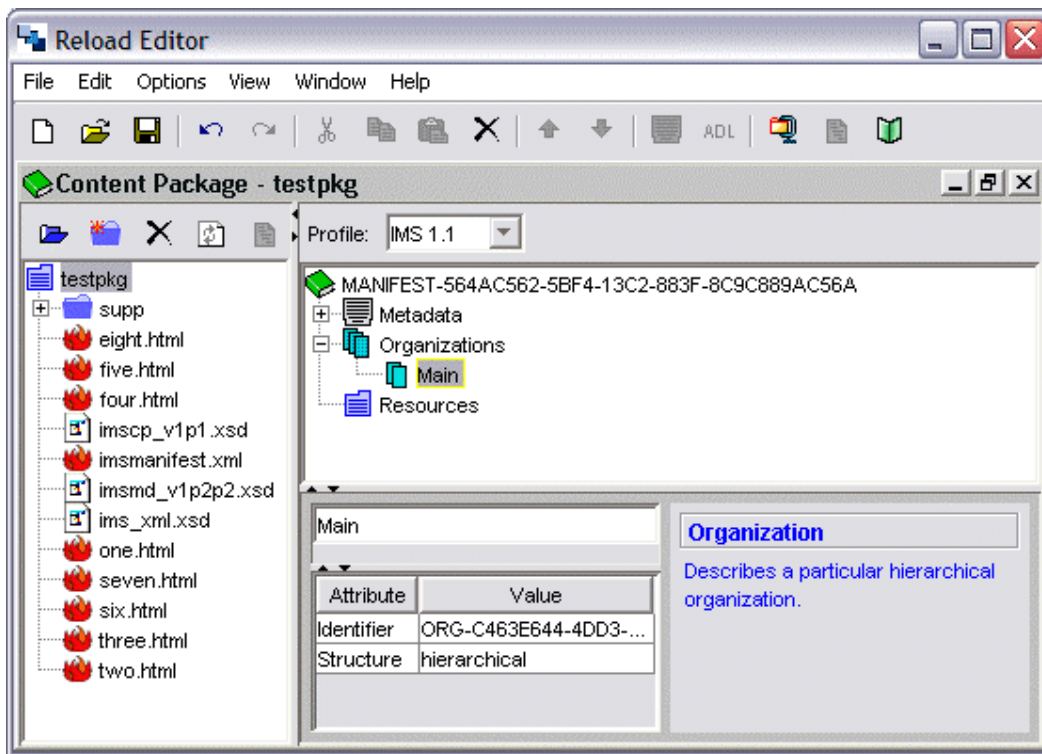
A content package consists of one or more organisations of content. By default the CP we create has no organisations, so we have to add one.

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

From the context sensitive menu, click 'Add Organization'

Select the new Organization and, in the attribute pane at the bottom of the workspace, enter the value `Main` in the white box. This defines this organization as the default for the package.

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).

