



# Alfanet Authoring Tool User Manual

## Table of Contents

1. INTRODUCTION	4
1.1 ALFANET Background	4
1.2 Project Objectives	5
1.3 Authoring Tool	6
1.4 Standards	7
1.4.1 IMS Global Learning Consortium	7
1.4.2 IMS Learning Design	7
1.4.3 IMS Content Packaging	9
1.4.4 IMS-LOM (Resource Metadata)	10
1.4.5 Integration between Standards	11
2. INSTALLATION	13
2.1 Requirements	13
2.2 Installation Guidelines	14
3. TOOL STRUCTURE	18
3.1 Manage Global Resources	19
3.1.1 Options of Manage Global Resources module	20
3.1.1.1 New	20
3.1.1.2 Edit	22
3.1.1.3 Duplicate	22
3.1.1.4 Remove	23
3.1.1.5 Save Global Resources	24
3.2 Manage Global Metadata	26
3.2.1 Metadata Information Categories	26
3.2.1.1 General	27
3.2.1.2 Life Cycle	28
3.2.1.3 Metametadata	29
3.2.1.4 Technical	30
3.2.1.5 Educational	31
3.2.1.6 Rights	33
3.2.1.7 Relations	34
3.2.1.8 Annotations	35
3.2.1.9 Classifications	37
3.3 Learning Editor	39
3.4 Repository Interface	42
3.4.1 How to Configure the DAV Explorer	43
3.5 Configuration Options	45
4. LEARNING EDITOR SUB MODULES	49
4.1 File Manipulation Options	51
4.2 Create Package Interchange File Option	57
4.2.1 How to Install and Configure the WinZip Command Line Support Add-On	60
4.3 Items	62
4.4 Local Resources	67
4.4.1 Add to Local	68
4.4.2 Remove from local	68
4.4.3 Add all to local	70
4.4.4 Remove all from local	71
4.4.5 New Local Resource	72
4.4.6 Edit Local Resource	74
4.4.7 Make Local → Global	75
4.5 Title & Metadata	77
4.6 Learning Objectives	78
4.6.1 Items	79
4.6.2 Title & Metadata	80
4.7 Pre-Requisites	81
4.7.1 Items	82
4.7.2 Title & Metadata	82
4.8 Components	84
4.9 Roles	86
4.9.1 Learner Roles	87
4.9.2 Staff Roles	89
4.10 Environments	92
4.10.1 Services	98
4.10.1.1 Conference	102
4.10.1.2 Send Mail	103

4.10.1.3	Index/Search	104
4.10.2	Learning Objects	105
4.11	Activities	109
4.11.1	Learning Activities	110
4.11.1.1	Learning Activity Form	111
4.11.2	Support Activities	116
4.11.2.1	Support Activity Form	117
4.11.3	Activity Structures	120
4.11.3.1	Activity Structure Form	121
4.12	Method	126
4.13	Course Completion	129
4.14	Plays	132
4.14.1	Play Form	133
4.15	Acts	138
4.15.1	Act Form	140
4.15.2	Part's Form	143
APPENDIX 1	REFERENCES	145

## 1. Introduction

### 1.1 ALFANET Background

In the current competitive and global market there is a real need of continuously improve the training of employees in any organisation, in order to:

- offer a higher quality of service to their customers
- rapidly incorporate the new changing technologies of the sector. (University Research and Technology Transfer in a Changing World)
- achieve a greater level of fidelity of employees; to attract and retain top talent based on motivating people and achieving personnel satisfaction; adding social and community added value to the main strategic objectives.

At the same level that companies are using IT for improving their business processes, the new technologies can be especially adequate for improving the key process, which is training the Human Capital.

The current training needs of a company include:

- Changing training needs in the increasing competitive market: up-to-date and specialised training materials
- Needs of up-to-date, active and collaborative learning methods
- Different training depending on competence, personal skills and preferences of employees

There is a need for tools that can assist firms not only to instruct their employees but also assessing the level of performance acquired, evaluating the Human Capital of the company by means of some indicators.

## 1.2 Project Objectives

**ALFANET** concentrates in the new emerging market of e-learning, an area that will doubtless take advantage of the new technologies related with the internet, human interaction, and automatic learning. More specifically:

- **ALFANET** will allow organisation personnel to have an interactive, adaptive and personalised learning through the internet bringing them the opportunity to learn on those matters that are relevant to perform and to improve their work.
- **ALFANET** will allow organisations to improve their intellectual capital, promoting the evolution of employees in specialised and also multidisciplinary areas that are required in their daily work.
- **ALFANET** will allow learning content providers and educational centres to provide learning contents in such a way that these contents can be adapted to the personal needs.

Outcomes are:

- a working prototype of a LMS with intelligent personalisation capabilities and focused on continuous training needs in corporate firms.
- a proposal for contributing to the current standards of the e-learning field in the adaptation and personalisation on contents and learning methods.
- an evaluation of the prototype's impact on the current learner needs
- A description of the project results in terms of internal usage by the consortium and a set of components for potential exploitation by third parties (e-learning service providers)

The project will have a significant contribution to knowledge in the area of adaptive e-learning, e-learning standards usage, and the potential exploitation of some of the components by e-learning service providers.

## 1.3 Authoring Tool

The **Authoring Tool** is part of the **Alfanet** LMS prototype and its objectives are to provide an easy and user friendly interface to create and edit courses for the Alfanet System.

The Authoring Tool also has a module of integration with the Alfanet Repository and communicates with it between WebDav protocol.

The courses are based in a standard named IMS Learning Design that is explained in the Standards section.

The tool was developed in VB.Net based on a peer-to-peer application called Groove Workspace that give us several communication facilities and it's intended to be a collaboration platform for working teams. For more information please visit the following page:

<http://www.groove.net>;

**Note:** *The present version of this manual refers to the Level A tool, so any Level B or C concept or feature will not be available to the user and it's out of the scope of this manual.*

## 1.4 Standards

The Alfanet courses are based on three standards of Instructional Management Systems (IMS) Global Learning Consortium. The first standard (*IMS-LD*) is related to pedagogical definition of the course. The second standard (*IMS-CP*) is about the wrapping of the course and every related resources in a standardized organization. The last standard (*IMS-LOM*) is about the meta information about the course and the enclosed elements. These three standards combined make possible to create a complete definition of a course.

### 1.4.1 IMS Global Learning Consortium

In the IMS Global Learning Consortium (*IMS*) we can find the following description about what is the IMS:

*“The IMS Global Learning Consortium develops and promotes the adoption of open technical specifications for interoperable learning technology. Several IMS specifications have become worldwide de facto standards for delivering learning products and services. IMS specifications and related publications are made available to the public at no charge from [www.imsglobal.org](http://www.imsglobal.org). No fee is required to implement the specifications.*

*IMS is a worldwide non-profit organization that includes more than 50 Contributing Members and affiliates. These members come from every sector of the global e-learning community. They include hardware and software vendors, educational institutions, publishers, government agencies, systems integrators, multimedia content providers, and other consortia. The Consortium provides a neutral forum in which members with competing business interests and different decision-making criteria collaborate to satisfy real-world requirements for interoperability and re-use.”* In <http://www.imsglobal.org/aboutims.cfm> (2003/11/28)

### 1.4.2 IMS Learning Design

The IMS Learning Design (*IMS-LD*) is the core standard of the Alfanet System. The *IMS-LD* it's a xml standard that defines a language to assemble courses and use a terminology that are drawn from the metaphor of theatre with several equivalent concepts such as Play, Act, Role , Role Part, etc...

In the *IMS-LD* specification (*IMS Learning Design Information Model*) we have the following definition about the Learning Design:

#### **“Conceptual Model**

*The model shows that learning design provides a semantic view of a collection of resources on one hand, and on the other hand it integrates a method, specifying the dynamic aspects of the learning design.*

*The model shows three levels of semantic aggregation (the three horizontal layers of grey colored classes). The semantically highest level is the learning design, it aggregates a collection of components, objectives/prerequisites (short for: learning objectives & prerequisites), and a method. The lowest levels of aggregation are the resource, play, condition, and notification. The resources are aggregated into components and objectives/prerequisites. The plays, conditions, and notifications are aggregated into the method.*

*A component can be one of seven different types: role, property group, property, activity structure, activity, environment, or outcome. With the exception of outcome, these are all elements in the LD Information Model. Role can be one of two types: learner or staff.*

*A resource can be one of five different types: web content, imslid content, person, service facility, or dossier. These resources can be referenced from a Learning Design but are not explicitly part of the Information Model.*

*Specific types of components are bound to specific types of resources. The moment in time when the resource is bound in the learning design differs:*

*Example: The textual description of a learning objective for instance, can be written during the design and delivered as a fixed resource with the learning design. However also an absolute URL can be provided to a location where the file can be edited at any time.*

*The resources that are not bound during design time are not part of this specification. This applies to concrete persons and to the actual dossiers of persons. However, the roles that persons can take into the learning design and the properties that have to be present in the dossiers are part of the learning design.*

(...)

### **Conceptual Structure of the Learning Design**

The core concept of the Learning Design Specification is that regardless of pedagogical approach, a person gets a role in the teaching-learning process, typically a learner or a staff role. In this role he or she works toward certain outcomes by performing more or less structured learning and/or support activities within an environment. The environment consists of the appropriate learning objects and services to be used during the performance of the activities. Which role gets which activities at what moment in the process, is determined by the method or by a notification.

**Note:** most of the concepts mentioned above are reflected in the information model, but some only exist at the conceptual level (person, outcome).

The method is designed to meet learning objectives (specification of the outcomes for learners), and presupposes certain prerequisites (specification of the entry level for learners). The method consists of one or more concurrent play(s); a play consists of one or more sequential act(s) and an act is related to one or more concurrent role-part(s), each role-part associates exactly one role with one activity or activity-structure. The teaching-learning process is modeled in the method on the notion of a theatrical play. A play has acts, and in each act has one or more role-parts.

The acts in a play follow each other in a sequence (although more complex sequencing behavior can take place within an act). The role-parts within an act associate each role with an activity. The activity in turn describes what that role is to do and what environment is available to it within the act. In the analogy, the assigned activity is the equivalent of the script for the part that the role plays in the act, although less prescriptive. Where there is more than one role-part within an act, these are run in parallel.

(...)

The explicit roles specified in this language are those of learner and staff roles. Each of these can be specialized into sub-roles, but no vocabulary is put forward for this. It is left open to the learning designer to name the (sub)-roles and specify their activities. For example, in simulations and games different learners can play different roles, each performing different activities in different environments.

Activities can be assembled into activity-structures. An activity-structure aggregates a set of related activities into a single structure, which can be associated to a role in a role-part. A structure can model a sequence or a selection of activities. In a sequence, a role has to complete the different activities in the structure in the order provided. In a selection, a role may select a given number of activities from the set provided in the activity-structure. This can, for instance, be used to model situations where students have to complete two activities, which they may freely select from a collection of e.g., five activities contained in the activity-structure.

(...)

Environments can contain two basic types:

- *Located learning objects*, typically specified by a URL with optional metadata. A user may further classify these learning objects by means of the vocabulary provided in the IMS-LOM Meta-Data (5.2 Learning Resource Type) or the generic 'class' attribute that is available on all elements. In EML [LD1], the learning objects are classified in the following types: knowledge-objects, tool-objects, and test-objects.
- *Generic services*. A service relates to a concrete service facility available at runtime. During design a service has no URL assigned to it, but must be given a URL when the Learning Design is instantiated at runtime. Examples of a Service include a discussion forum, chat rooms, monitoring tools, search facilities, etcetera. In Learning Design the conditions for setting up a service at runtime are specified at an abstract level. For example, for discussion groups it specifies which learning design roles have what type of access (participant, observer, moderator, etc.).

**Note:** if a discussion forum is to be used within a learning design, were it given a predefined URL, then all instances of the Unit of Learning that includes the learning design, wherever and whenever instantiated, would have the same one specific discussion forum. While this may lead to serendipitous learning, it is probably not what was intended by the learning designer! (However, should this be desirable, and then a normal Resource element with a fixed URL can be provided.)

For this version of the specification, the types of services specified are limiting these to those that are now found in typical LMS systems. It is possible to inherit from a generic service and thus specify new types as extensions to the vocabulary. As many services are targeted for use in a specific instance of a learning design, the actual members will need to have been assigned to roles before the service is instantiated. As



different roles may have different permissions within a service, there is the facility to specify these within the particular service definition.

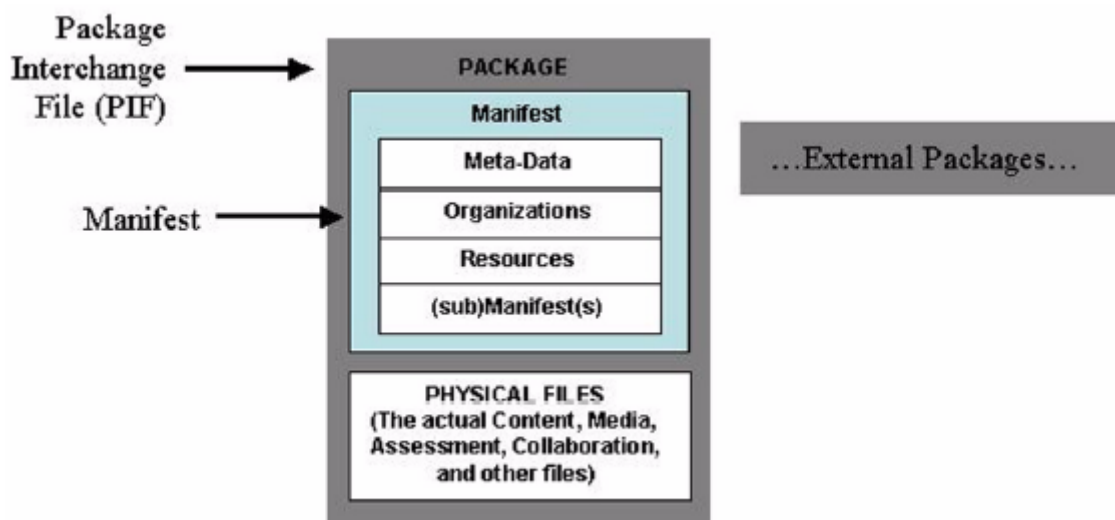
EML included a complete content vocabulary based on the OASIS DOCBOK specification [LD12]. For Learning Design it was decided not to include any content specification, but let the users of the Learning Design Specification decide which one to use. In order to allow for runtime interactions with the end users, specific global learning design elements are provided separately at level B which can be namespaced into any XML-based content schema. A suggestion is to use XHTML for content and to namespace the global elements of learning design into XHTML.”