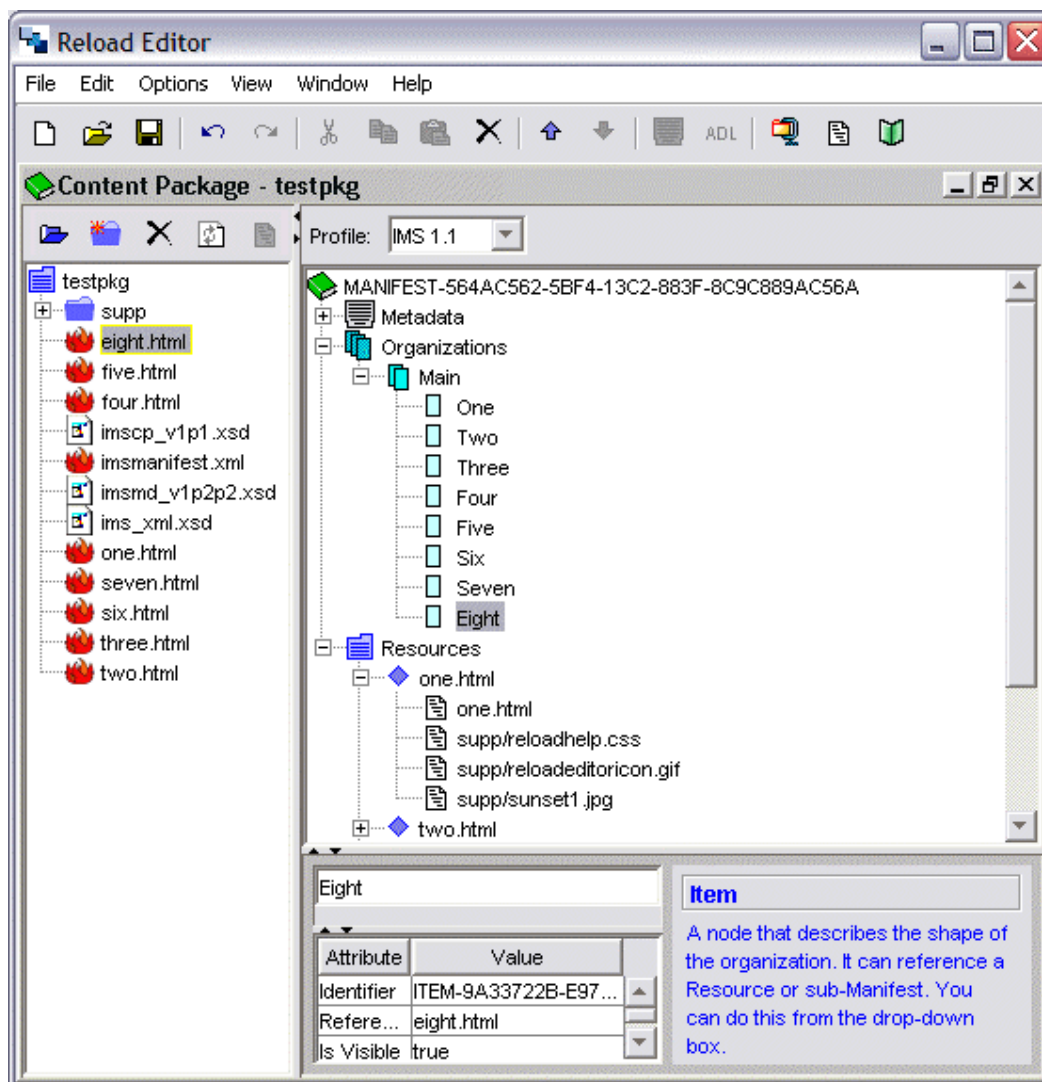
Drag the file to the 'Main' Organizations node of the manifest file. A new Item appears, with the title One. When the editor adds content to an organization it will attempt to provide a sensible name - for instance by parsing the html to read the contents of the file's <title> tag.

Notice that as you add content to the 'Main' Organizations node, 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 the order, 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 Organizations 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

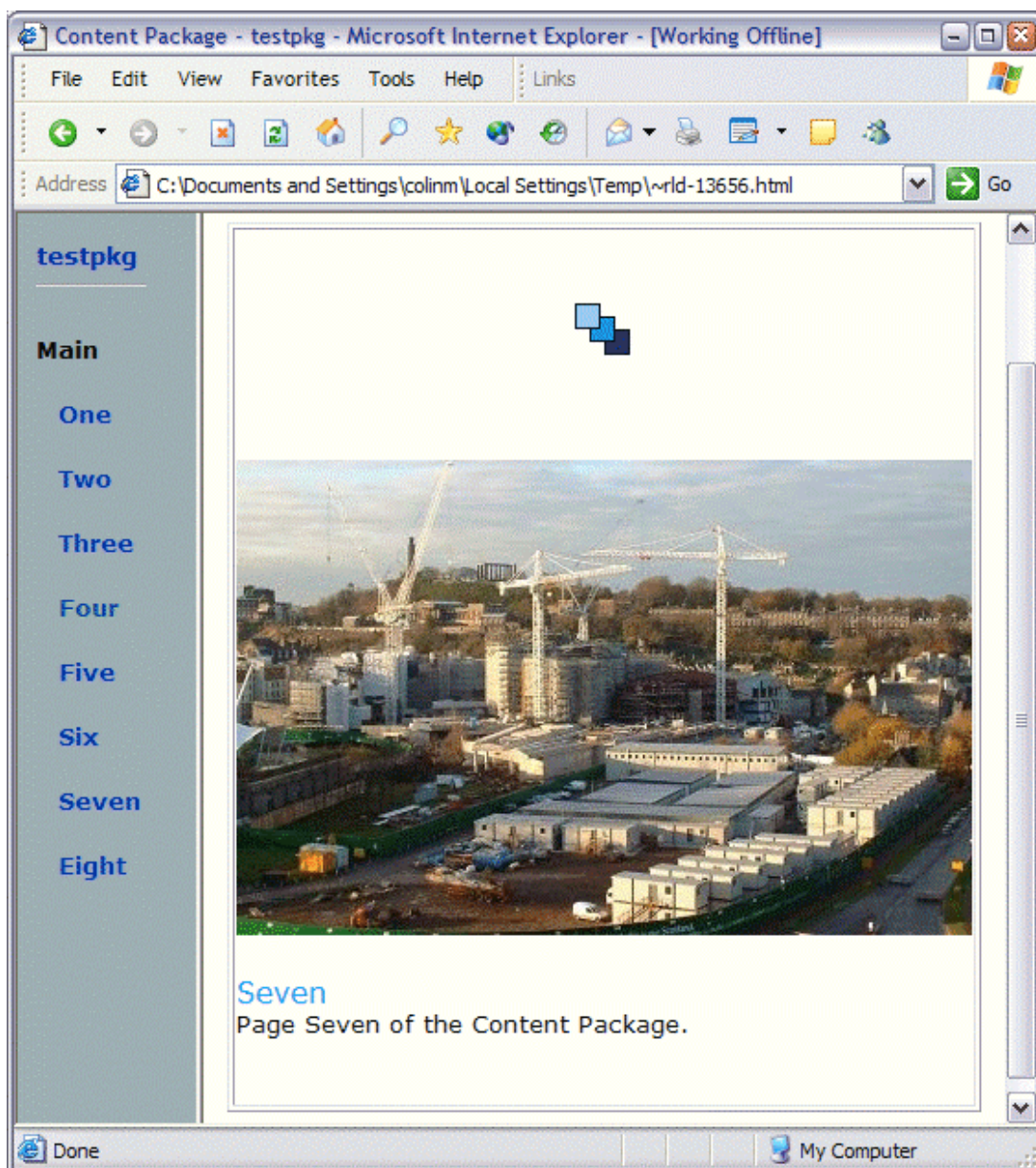
Once all content has been added to the main organization, you are ready to preview the content package in a web browser.

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 two-frame page will load, the left frame will correspond to the structure of content package you have defined, the right pane will initially be empty.

In the left pane, the top line represents the package name. Beneath the title, the first organisation 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.

Your previewed content package should resemble that shown below.



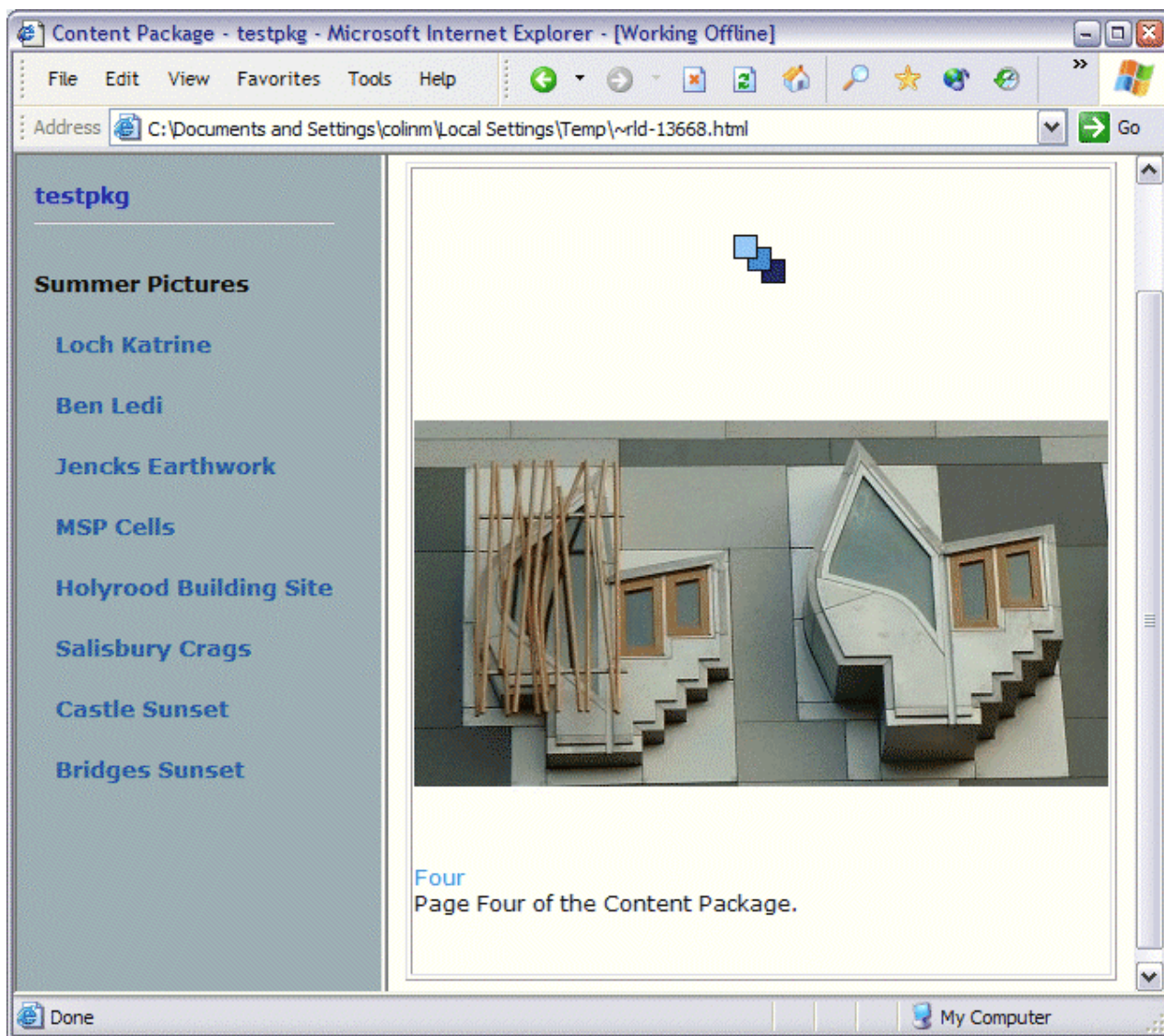
## Tidy Content Package

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.)

Preview Again, the Content Package previewer picks up the new name for the files.

Your previewed content package should resemble that shown below.



If you wish, 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.



## Add Metadata

The eventual destination of a content package will probably be within a Learning Management System (LMS) but it is also expected 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 draft Metadata Framework (UK Learning Object Metadata Framework: <http://www.cetis.ac.uk/profiles/uklomcore/>) has been produced and is in the