### 1.4.3 IMS Content Packaging

In the IMS Content Packaging (IMS-CP) Information Model we can find the following description about what is the IMS-CP, the objectives, structure and elements of this standard:

“The IMS Content Packaging (IMS-CP) Information Model describes data structures that are used to provide interoperability of Internet based content with content creation tools, learning management systems (LMS), and run time environments. The objective of the IMS-CP Information Model is to define a standardized set of structures that can be used to exchange content. These structures provide the basis for standardized data bindings that allow software developers and implementers to create instructional materials that interoperate across authoring tools, LMSs, and run time environments that have been developed independently by various software developers.

(...)



IMS Content Packaging scope.

The IMS Package depicted in the above Picture consists of two major elements: a special XML file describing the content organization and resources in a Package, and the physical files being described by the XML. The special XML file is called the IMS Manifest file, because course content and organization is described in the context of ‘manifests’. Once a Package has been incorporated into a single file for transportation, it is called a Package Interchange File. The relationship of these parts to the content container is described below:

**Package Interchange File** – a single file, (e.g., ‘.zip’, ‘.jar’, ‘.cab’) which includes a top-level manifest file named “imsmanifest.xml” and all other physical files as identified by the Manifest. A Package Interchange File is a concise Web delivery format, a means of transporting related, structured information. PKZip v2.04g (.zip) is recommended as the default Package Interchange File format. Any ZIP file format MUST conform to RFC1951.

**Package** – a logical directory, which includes a specially named XML file, any XML control documents it references (such as a DTD or XSD file), and contains the actual physical resources. The physical resources may be organized in sub-directories.

- *Top-level Manifest* – a mandatory XML element describing the Package itself. It may also contain optional (sub) Manifests. Each instance of a manifest contains the following sections:
- *Meta-data section* – an XML element describing a manifest as a whole;
- *Organizations section* – an XML element describing zero, one, or multiple organizations of the content within a manifest;
- *Resources section* – an XML element containing references to all of the actual resources and media elements needed for a manifest, including meta-data describing the resources, and references to any external files;
- *(sub) Manifest* – one or more optional, logically nested manifests;
- *Physical Files* – these are the actual media elements, text files, graphics, and other resources as described by the manifest(s). The physical resources may be organized in sub-directories.

**Package** – A Package represents a unit of usable (and reusable) content. This may be part of a course that has instructional relevance outside of a course organization and can be delivered independently, as an entire course or as a collection of courses. Once a Package arrives at its destination to a run time service, such as an LMS vendor, the Package must allow itself to be aggregated or disaggregated into other Packages. A Package must be able to stand alone; that is, it must contain all the information needed to use the contents for learning when it has been unpacked.

Packages are not required to be incorporated into a Package Interchange File. A Package may also be distributed on a CD-ROM or other removable media without being compressed into a single file. An IMS Manifest file and any other supporting XML files required by it (DTD, XSD) must be at the root of the distribution medium.

**Manifest** – A manifest is a description in XML of the resources comprising meaningful instruction. A manifest may also contain zero or more static ways of organizing the instructional resources for presentation.

The scope of manifest is elastic. A manifest can describe part of a course that can stand by itself outside of the context of a course (an instructional object), an entire course, or a collection of courses. The decision is given to content developers to describe their content in the way they want it to be considered for aggregation or disaggregation. The general rule is that a Package always contains a single top-level manifest that may contain one or more (sub) Manifests.

The top-level manifest always describes the Package. Any nested (sub) Manifests describe the content at the level to which the (sub) Manifest is scoped, such as a course, instructional object, or other.

For example, if all content comprising a course is so tightly coupled that no part of it may be presented out of the course context, a content developer would want to use a single manifest to describe that course's resources and organization.

However, content developers who create "instructional objects" that could be recombined with other instructional objects to create different course presentations would want to describe each instructional object in its own manifest, and then aggregate those manifests into a higher-level manifest containing a course organization. Finally, a content developer who wants to move multiple courses in a single Package (a curriculum), would use a top-level manifest to contain each course-level manifest and any instructional object manifests that each course might contain.

**Resource** – The resources described in the manifest are physical assets such as Web pages, media files, text files, assessment objects or other pieces of data in file form. Resources may also include assets that are outside the Package but available through a URL, or collections of resources described by (sub) Manifests. The combination of resources is generally categorized as content. Each resource may be described in a <resource> element within a manifest's XML.

This element includes a list of all the assets required to use the resource. The files included in the Package are listed as <file> elements within such <resource> elements."

#### 1.4.4 IMS-LOM (Resource Metadata)

IMS Learning Object Metadata (IMS-LOM) or simply Resource Metadata it's the standard that defines the metadata associated with the learning materials used in the Authoring Tool.

Metadata in a simple way to define it's the "data about data", or for other words, *it's the information that helps us to catalog, identify or describe the contents of the learning materials.*

We can find the following definition about Metadata in the *IMS Learning Resource Meta-Data Best Practice and Implementation Guide*:

*"Descriptive labels can be used to index learning resources to make them easier to find and use. Such labels are "data about data" and are referred to as "meta-data". An example of meta-data is the label on a can of soup, which describes the can's ingredients, weight, cost, and so forth. Another example is a card in a library's card catalog, which describes a book, its author, subject, location within the library, and so forth."*

A meta-data specification makes the process of finding and using a resource more efficient by providing a structure of defined elements that describe, or catalog the learning resource, along with requirements about how the elements are to be used and represented.

The *IMS-LOM* defines several kinds of information about the object such as:

- *general;*
- *lifecycle;*
- *metametadata;*
- *technical;*
- *educational;*
- *rights;*
- *relation;*
- *annotation;*
- *classification;*

These items are described in more detail in the "Manage Global Metadata" section of this manual.

## 1.4.5 Integration between Standards

In the *IMS-LD* Information Model we can find the following description of the integration between *IMS-LD* and *IMS-CP*:

### ***"Unit of Learning = IMS Content Package + IMS Learning Design***

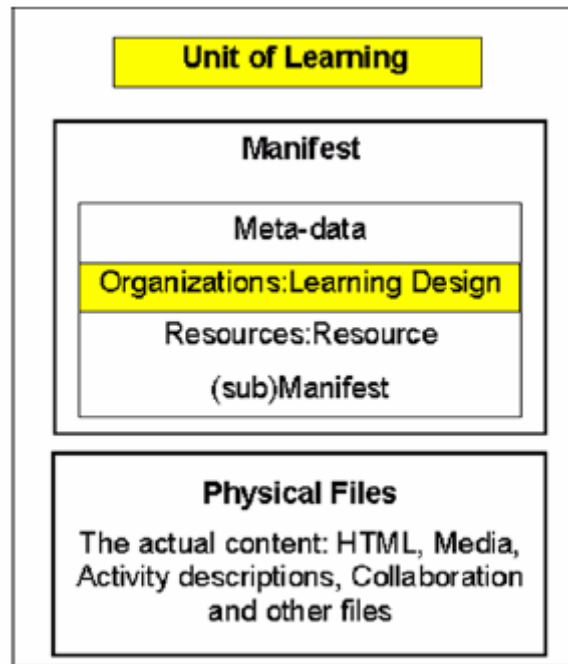
*The primary use of IMS Learning Design is to model units of learning by including an IMS Learning Design in a content package, preferably – but not necessarily - an IMS Content Package. In this specification it is assumed that IMS Learning Design is being used with IMS Content Packages to model units of learning. How this is done is explained in this section.*

*IMS Content Packages describe their contents in an XML document called the 'package manifest'. The Manifest may include structured 'views' into the resources contained in that package; each 'view' is described as a hierarchy of items called an 'organization'. Each item refers to a Resource that, in turn, can refer to a physical file within the package. It can however also refer to an external resource.*

*(...)*

*The Manifest is the information structure defined in the Content Packaging Specification. It is contained within a package as an XML file with a fixed, pre-defined name ("imsmanifest.xml"). This enables it to be found amongst the many other content files that may be contained in a package.*

*(...)*



The integration of a Learning Design into the Content Packaging Structure is set out in the above Figure.

To create a unit of learning, IMS Learning Design is integrated with an IMS Content Package by including the learning design element as another kind of organization within the <organizations> element, using the standard namespace for Learning Design. When the standard namespace is “[standard-namespace-for-learning-design]”, then learning design elements are included as follows (ignoring irrelevant elements and attributes):

```

<manifest>
<metadata/>
<organizations>
<learning-design xmlns="[standard-namespace-for-learning-design]">
[add learning design elements here]
</learning-design>
</organizations>
<resources/>
</manifest>
  
```

The italics have to be filled in with the appropriate namespace and elements respectively.

In a package that includes a learning design element, the optional organization element within organizations is ignored.

This mechanism is in conformance with the extensibility mechanisms IMS Content Packages provide. If an organizations element contains a learning design element, any ‘organization’ element in the same organizations element is ignored and only the learning design element is read by the runtime system. Where other content organization elements are desired, they can be included in sub manifests, as sub packages may be aggregated in the same way as in normal content packages.”

About the integration between IMS-LOM and the others standards we can say simply that every metadata sections of the IMS-LD and IMS-CP will be filled with information that follows the IMS-LOM standard.

## 2. Installation

### 2.1 Requirements

#### Operating System:

- *Microsoft® Windows® 98*
- *Microsoft Windows ME*
- *Microsoft Windows NT® 4.0 (with Service Pack 5 or later)*
- *Microsoft Windows 2000*
- *Microsoft Windows XP (Designed For)*

**Note:** *With the release of Groove Workspace 2.0, Groove no longer supports Microsoft Windows 95.*

#### Hardware Requirements:

- *Intel® Pentium® processor, 400 MHz or higher*
- *128 MB RAM for Groove Workspace, 256 MB RAM for synchronizing a Groove® Mobile Workspace with a Microsoft® SharePoint™ site.*
- *100 MB free disk space, with additional space required for your data.*
- *Display resolution 800 x 600, 15-bit (32,768) colour minimum*
- *Sound card, speakers, and microphone required to use voice features*

#### Software Requirements:

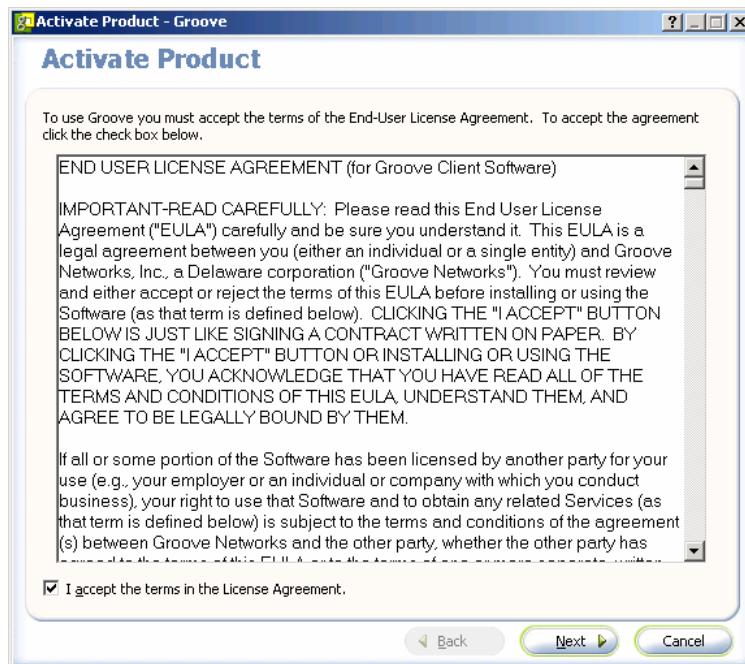
- *Microsoft Internet Explorer version 4.0 or later required, version 5.5 or later recommended.*
- *Internet Explorer 6.0 or later required to use the Groove Forms Tool, Groove Mobile Workspace for Microsoft® SharePoint™ or Groove Web Services.*

#### Internet Connection:

- *56 kbps modem minimum - LAN (local area network), DSL, or cable modem preferred.*

## 2.2 Installation Guidelines

- **Download Groove Workspace Preview Edition from:**  
<http://www.groove.net/downloads/groove/>
- **Install Groove (following the Wizard).**
- **Start Groove and Create an Account – Following the wizard (filling e-mail, name and choosing a password).**



Fill the e-mail data

**Activate Product - Groove**

## Activate Product

### Welcome To Groove

Before you can use Groove Workspace, you need to take a minute or so to set up an account. First, we'd like to ask you a couple of questions:

**Many people find our Groove monthly bulletin very useful.  Yes  No**  
**Would you like to subscribe to it?**

After you activate the Preview version of Groove, you will receive some "Getting the Most out of Groove" emails containing helpful hints and other information you may find useful. Each message includes instructions for opting-out of receiving future messages.