process of being finalised (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 record, 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 1.0 - 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.

Name	Value	Attr
Metadata		
General		
Life Cycle		
Metametadata		
Contribution		
Metadatascheme	UKCMF	
Language	en	
Technical		
Format	text/html	
Location	http://www.reload.ac.uk/ex/testpkg.zip	URI
Installation remarks		
Rights		
Annotation		

text/html  
text/html

Attribute Value

Format  
Technical data type of the resource.

Import... Export... OK Cancel

## 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 effectively defined the filename (package location) at the start of the session. You can view the manifest again now: select the manifest.xml file in the Resources Pane, and click on the view file icon in the Resource Pool 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.

## 4 Metadata

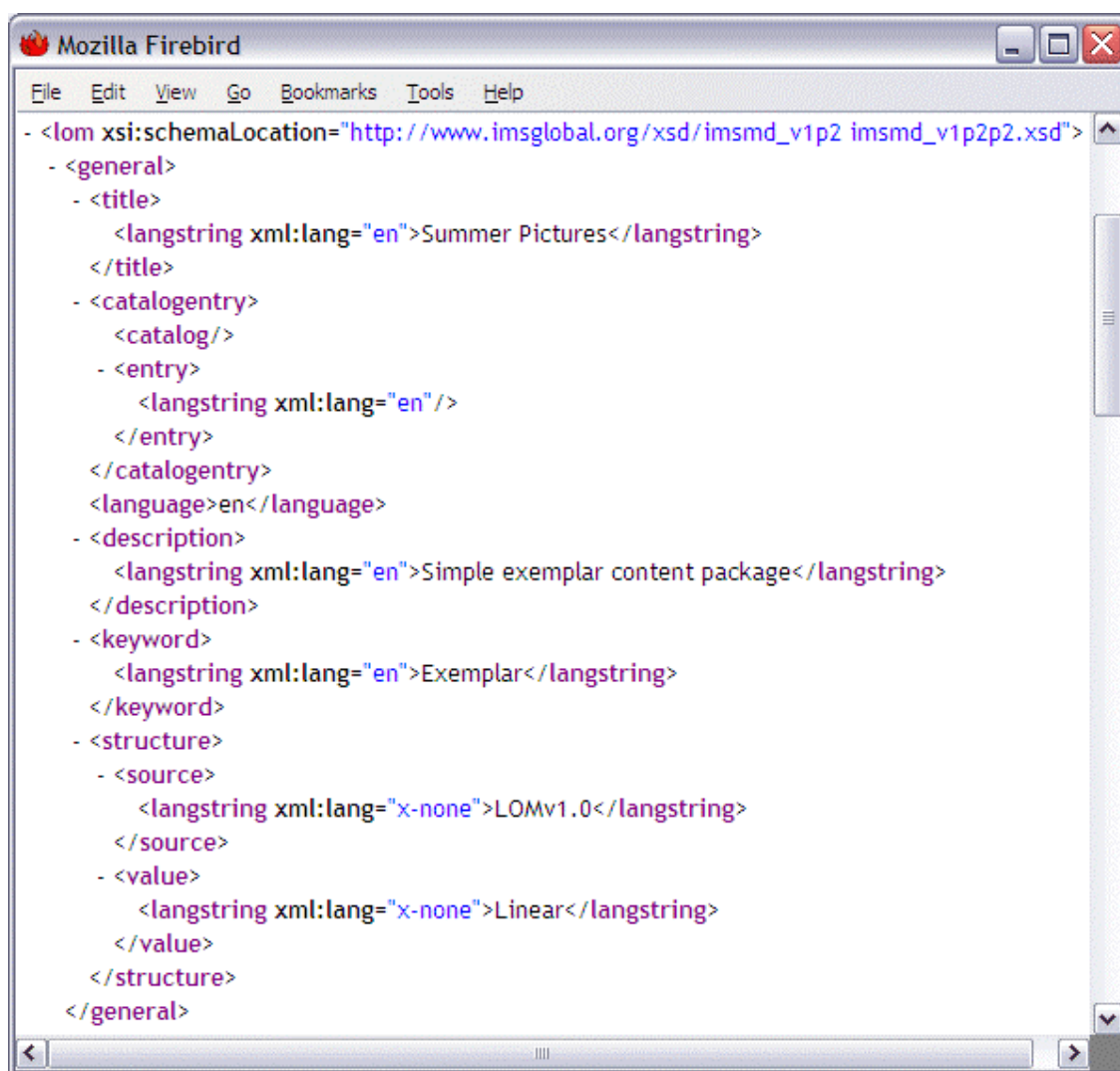
We have already seen some of the metadata capabilities of the RELOAD Editor in passing, but now we shall look more closely at some of the Metadata functionality.

### Creating a Standalone Metadata File

Although primarily a content package editor, the RELOAD 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 1.2.2 Metadata File. A new blank metadata file is created and you can choose the desired application profile by clicking on the drop-down list.

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.



## 5 Supporting Other Application Profiles and Vocabularies

By default, RELOAD 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. 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.

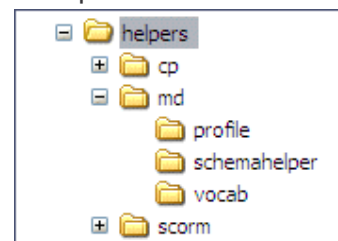
Such rules constitute an 'Application Profiles' and are normally set down by a community to ensure that resources created and used by that community (and shared, through object repositories) are interoperable.

In RELOAD, 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. Or you can provide your own application profile, as described below.

### Metadata Profiles

The RELOAD 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.1.1.1) 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 RELOAD editor which reads the contents of a folder of helper files situated at: `[RELOAD INSTALL]/helpers/md/profile/`. 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).

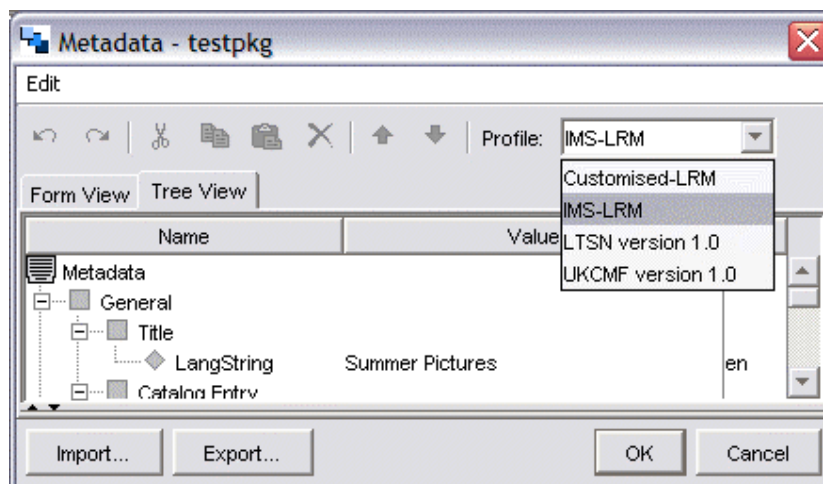


To add a new profile, you need to provide three XML files, the profile itself, a schemahelper, and (optionally - you could use an existing one) a vocabulary. You would normally base your profile on an existing profile - so the easiest way to create a new profile is copy the relevant files, rename them 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"?>
<!-- RELOAD Metadata Profile -->
<profile name="IMS-LRM" vocabfile="ims-lrm_en_vocab.xml"
      schemahelperfile="imsmd_en_helper.xml">
```

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 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.



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.

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/uklomcore> and in particular the document entitled: uklomcorev0p1.doc 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.



## 6 SCORM Elements

A new feature in v1.1 of the RELOAD 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 are a few important differences:

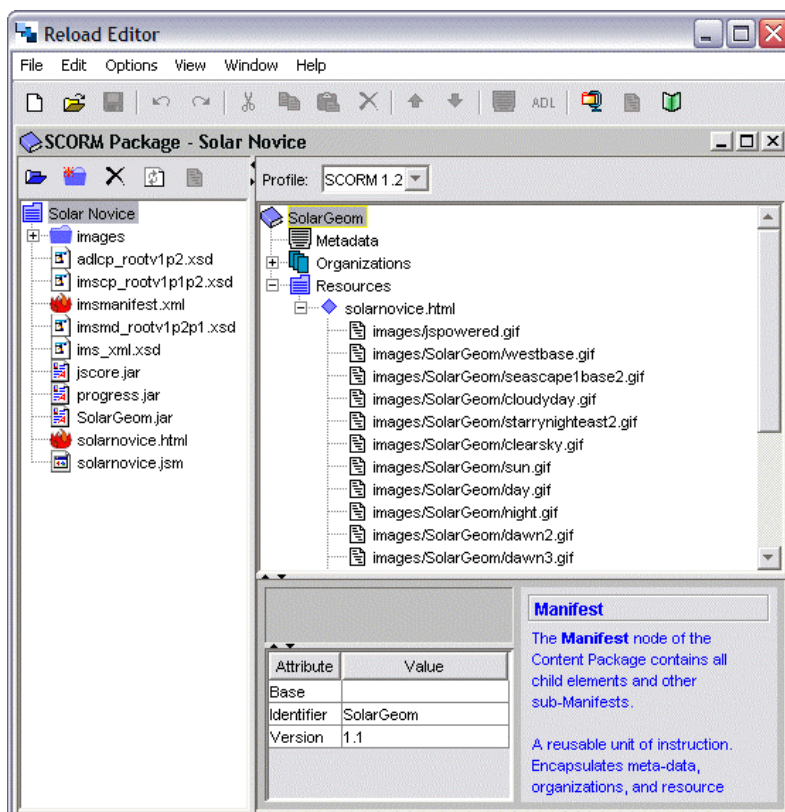
- SCORM packages can be of two types - a resource package and a content aggregation package.