**Email Address:**

**In what format would you prefer your emails?**  Text  HTML  
(AOL users choose Text)

**Optional:** How did you hear about Groove software?

[Having difficulties? Click here to report them](#)

Choose username and password

**Activate Product - Groove**

## Activate Product

The name you enter here is how other Groove users will recognize you.

**Account Name:**

Enter a passphrase for your new account. Pick a passphrase that is both secret and reasonably long.

**Passphrase:**

**Confirm passphrase:**

Remember my passphrase on this computer  
 List my name in the groove.net public directory

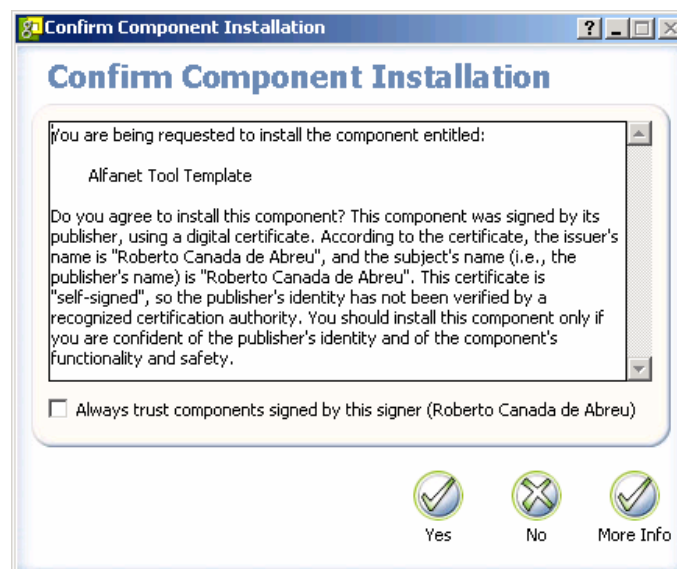
When you click Finish, your account and identity will be created.

- **Open a Space (click Start Here).**



- **Install the Alfabet Authoring Tool.**

Copy the ".grv" file (normally called "Alfanet.grv") file to disk and double click.  
Accept the Certificate



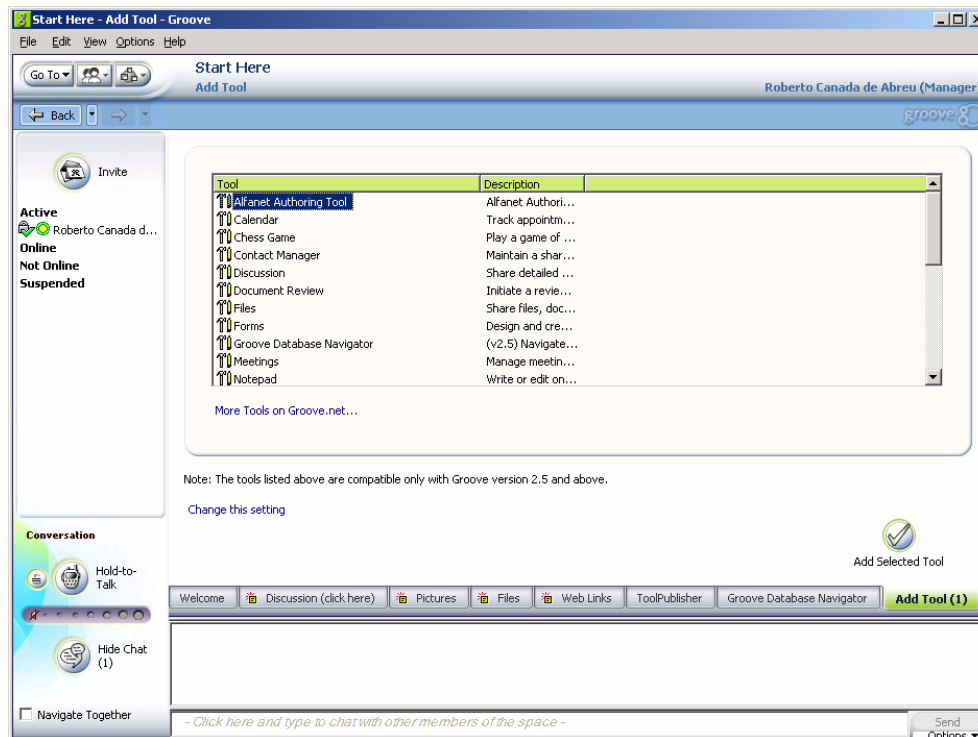
**Note:** The installation process will include some Groove components and could also, if needed, install the Microsoft .Net Framework.



- **Add the Alfabet Authoring Tool to the Space.**

Select the “Add Tool”

Select “Alfabet Authoring Tool” and double click



- **Start using the Alfabet Authoring Tool.**

Select the tab of the space with the Alfabet Authoring Tool and start use it



### 3. Tool Structure

The Alfabet Authoring Tool is divided into 5 modules with specific functions that will be explained in the following sections.

In the following picture we can see the main window of the Tool and also the main menu with the 5 different modules:

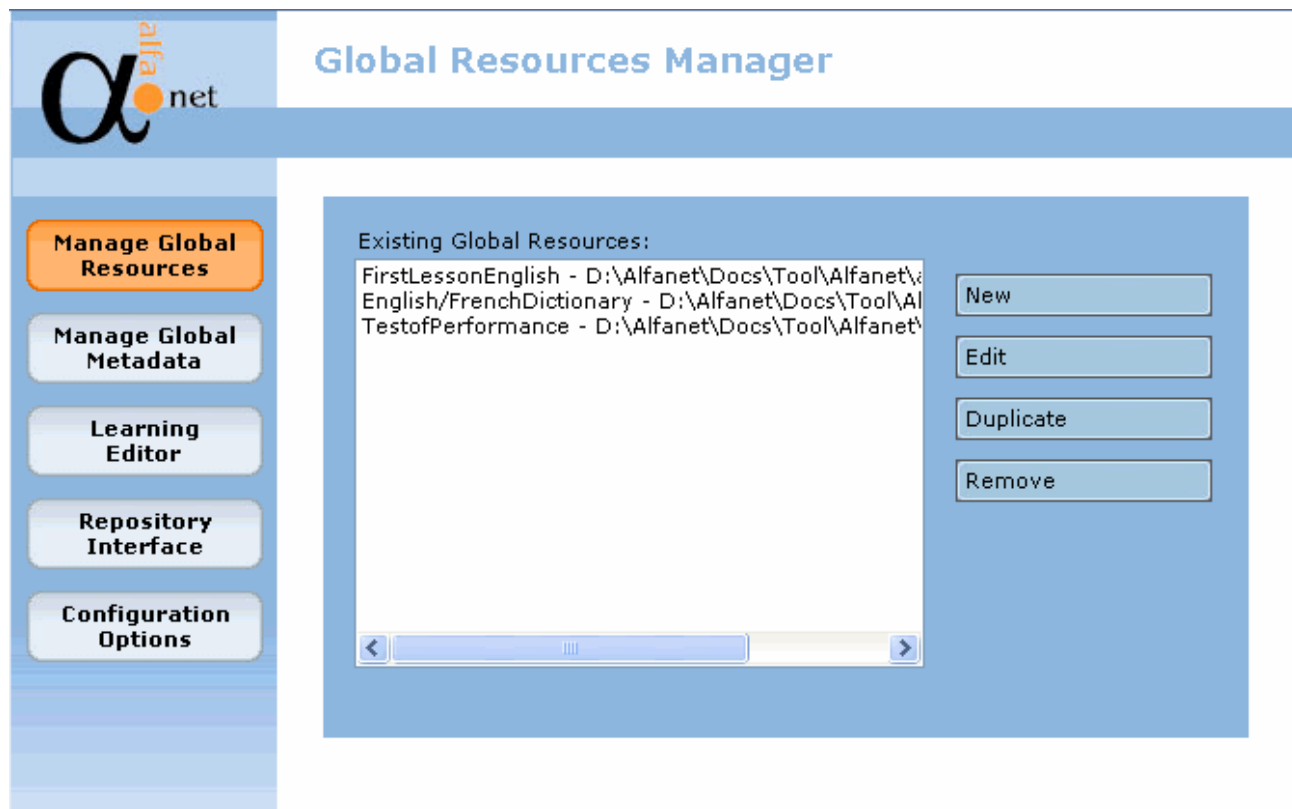


### 3.1 Manage Global Resources

Like it's said in the IMS Content Packaging section, Resources are files that represent the learning materials and objects that will be used in the course and are external to it.

These materials can be used and reused in several courses and, following that philosophy, we create a module where it's possible to the user to define resources once and then, when defining a course, will only need to load the global resource to the course.

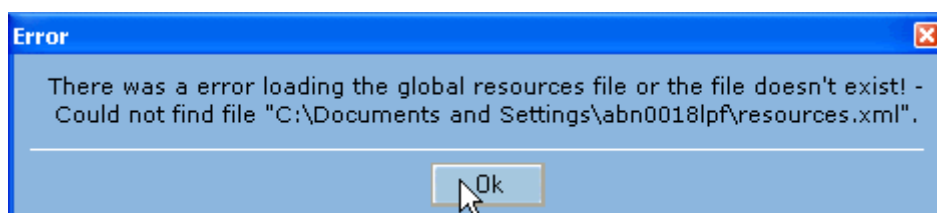
To accede to this module please choose, in the main menu, the button that says "Manage Global Resources".



This module will try to open a file of resources, already defined by the user with the tool, with the name "resources.xml" on the folder correspondent to the variable "Global Resources Directory" of the "Configuration Options" module. If this variable it's not defined the tool will assume a default directory, normally the user profile default directory or the home directory of Groove.

**Note:** The philosophy of this module is to define working folders where the user can centralize the resources and the correspondent files. This means that the user can create several kinds of resources and organize them in different folders and then load them to the courses. Despite the user could choose a file from any location of the file system, **we advise the user to have the physical files in the same folder of the Global Resources file.**

If there was any error loading the resources file, because it's not defined, doesn't exist or it's not valid, the tool will show you a message like the following picture and the list will be empty like in the above picture:



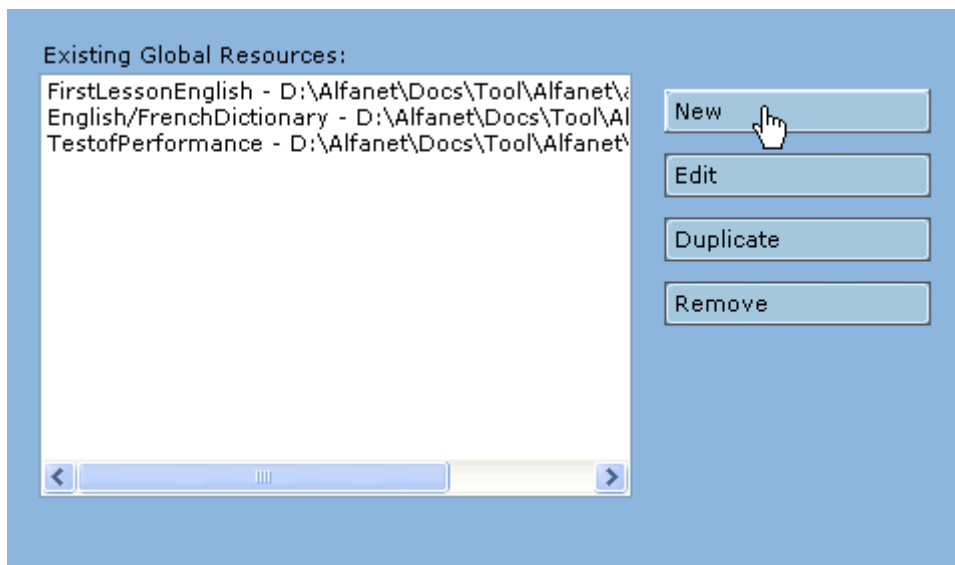
After loading the Global Resources file we can see a list of resources with the ID of the resource and the location of the physical file that the resource represents. On the right of that list we can find four buttons that allows us to manipulate those resources.

### 3.1.1 Options of Manage Global Resources module

#### 3.1.1.1 New

This option allows the user to create a new global Resource.

The user must click on the “New” button.



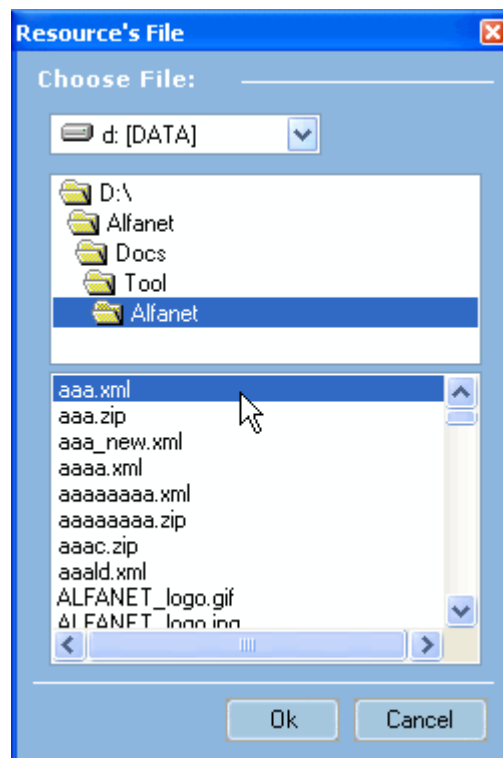
After this step will be presented to the user to fill the Resource data on a form like the picture below:

- **Title** - Represents the title that the user will give to the resource. This is a mandatory field, if this field is not filled the Resource can't be created;

**Note:** Despite this value is not the ID of the Resource (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with

some existing resource or other kind of element. **According to this, try to put an explicit value on the title of the resource to let you easily remember and identify the different resources;**

- **URI Reference** - Reference to an external file that can't be transform in a local file, e.g. a web site;
- **Type** – Type of resource. A resource can be “webcontent” or “imsldcontent”. The “webcontent” resource means that this is a file that a browser can show. The “imsldcontent” resource means that this is other kind of file. This is a mandatory field, if this field is not filled the Resource can't be created;
- **Metadata** – Button that allows the user to define the metadata associated with the Resource. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Local File** – Physical location of the file that the Resource represents;
- **Browse** – Button that allow the user to navigate through his file system in order to locate the target file. The form presented to the user is like the picture below. The user can navigate through the file system to choose the file that we want. By default, if the Local File is not filled, this window will open on the Global Resources folder. The user must choose a file from the list or cancel the operation, otherwise the user can't advance.