- Metadata can be kept in an external file and referenced, rather than being kept within the content package as part of the main manifest.
- SCORM packages utilise 5 extra elements which provide information on 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.





**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.

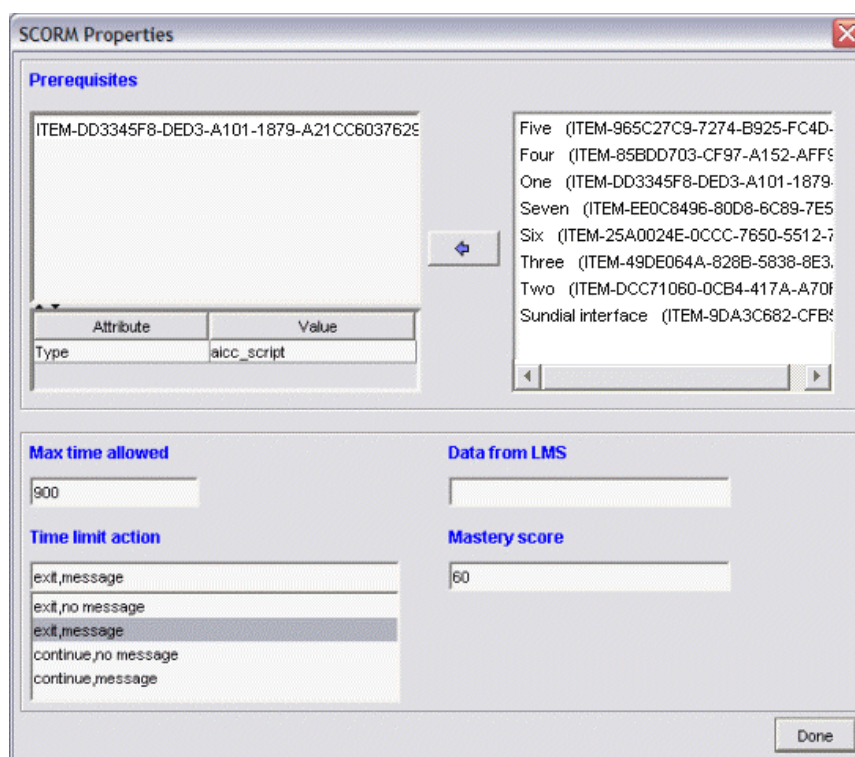
### datafromlms

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  icon on the toolbar. The editor pane for these elements is shown below

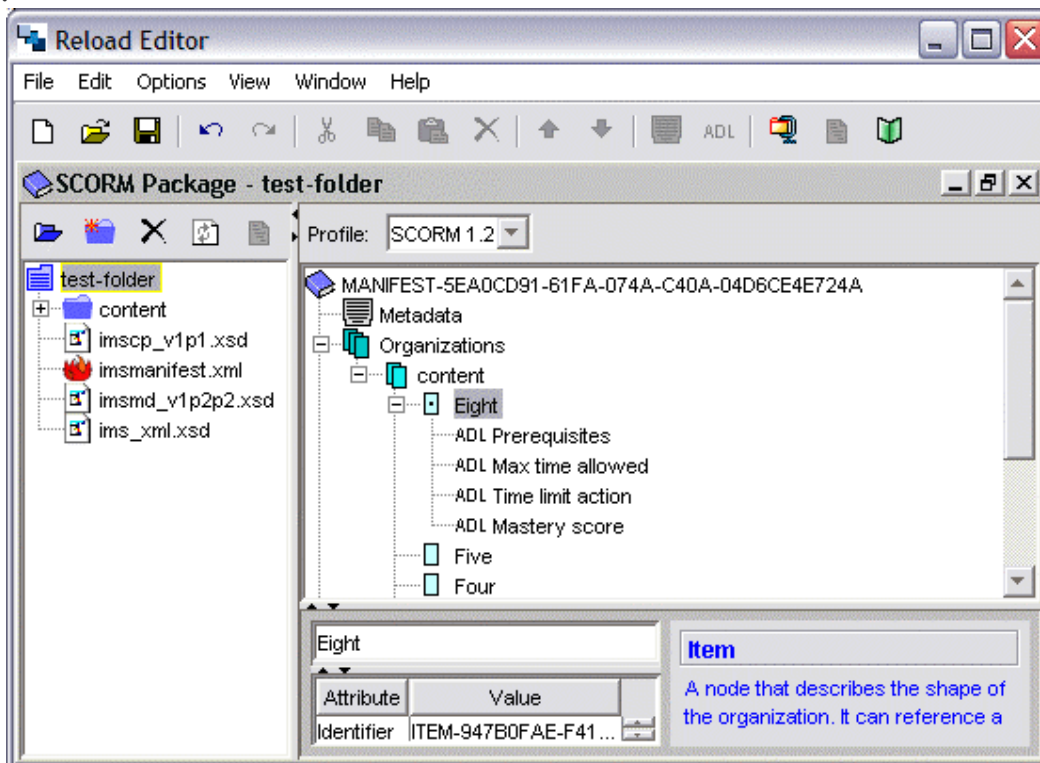


The SCORM Properties dialog box is shown with the following sections:

- Prerequisites:** A list of prerequisite items on the right and a table below.
 

Attribute	Value
Type	aicc_script
- Max time allowed:** A text field containing '900'.
- Time limit action:** A list box with options: 'exit,message', 'exit,no message', 'exit,message' (selected), 'continue,no message', and 'continue,message'.
- Data from LMS:** An empty text field.
- Mastery score:** A text field containing '60'.
- Done:** A button at the bottom right.

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:



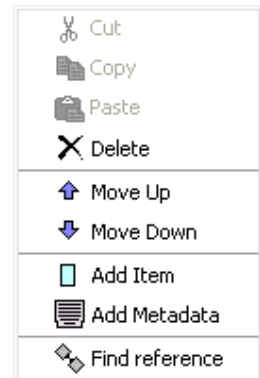
## 7 Further Actions

### References and Dependency

The RELOAD 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

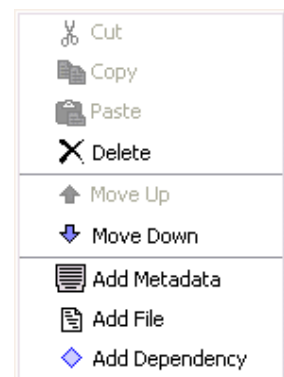
In complex content packages, it can often be hard to match up Items and the resources they reflect. By selecting a given item and right-clicking, a context sensitive menu appears with the option **Find Reference** at the bottom (see screenshot). Clicking this option will move focus to the relevant resource 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 RELOAD 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.



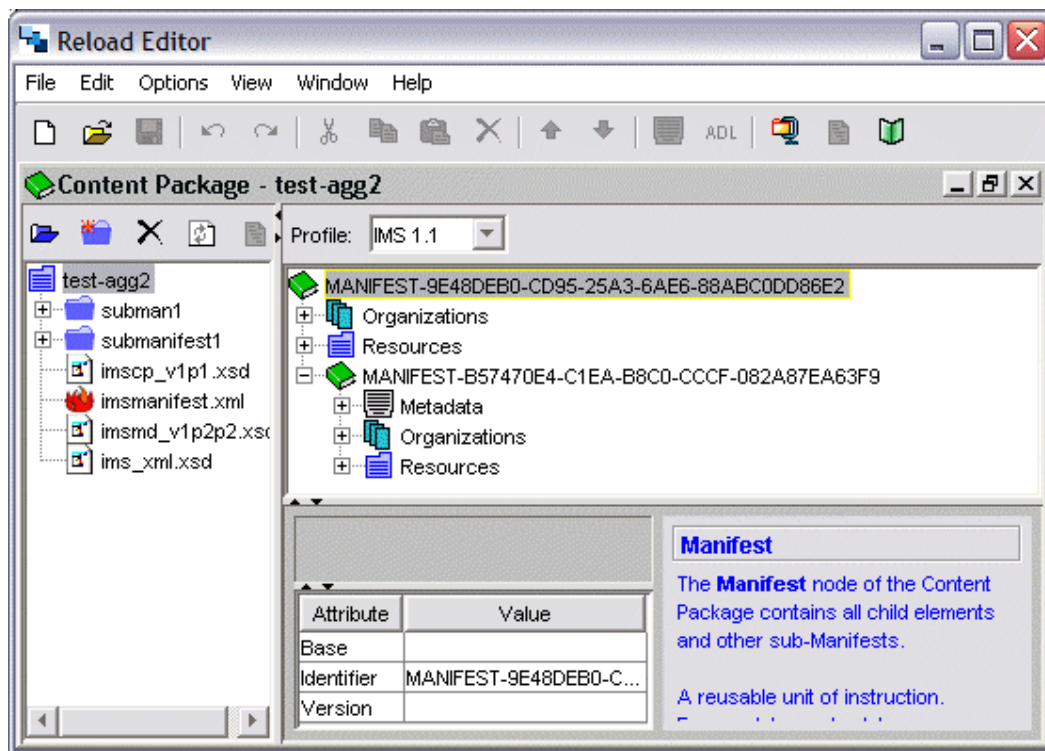
### Manifests and Sub-Manifests

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.

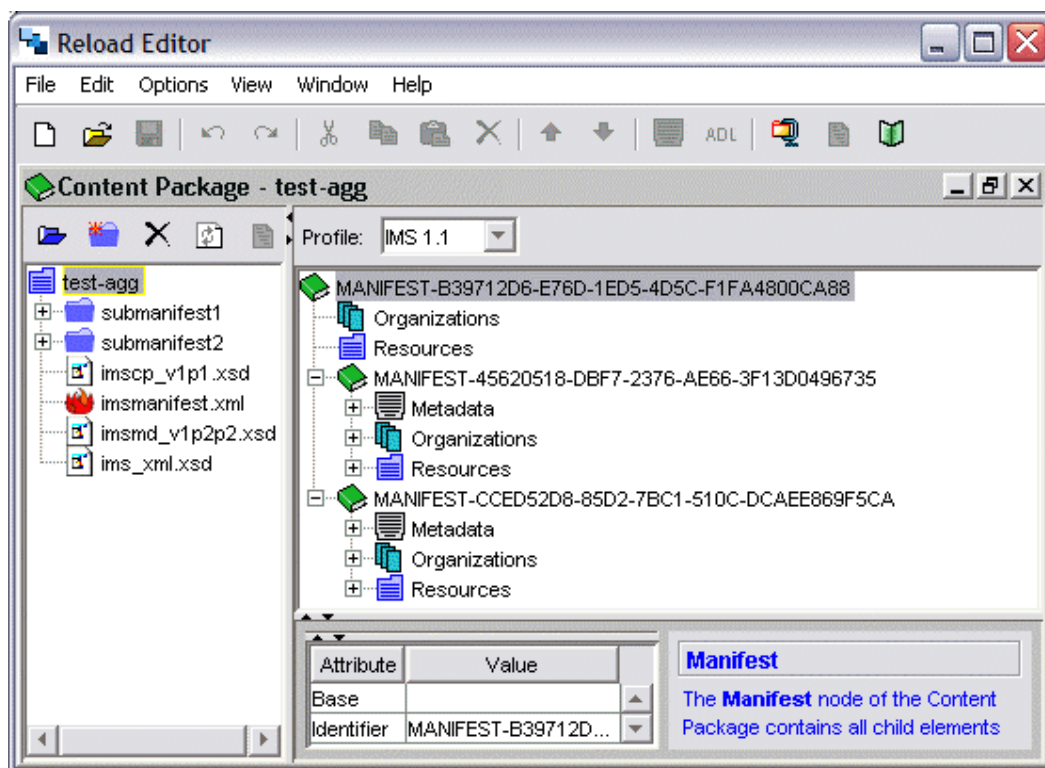
#### Aggregation

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.

There are two main ways to aggregate content: either as a sub-manifest inside a standard content package



or as two sub-manifests side by side within an otherwise empty Content Package.



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 Reload 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 Reload, you can then Open this new Content Package as usual.