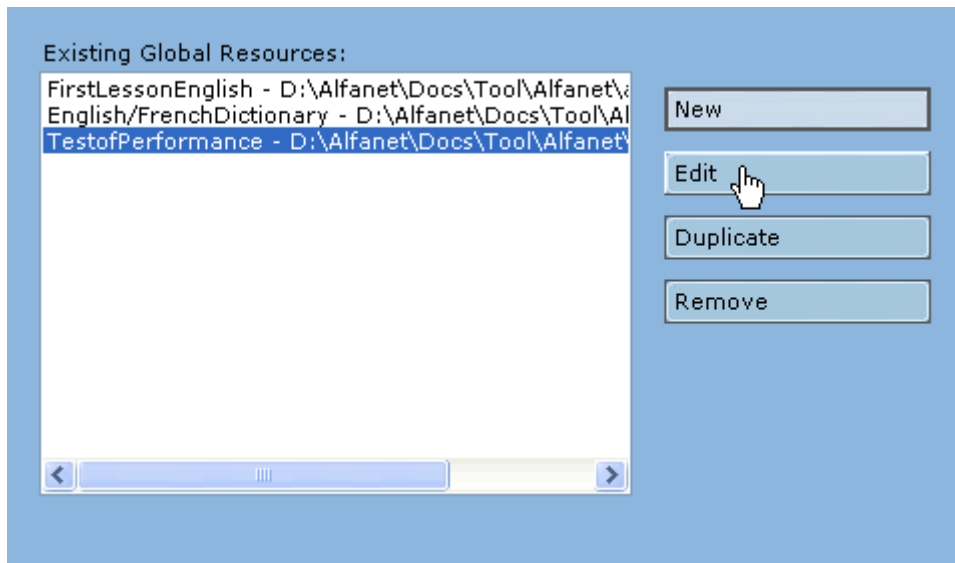


- **Open Chosen File** – Button that allow the user to see the chosen file with the default editor/reader application associated to that kind of file on Windows. If there is no application associated to the kind of the file this option will not work;
- **Resource Dependencies** – This is a list of the existing global resources. In this list the user can choose other resources that this new resource is depending of. Only the selected resources will be considered as dependencies to the new resource.

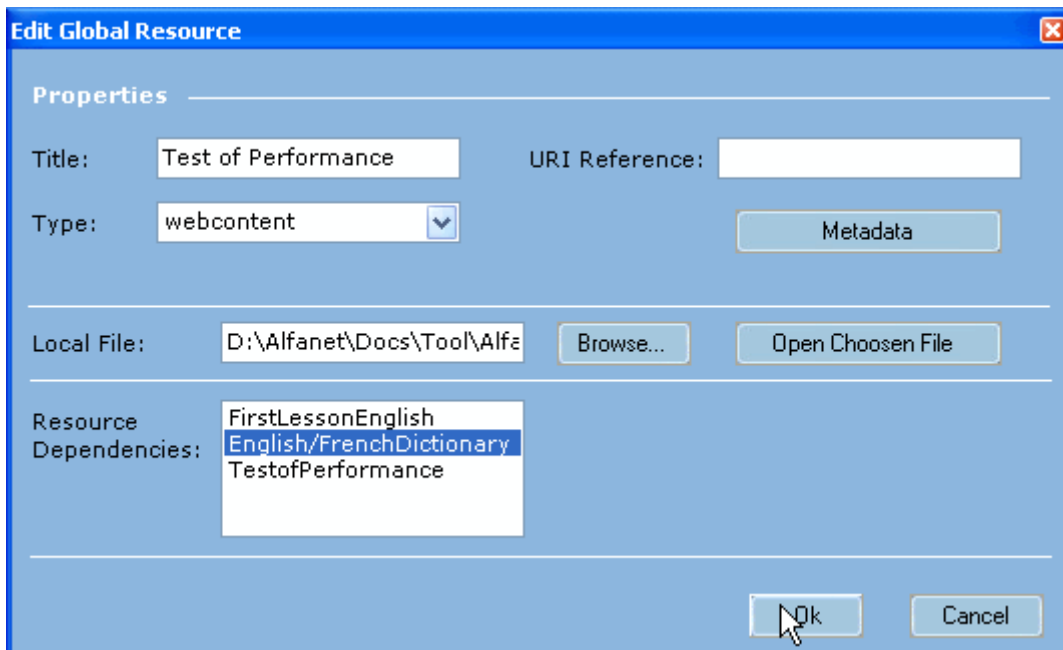
### 3.1.1.2 Edit

This option allows the user to see/edit/change an existing global Resource.

The user must choose from the list one existing Resource and click on the “Edit” button.



After this step will be presented to the user a form similar to the “New” option form but will appear filled with the Resource data.

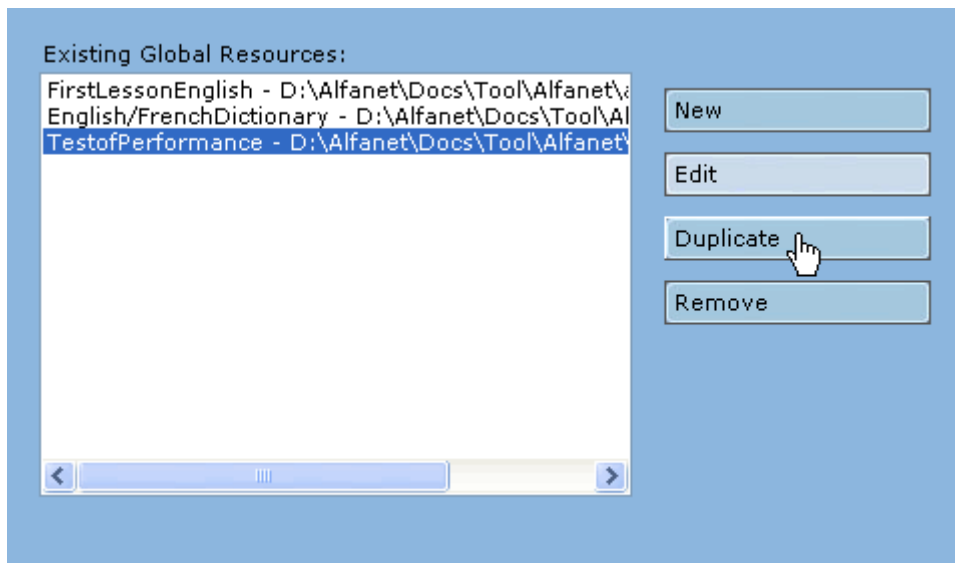


In this option the fields have the same meaning of the “New” option.

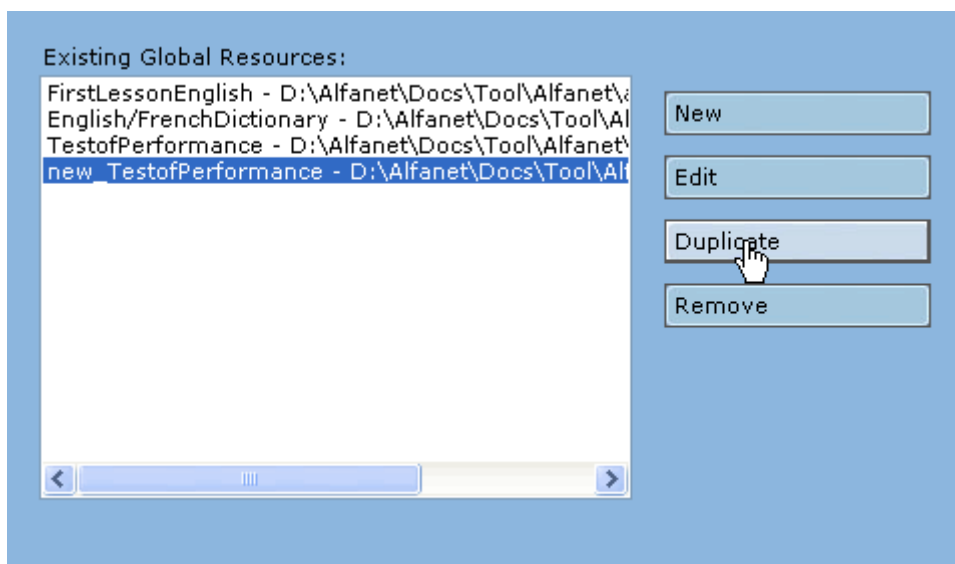
### 3.1.1.3 Duplicate

This option allows the user to create a new global resource similar to an existing one with only the exception of the Resource ID.

The user must choose from the list one existing resource and click on the “Duplicate” button.



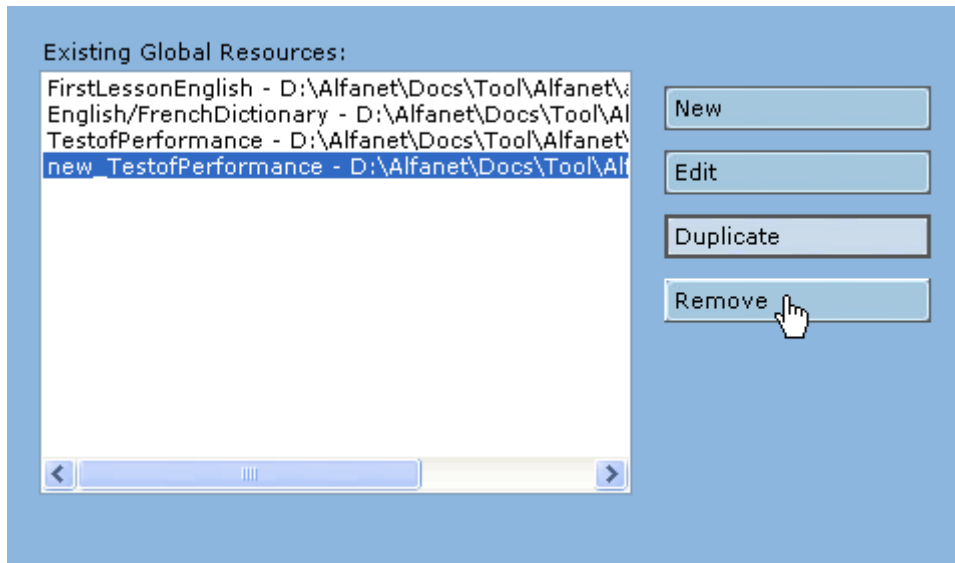
After this step, if there is no problem in the duplication of the resource, the list will be refreshed with the new Resource selected as you can see in the below picture.



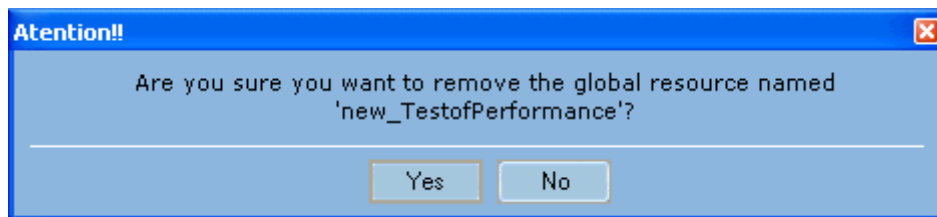
#### 3.1.1.4 Remove

This option allows the user to remove an existing global Resource.

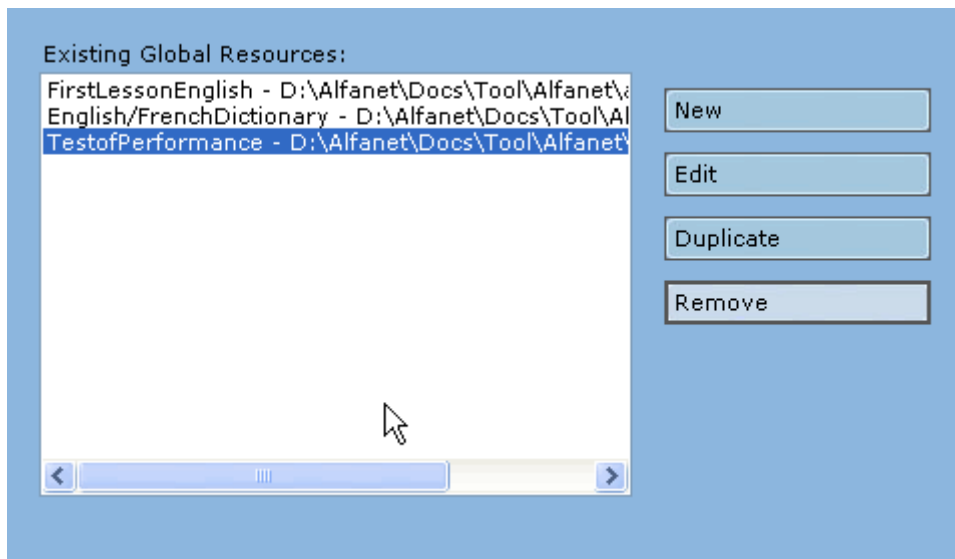
The user must choose from the list one existing Resource and click on the "Remove" button.



After this step the tool will present to the user a message like the below picture where he must confirm the decision of remove that Resource.



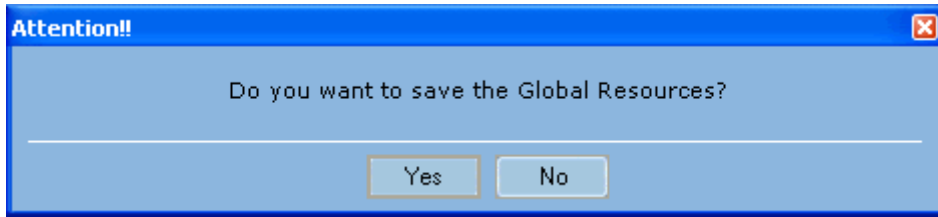
If the user confirms his decision the list will be refreshed without the deleted Resource and with the previous Resource of the list selected.



### 3.1.1.5 Save Global Resources

When the user is on the Manage Global Resources module and clicks on another button in the main menu the tool will present to him a message, like the picture below, if he wants to save the Global Resources.





If the user confirms the decision the Global Resources will be saved in a file named "resources.xml" on the folder chosen on the Configuration Options module.

## 3.2 Manage Global Metadata

Like it's said in the *IMS-LOM* section, the Authoring Tool uses the *IMS-LOM* standard to define the metadata associated to any kind of elements in the course.

The Manage Global Metadata module allows the user to define a template metadata to use in every new course element. So, if the user wants to, the information defined in this module *will be copied to the new elements when the user clicks on the Metadata related button(s) of the element.*

To accede to this module please choose, in the main menu, the button that says "Manage Global Resources".

As you can see in the picture above, this module doesn't have specific manipulations options. This module consists in the edition of the global metadata information.

This information is present in the file defined on the field "Global Metadata File" of the Configuration Options module. If this value is not defined or is not a valid file, the global metadata form will be presented unfilled otherwise will be filled with the metadata information.

**Note:** You might have two kinds of approach to this module.

*The first is to use it as a template for every objects of a course that you can adapt or complete later when you are editing a specific one.*

*The other approach is to see this module as a complete metadata template that you don't want to change and will be equal for every objects of the course.*

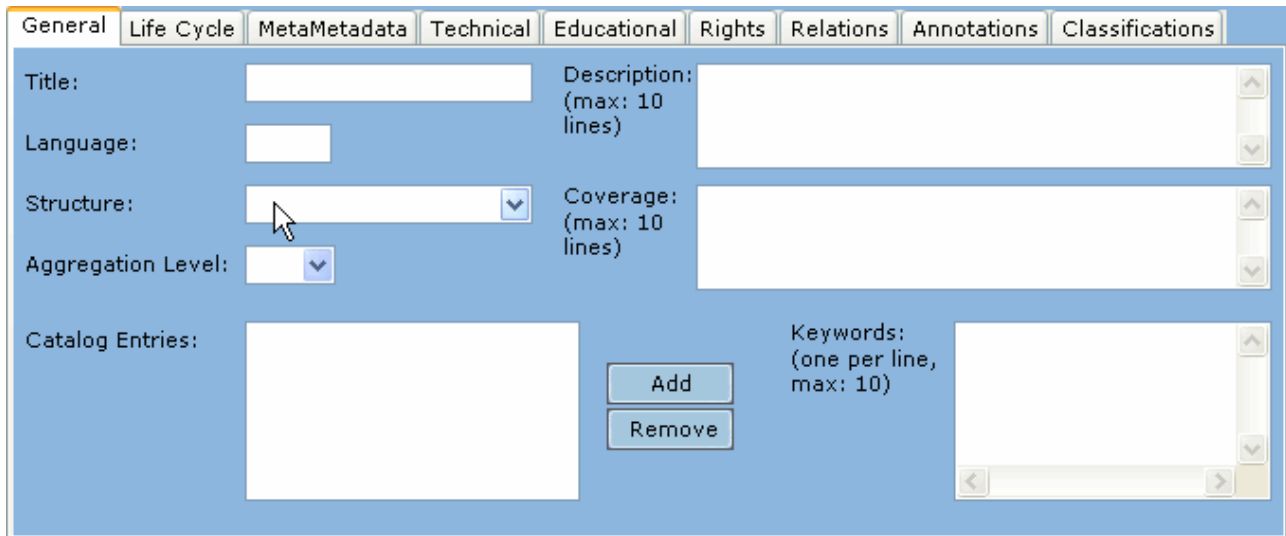
*You can also don't use this module and define the metadata of each object one by one, if the metadata information of the objects of the course aren't similar.*

### 3.2.1 Metadata Information Categories

The metadata information is divided in nine categories, each one with a specific context and objective.

**Note:** *Despite this information is on the Manage Global Metadata module this will be referenced by every Metadata sections/buttons of the elements of the Learning Editor module because the information and interface are similar.*

### 3.2.1.1 General



In the General category we can find information with an open scope like:

- **Title** – Text value that represents the title that the user will give to the metadata information or object;
- **Language** – Acronym value that represents the language that the object is created.  
The acronym is in the form of two letters defining the language, e.g.: Portuguese → “pt”; English → “en”; French → “fr”.  
Optionally we can add a reference to a specific country. This reference has also two letters, e.g.: English (GB) → “en-GB”; English (US) → “en-US”.  
If the acronym is not in the right format the metadata can not be saved;
- **Structure** – Type of organization of the object. This field has fixed values that are: “*Collection*”, “*Mixed*”, “*Linear*”, “*Hierarchical*”, “*Networked*”, “*Branched*”, “*Parceled*”, “*and Atomic*”. Please choose the closest description that applies to the object;
- **Aggregation Level** – This value refers to the functional size of the object.  
Level 1 means smallest level of aggregation, e.g. raw media data or fragments.  
Level 2 refers to a collection of atoms, e.g. an HTML document with some embedded pictures or a lesson.  
Level 3 indicates a collection of level 1 resources, e.g. a 'web' of HTML documents, with an index page that links the pages together or a unit.  
Finally, level 4 refers to the largest level of granularity, e.g. a course;
- **Description** – Text value (max: 10 lines) to briefly describe the object content;
- **Coverage** – Text value (max: 10 lines) to briefly describe the object historical context of the content;
- **Catalog Entries** – List of entries that define the object as a part of some catalog.  
This feature allows the user to **Add** or **Remove** entries.  
When the user click on the “Add” button the tool presents a form like the below picture where the user must fill the catalog name and the value of the entry;

- **Keywords** – List of words (max: 10) that can help to classify the object. Each line must have only one word;

### 3.2.1.2 Life Cycle

In the Life Cycle category we can find information about the evolution of the content in a basis of versions and status:

- **Version** – Text value that represents the actual version of the content of the object;
- **Status** – Status of the content of the object. This field has fixed values that are: *“Draft”, “Final”, “Revised”, and “Unavailable”*. Please choose the closest description that applies to the object;
- **Contributes** – List of contributions of entities to the actual and previous versions of the content. This features allows the user to create **New** contributes, **Edit** existing contributes, **Duplicate** contributes or **Remove** contributes.

When the user choose to create a New contribute or Edit an existing one, he has to fill a form that the tool presents to him like the below picture.

The difference between the two options is that, in the creation the form will appear empty, but in the edition the form will appear with the contribute data.

In the field “Kind of Contribution” the user must choose the role of the entities of this contribute between the following values: “Author”, “Publisher”, “Unknown”, “Initiator”, “Terminator”, “Validator”, “Editor”, “Graphical Designer”, “Technical Implementer”, “Content Provider”, “Technical Validator”, “Educational Validator”, “Script Writer”, “Instructional Designer”.

Please choose the closest description that applies to the entities of this contribute because *this is a mandatory field*.

Besides the role user must introduce the “Date” of the contribution (by default if this value is not defined the tool will provide the current date) and the list of “Entity(ies) involved”.

The list of entities consists in the description (one entity per line and max: 40) of the entities that corresponds to the role chosen in the “Kind of Contribution” field.

### 3.2.1.3 Metametadata

In the Metametadata category we can find information about the information on this metadata section. For other words it’s the data about the data about the object content:

- **Metadata Scheme** – Text value that represents the metadata schema used on this block/file. By default the value is LOMv1.0;
- **Language** – Acronym value that represents the language that the metadata is created.

The acronym is in the form of two letters defining the language, e.g.: Portuguese → “pt”; English → “en”; French → “fr”.

Optionally we can add a reference to a specific country. This reference has also two letters, e.g.: English (GB) → “en-GB”; English (US) → “en-US”.

If the acronym is not in the right format the metadata can not be saved;

- **Catalog Entries** – See “Catalog Entries” on General category. In this context the object is the metadata itself;
- **Contributes** – See “Contributes” on Life Cycle category. In this context the content is the metadata itself;

### 3.2.1.4 Technical

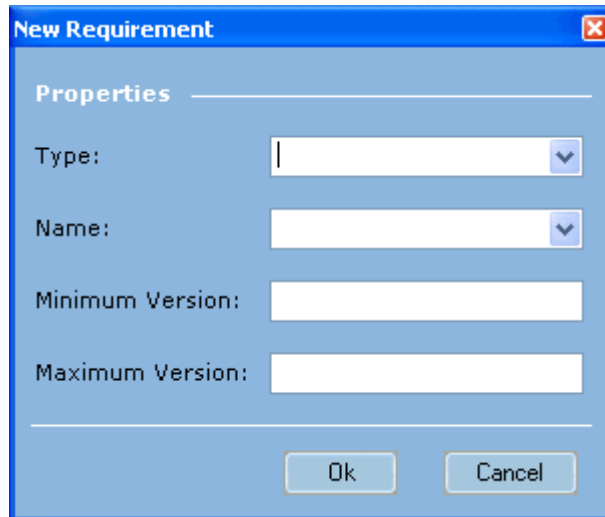
In the Technical category we can find information about the file that the object refers like format, size...

- **Format** – Text value that represents the format of the associated file in MIME Type.  
There is already a list of values that the user can choose, but the user can introduce an additional type because the MIME Types list is not static;
- **Duration** – Three values that represents the time that the associated file will take to be showed or played. The field “H” represents hours, the field “M” represents minutes and the field “S” represents seconds. All three values must be numeric;
- **Size** – Numeric value that represents the physical size of the associated file in Bytes;
- **Location(s)** – List of physical locations or methods (one per line, max: 10) that resolves the location of the associated file. The first location it’s the preferable location;
- **Installation Remarks** – Text value to briefly describe how to install the associated file/application;
- **Other Platform Requirements** – Text value about software required to the object other than Operating System or Browser;
- **Requirements** – List of Operating System or Browser requirements needed to the object.

This feature allow to the user create **New** requirements, **Edit** existing requirements, **Duplicate** requirements or **Remove** requirements.

When the user choose to create a New contribute or edit an existing one he has to fill a form that the tool presents to him like the below picture.

The difference between the two options is that, in the creation the form will appear empty, but in the edition the form will appear with the requirement data.



In the “Type” field the user can choose if the requirement is about an “*Operating Browser*” or about an internet “*Browser*”. This is a *mandatory field*.

If the user chooses the “*Operating System*” value in the “Name” field he can choose between the following values: “*PC-DOS*”, “*MS-Windows*”, “*MacOS*”, “*Unix*”, “*Multi-OS*”, “*Other*” and “*None*”.

Otherwise, if the user chooses the “*Browser*” value in the “Name” field he can choose between the following values: “*Any*”, “*Netscape Communicator*”, “*Microsoft Internet Explorer*”, and “*Opera*”.

The Name field is a *mandatory field*, so the user have to fill it, otherwise the user can’ advance.

In the “Minimum Version” and “Maximum Version” the user must introduce between which versions of the selected operating system or browser this requirements refers to.

**Note:** *This category is very specific to objects that have some kind of digital support associated (files, applications, plug-ins, etc...) so its recommended to fill this category differently for each object and not as a template in the global metadata.*

### 3.2.1.5 Educational

In the Educational category we can find information about the pedagogical issues of the object:

- Interactivity Type** – Type of interaction between the object and the final user.  
This field has fixed values that are: *“Active”*, *“Expositive”*, *“Mixed”* and *“Undefined”*.  
In an *Expositive* object, the information flows mainly from the resource to the learner. Expositive documents are typically used for learning-by-reading.  
In an *Active* object, information also flows from the learner to the resource. *Active* documents are typically used for learning-by-doing.  
Note: Activating links to navigate in hypertext documents is not considered as an information flow. Thus, hypertext documents are expositive.  
Please choose the closest description that applies to the object;
- Description how to use** – Text value to briefly describe how to pedagogical use the object;
- Interactivity Level** – Level of interaction between the object and the end user.  
This field has fixed values that are: *“Very Low”*, *“Low”*, *“Medium”*, *“High”* and *“Very High”*.  
Please choose the closest description that applies to the object;
- Language** – Acronym value that represents the language that the pedagogical object is created.  
The acronym is in the form of two letters defining the language, e.g.: Portuguese → “pt”; English → “en”; French → “fr”.  
Optionally we can add a reference to a specific country. This reference has also two letters, e.g.: English (GB) → “en-GB”; English (US) → “en-US”.  
If the acronym is not in the right format the metadata can not be saved;
- Typical Age Range** – Text value that represents the target end user age ranges that the object is recommended to be applied;
- Learning Resource Type(s)** – Type of the object in a pedagogical view. The first type is the predominant one.  
This feature allows the user to **Add** and **Remove** learning resource types.  
When the user selects to Add a new object the tool presents to him a form with a list of values to choose.  
This list has the following values: *“Exercise”*, *“Simulation”*, *“Questionnaire”*, *“Diagram”*, *“Figure”*, *“Graph”*, *“Index”*, *“Slide”*, *“Table”*, *“Narrative Text”*, *“Exam”*, *“Experiment”*, *“ProblemStatement”*, and *“SelfAssessment”*.



Please choose the closest description that applies to the object;

- **Difficulty** – Level of pedagogical difficulty of the object for the typical target audience. This field has fixed values that are: “*Very Easy*”, “*Easy*”, “*Medium*”, “*Difficult*”, and “*Very Difficult*”. Please choose the closest description that applies to the object;
- **Semantic Density** – Subjective measure of the learning object's usefulness as compared to its size or duration. This field has fixed values that are: “*Very Low*”, “*Low*”, “*Medium*”, “*High*” and “*Very High*”. Please choose the closest description that applies to the object;
- **End User Role(s)** – Normal kind of end user that uses this object.  
This feature allows the user to **Add** and **Remove** roles.  
When the user selects to Add a new object the tool presents to him a form with a list of values to choose.  
This list has the following values: “*Very Low*”, “*Low*”, “*Medium*”, “*High*” and “*Very High*”.  
Please choose the closest description that applies to the object;
- **Typical Learning Time** – Three values that represent the approximate or typical time it takes to work with the object.  
The field “H” represents hours, the field “M” represents minutes and the field “S” represents seconds. All three values must be numeric;
- **Context(s)** – The typical learning environment where use of the object is intended to take place.  
This feature allows the user to **Add** and **Remove** contexts.  
When the user selects to Add a new object the tool presents to him a form with a list of values to choose.  
This list has the following values: “*Primary Education*”, “*Secondary Education*”, “*Higher Education*”, “*University First Cycle*”, “*University Second Cycle*”, “*University Postgrade*”, “*Technical School First Cycle*”, “*Technical School Second Cycle*”, “*Professional Formation*”, “*Continuous Formation*”, “*Vocational Training*”.  
Please choose the closest description that applies to the object;

**Note:** This category is very specific to objects with a pedagogical objective so it's recommended to fill this category differently for each object and not as a template in the global metadata.

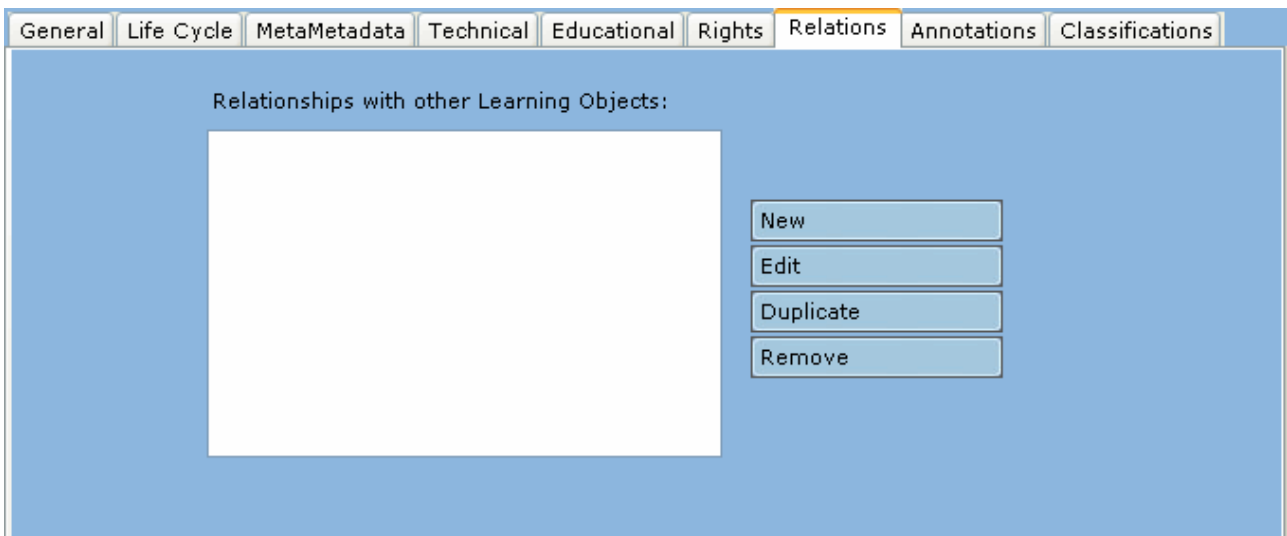
### 3.2.1.6 Rights

The screenshot shows a software interface with a tabbed menu at the top. The tabs are: General, Life Cycle, MetaMetadata, Technical, Educational, Rights (highlighted), Relations, Annotations, and Classifications. The main content area is blue and contains a 'Description:' label followed by a large empty text box with a vertical scrollbar. Below the text box is a section titled 'Type Of Use Cost:' with two radio button options: 'Free Use' (which is selected) and 'Use is Not Free'. At the bottom of the interface, there is a checkbox labeled 'CopyRight and Other Restrictions must be Apply'.

In the Rights category we can find information about the conditions of use of the object content:

- **Description** – Text value that briefly describes the conditions of use of the object;
- **Type of Use Cost** – Field to the user define if the use of the object is free (“Free Use”) or has any cost (“Use is Not Free”);
- **Copyright and Other Restrictions must be Apply** – Field to the user define if the object has copyrights or other kind of use restriction;

### 3.2.1.7 Relations



In the Relations category we can find information about the relationships between this object and other objects.

This feature allows the user to:

- Create **New** relationships;
- **Edit** existing relationships;
- **Duplicate** existing relationships;
- **Remove** existing relationships.

When the user tries to create a New relationship or Edit an existing one, the tool will present to him a form like the below picture. The difference between the two options is that in the creation the form will appear empty but in the edition the form will appear with the relationship data.

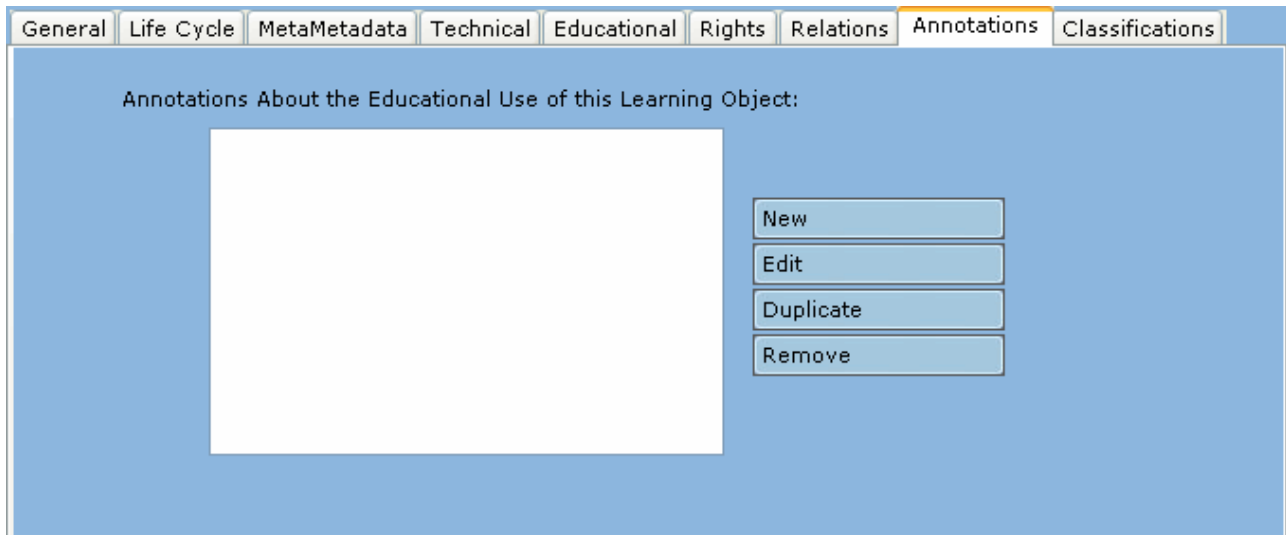
The image shows a 'New Relation' dialog box with the following fields and controls:

- Properties:** A dropdown menu for 'Kind of Relationship'.
- Related Resource:** A dropdown menu for 'Resource'.
- Description of the Related Resource:** A text input field for 'Description'.
- Catalog Entries:** A list box with 'Add' and 'Remove' buttons next to it.
- Buttons:** 'Ok' and 'Cancel' buttons at the bottom.

- **Kind of Description** – Field where the user chooses which kind of relationship that we wants to create.  
 This field has fixed values that are: “*IsPartOf*”, “*HasPart*”, “*IsVersionOf*”, “*HasVersion*”, “*IsFormatOf*”, “*HasFormat*”, “*References*”, “*IsReferencedBy*”, “*IsBasedOn*”, “*IsBasisFor*”, “*Requires*”, “*IsRequiredBy*”.  
 Please choose the closest description that applies to the object because this is a *mandatory field*;
- **Resource** – List of IDs of resources that are available in the context. The user must choose one Resource on which the object has the relation.  
 If you filling this value in the Global Metadata probably there will not appear any resource and the list will be empty. If this situation happens it’s impossible, in practice, to complete the Relationship because this is a *mandatory field*;
- **Description** – Text field that briefly describes the Resource on which the object has this relationship;
- **Catalog Entries** – See “Catalog Entries” on General category. In this context the object is the Related Resource;

**Note:** Since this category is very dependent of the working context and deals with very specific information of each object, it’s recommended to fill this category differently for each object and not as a template in the global metadata.

### 3.2.1.8 Annotations



In the Annotations category we can find comments about the use of the object, educational or not.

This feature allows the user to:

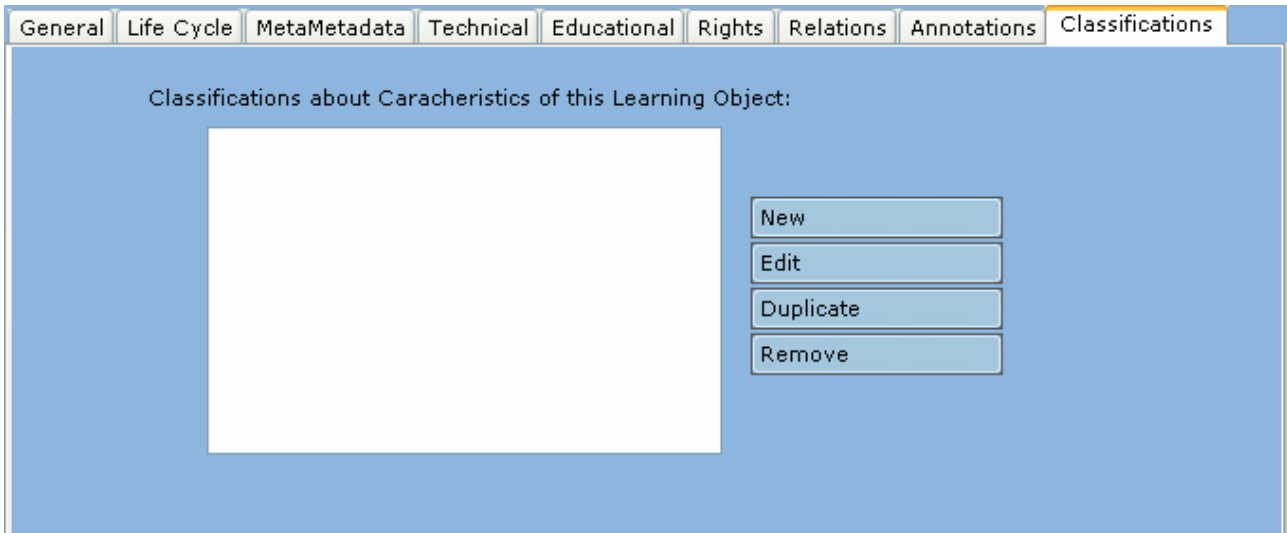
- Create **New** annotations;
- **Edit** existing annotations;
- **Duplicate** existing annotations;
- **Remove** existing annotations.

When the user tries to create a New annotation or Edit an existing one, the tool will present to him a form like the below picture. The difference between the two options is that, in the creation the form will appear empty, but in the edition the form will appear with the annotation data.

- **Author First Name, Author Middle Name and Author Last Name** – Three text values that the user must fill to give the name of the annotation author. The first and the last names are *mandatory fields*;
- **Author E-Mail** – Text value to the user give the author's e-mail address as a contact information about the annotation content. This field must be filled because is a *mandatory field*;

- **Date** – Three numeric fields that allows the user to give the date of the comment text.  
By default and if this value is not filled the tool will automatically put the current date. The format of the date is Year – Month – Day.  
If the date is not filled nor have non-numeric fields there will be impossible to advance also because it's a mandatory field.
- **Annotation Text** – Text value where the user can fill the comment itself. This field must be filled because it's a mandatory field.

### 3.2.1.9 Classifications



In the Classifications category we can find information about classify any specifics characteristics of the object.

This feature allows the user to:

- Create **New** classifications;
- **Edit** existing classifications;
- **Duplicate** existing classifications;
- **Remove** existing classifications.

When the user tries to create a New classification or Edit an existing one, the tool will present to him a form like the below picture. The difference between the two options is that, in the creation the form will appear empty, but in the edition the form will appear with the classification data.

- **Characteristic** – Field where the user chooses which characteristic of the object we wants to classify.  
This field has fixed values that are: “*Discipline*”, “*Idea*”, “*Prerequisite*”, “*Educational Objective*”, “*Accessibility Restrictions*”, “*Educational Level*”, “*Skill Level*”, “*Security Level*”.  
Please choose the closest description that applies to the object because this is a *mandatory field*;
- **Description** – Text value to the user briefly describe the object according to the characteristic chosen. This field must be filled because is a *mandatory field*;
- **Keywords** – List of words that can help to describe the object according to the characteristic chosen. Each line must have only one word and with a maximum of 40 words;
- **Taxon Path** – Set of three values that represent a taxonomic path in a specific classification. It’s formed by **Source**, **Taxon Id** and **Taxon Entry**.

**Note:** *It’s recommended this part to be filled only by advanced users that are familiarized with the concepts involved.*

- **Source** – Text value where the user refers which specific classification is involved in this part;
- **Taxon Id** – Text value where the user refers the Taxon’s identifier in taxonomic system;
- **Taxon Entry** – Text value where the user refers the value of the taxonomic classification;

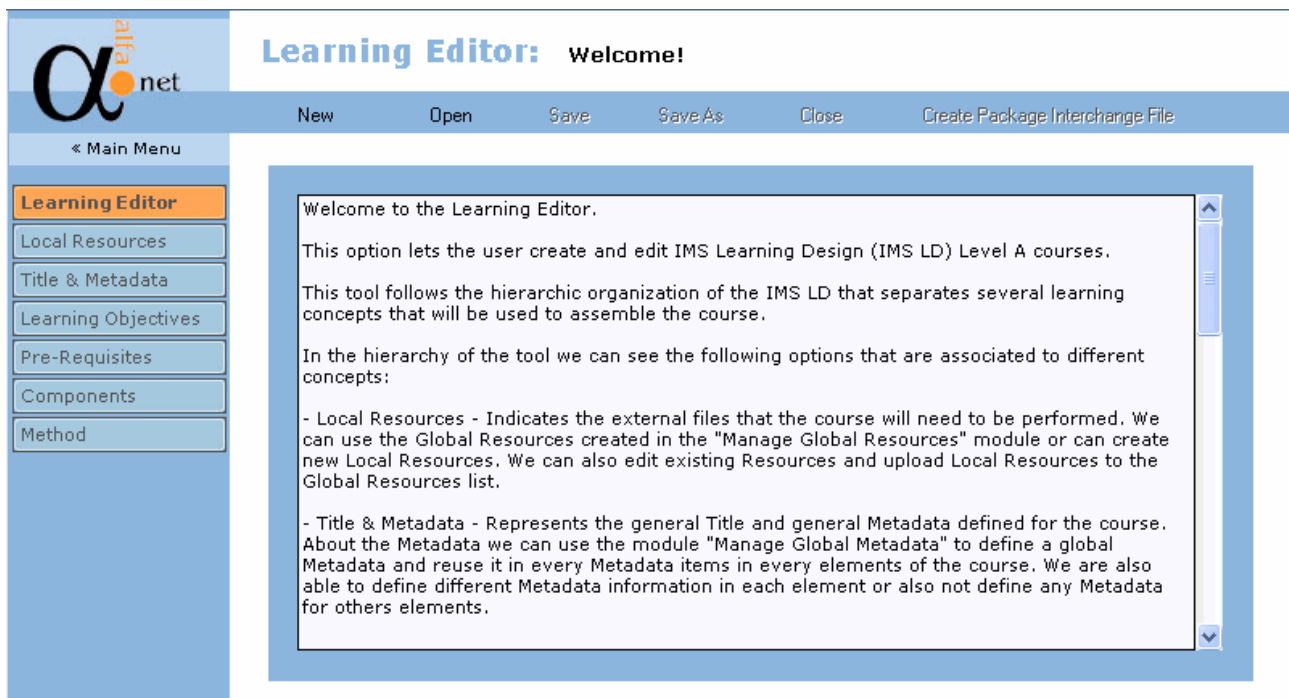
### 3.3 Learning Editor

This is the core module of the Authoring Tool. The Learning Editor module allows the user to create and edit courses in a very user-friendly way.

To accede to this module please choose, in the main menu, the button that says "Learning Editor".



After this step the tool will be presented to the user the like below picture.



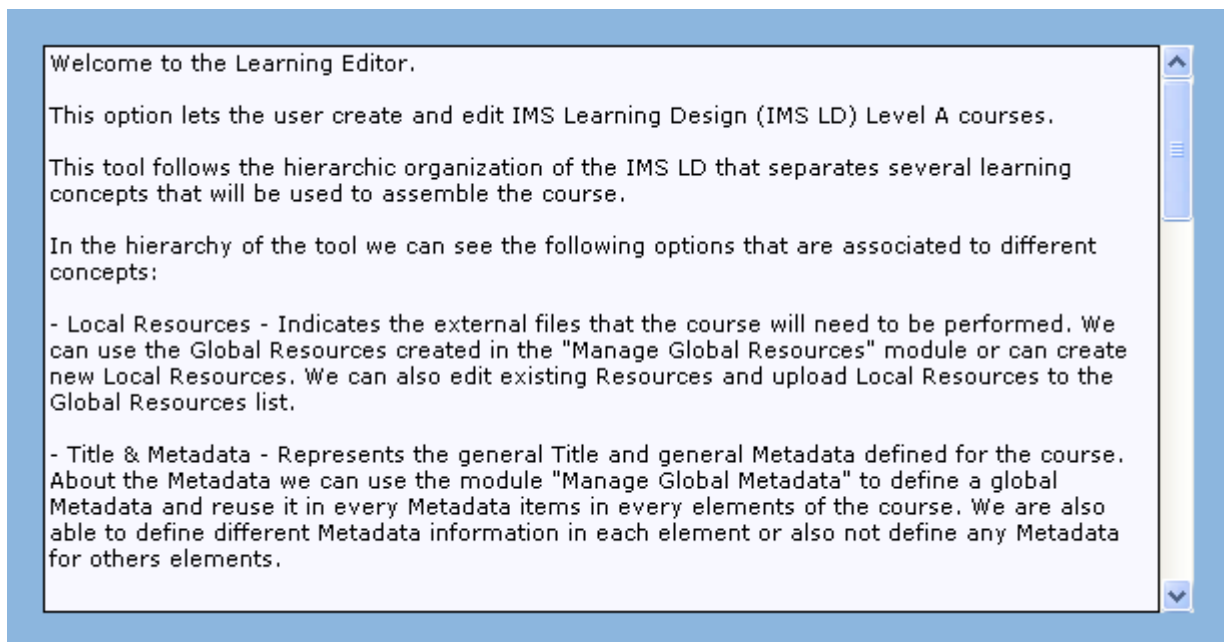
As you can see the main menu (at left) disappears to be replaced by the Learning Editor menu. If the user want to leave the Learning Editor module he needs to click on the "<< Main Menu" link to get back to the main menu, like showed on the below picture.



This Learning Editor menu that we can see on the above picture represents the hierarchy of the Learning Editor, which is similar to the hierarchy of Learning Design.

This will be explained later on this manual on the “Learning Editor Sub Modules” section.

The Learning Editor module has a brief text on the entry, like the below picture, to help the user to understand the philosophy of the tool and also of the Learning Design.



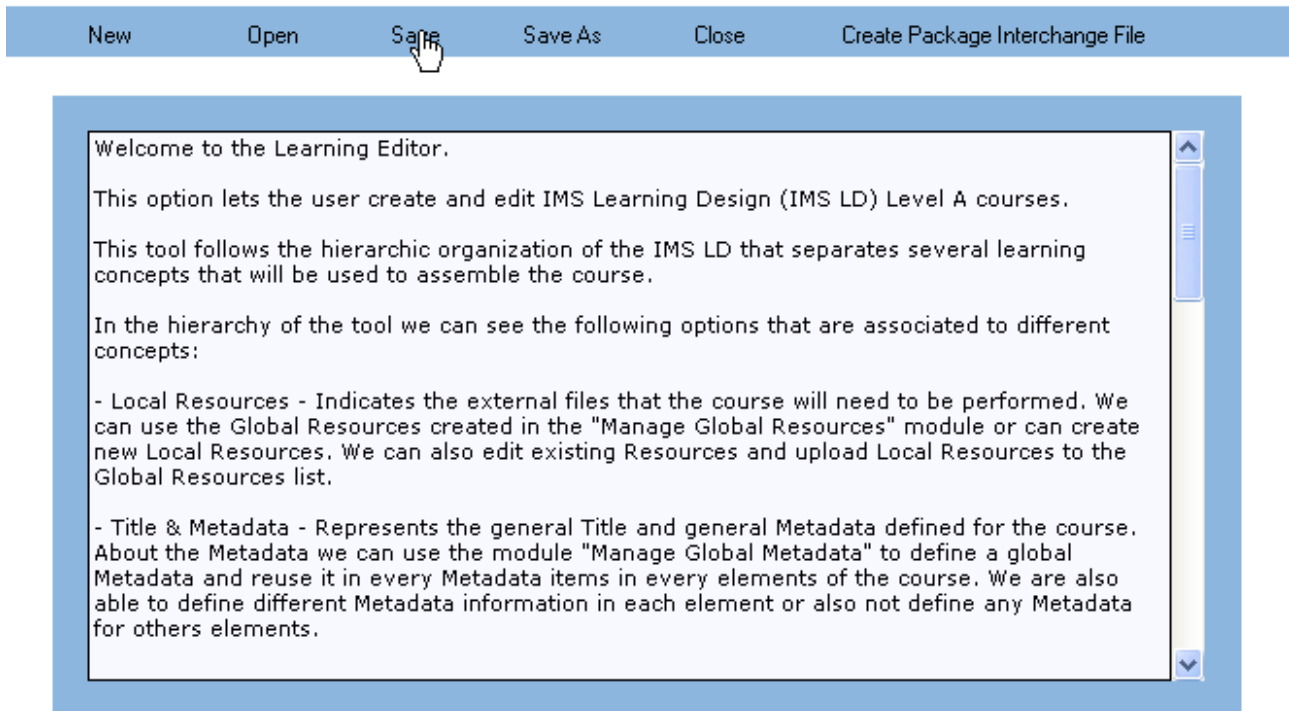
This kind of approach to help the user is also used on the “Components Manager” and “Method Manager” which are the sub-modules that agglomerate concepts that need to have some explanation even beyond the user manual.

The Learning Editor also has some file manipulations options such as a special feature named “Create Package Interchange File”.

This features we can find them on the main window of the Learning Editor like the below picture.



## Learning Editor: Welcome!

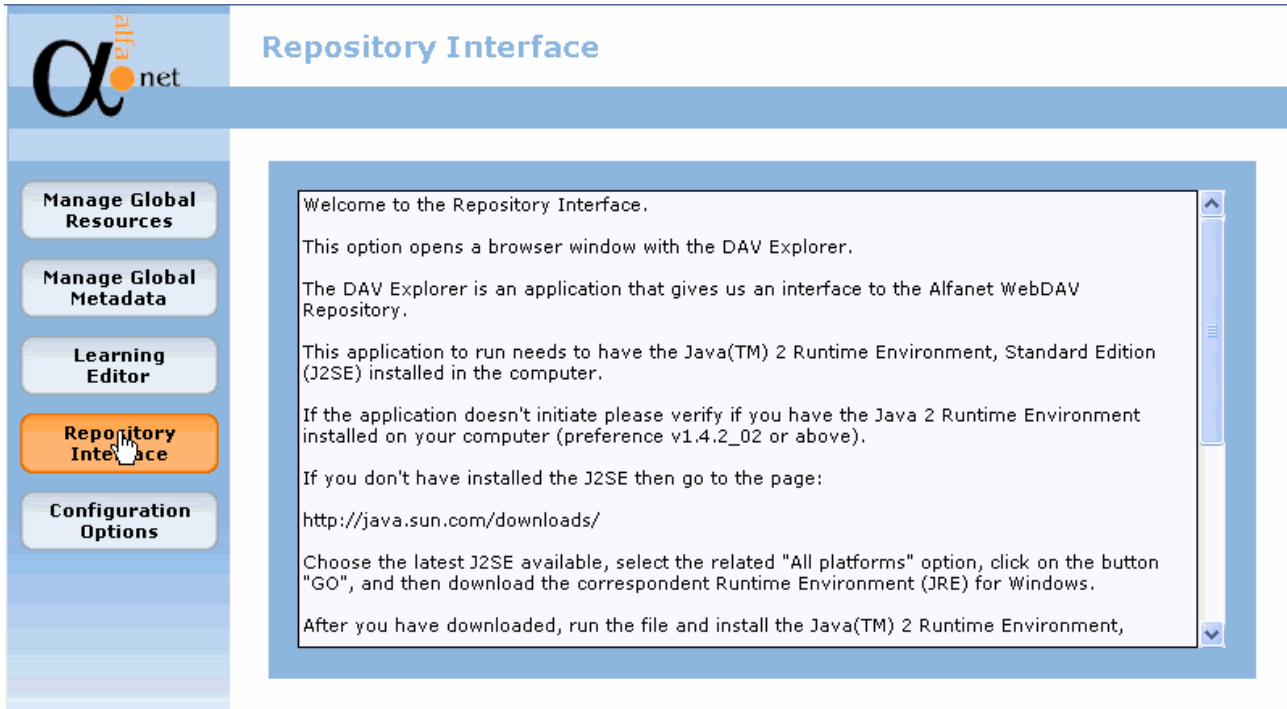


To more information about how to work with and how it's build the Learning Editor please read the section "Learning Editor Sub Modules".

### 3.4 Repository Interface

The Repository Interface module allows the user to have an interface with the Alfabet Repository system to transfer courses edited in the Authoring Tool and every other features that the WebDav protocol allows.

To accede to this module please choose, in the main menu, the button that says "Repository Interface".



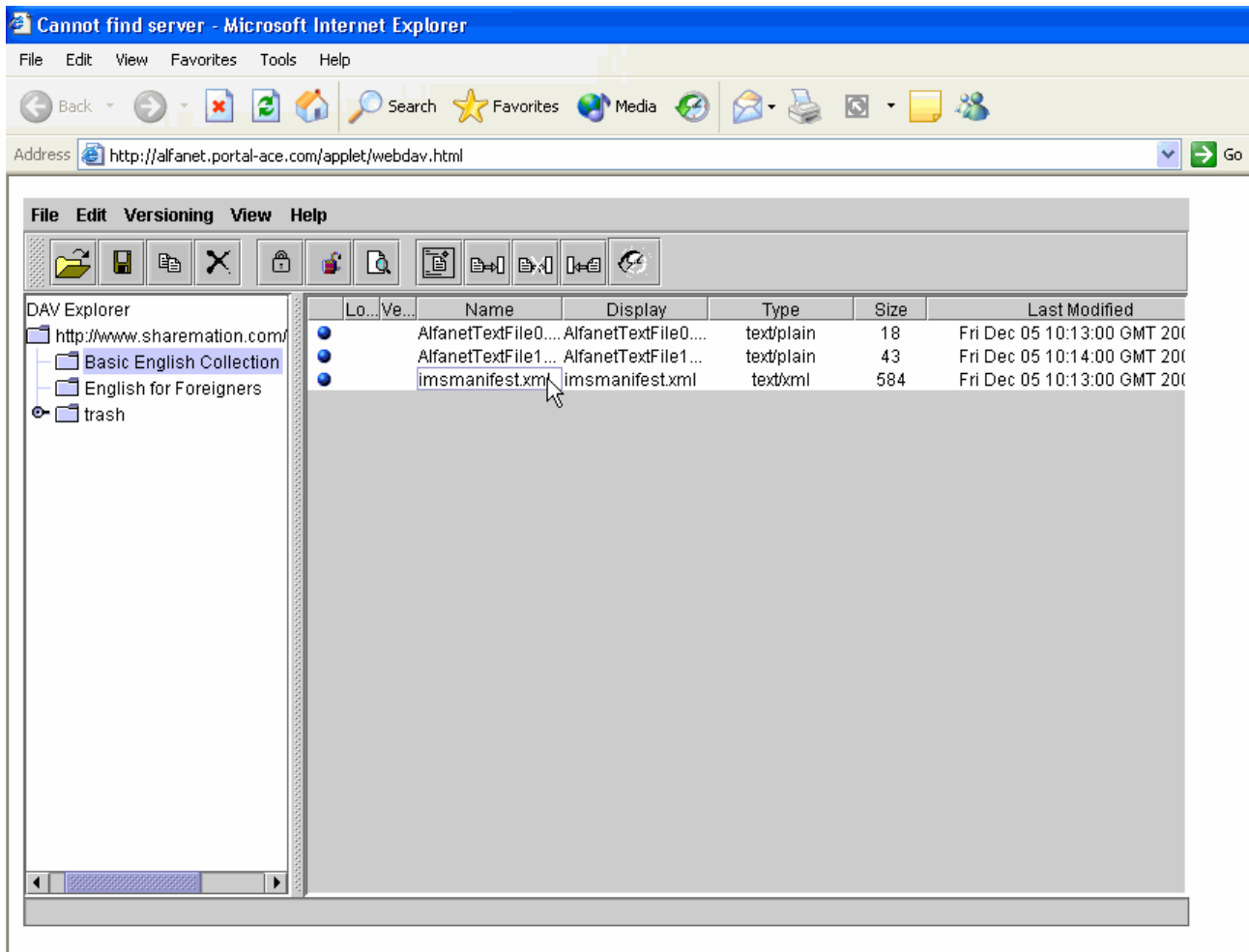
As referred before, the communication between the tool and the Alfabet repository is made by the WebDav protocol.

To do so the Authoring Tool use an application named "DAV Explorer".

This application supports almost every features of the WebDav protocol and it's available in the Authoring Tool on the form of an applet located in a web site.

The module launches an internet browser window with that web site, which contains the DAV Explorer applet.

The browser window will be like the following picture.



For more information how to use the DAV Explorer applet please consult the user manual of the DAV Explorer in the following web site:

<http://www.ics.uci.edu/~webdav/um.html>

### 3.4.1 How to Configure the DAV Explorer

This application to run needs to have the Java(TM) 2 Runtime Environment, Standard Edition (J2SE) installed in the computer.

If the application doesn't initiate please verify if you have the Java 2 Runtime Environment installed on your computer (preference v1.4.2\_02 or above).

If you don't have installed the J2SE then go to the page:

<http://java.sun.com/downloads/>

Choose the latest J2SE available, select the related "All platforms" option, click on the button "GO", and then download the correspondent Runtime Environment (JRE) for Windows.

After you have downloaded, run the file and install the Java(TM) 2 Runtime Environment, Standard Edition.

If after these steps the application still doesn't work or doesn't connect to Alfanet Repository please verify if you are using an HTTP Proxy Server to connect to the Internet.

If you are using the Internet Explorer go to "Tools" menu, choose "Internet Options", "Connections" & "LAN Settings", then verify if there is defined any Proxy Server.

If you have a Proxy Server go to the WebDav Explorer window, go to "Edit" menu, select "Edit Proxy Info" option and then write the Proxy Server Name or IP Address followed by ":" and then the Proxy Port (e.g.: "127.134.97.03:8080").

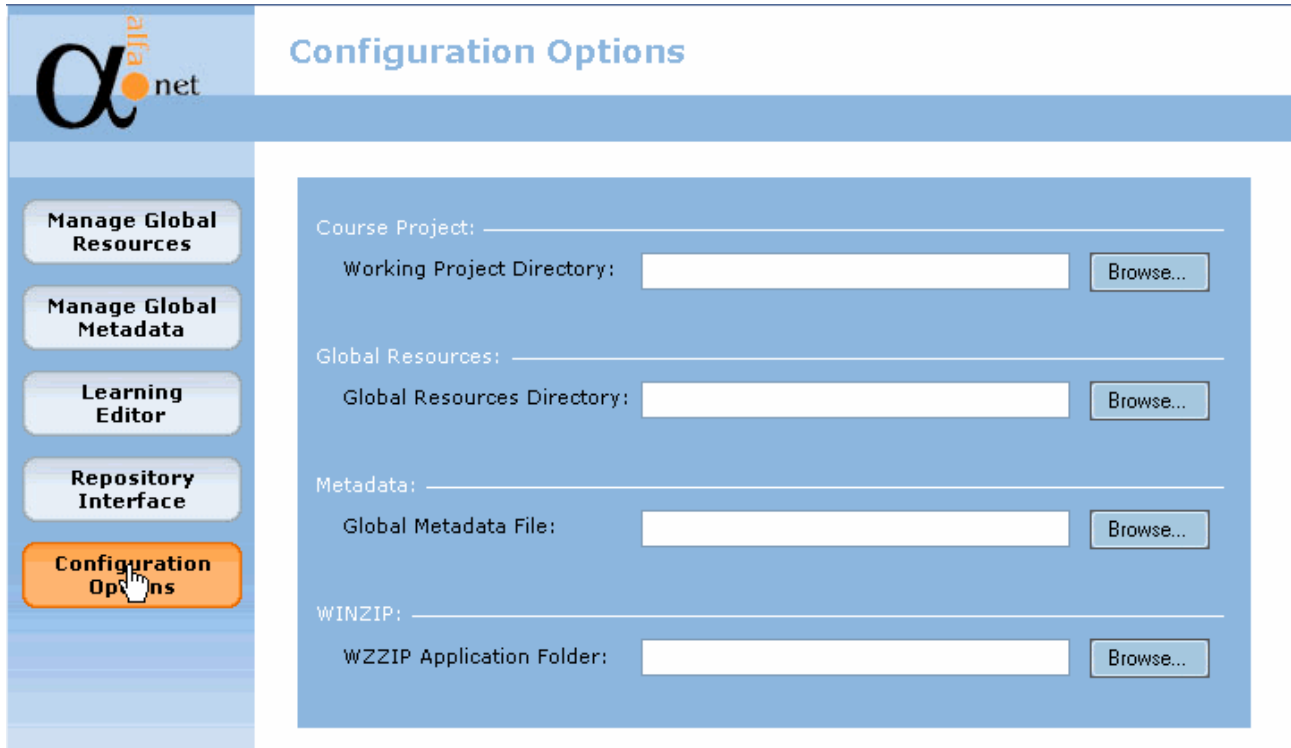
If you have any problem about the use of this application please visit the following page to see the 'Dav Explorer User Manual':

<http://www.ics.uci.edu/~webdav/um.html>

### 3.5 Configuration Options

The Configuration Options module allows the user to configure some configuration settings of the Authoring Tool.

To accede to this module please choose, in the main menu, the button that says “Configuration Options”.

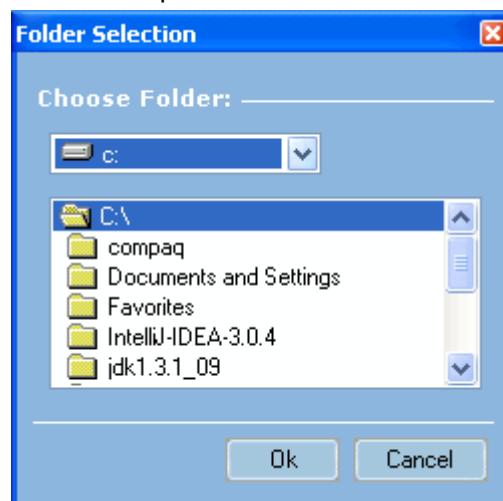


The settings to configure are:

- **Working Project Directory** – Text value to the user specify the folder where he wants to create the course projects.

When the user creates a new course, the window will assume by default this value, but if the user changes the location of the new course away from this folder, the tool will assume that folder as the new working folder and will forget the former value.

If the user prefers, the tool let the user input directly the value on the text box. But the user has, also, the option to click in the “Browse...” button near the textbox to navigate through the file system in order to choose the target folder. When the user clicks on the “Browse...” button the tool presents a form like the below picture:



**Note:** The user must choose a folder on the folders list by double clicking on the chosen one, otherwise the tool will assume the parent folder.

- **Global Resources Directory** – Text value to the user specify the folder where is located the global resources that he wants to use or define.

This directory needs to have a file named “resources.xml” where are defined the global resources of that folder.

This is the file that the Manage Global Resources module will try to open and save the resources on that module.

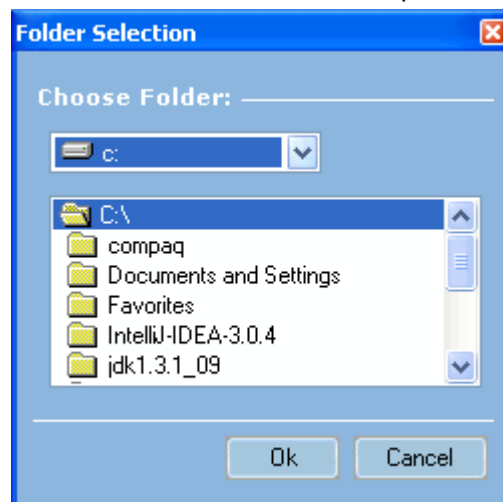
If the “resources.xml” doesn’t exist on that folder the Manage Global Resources will give an error on loading the global resources, but the user can still work and define the resources.

This means that the user can create several kinds of resources and organize them in different folders and then load them for the courses.

If the user decides to save the Global Resources the tool will save them on the “resources.xml” or, if doesn’t exist, will create that file.

If the user prefers, the tool let the user input directly the value on the text box. But the user has, also, the option to click in the “Browse...” button near the textbox to navigate through the file system in order to choose the target folder.

When the user clicks on the “Browse...” button the tool presents a form like the below picture:



**Note:** The user must choose a folder on the folders list by double clicking on the chosen one, otherwise the tool will assume the parent folder.

- **Global Metadata File** – Text value to the user specify the file where is defined the global metadata that he wants to use or define.

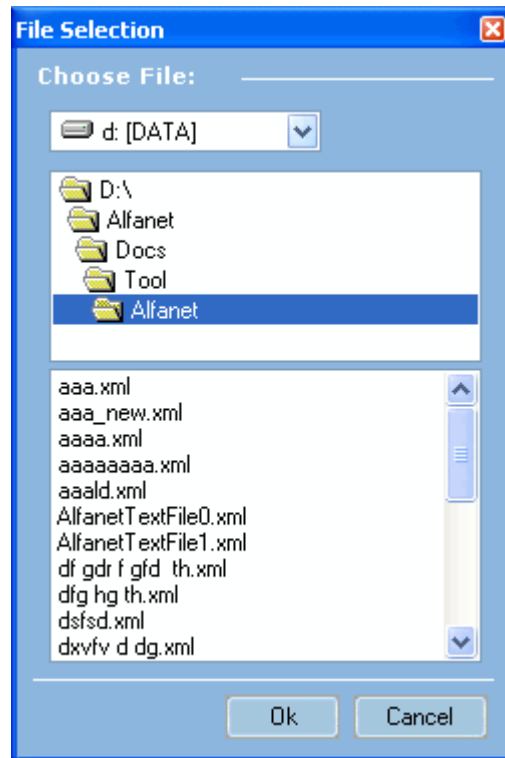
This is the file that the Manage Global Metadata module will try to open and save the global metadata defined on that module.

If the chosen file doesn’t have the right information format, the Manage Global Metadata module will appear empty but gives no error.

This philosophy of having global metadata saved in files means that the user can create several metadata templates and organize them in different files and then load the most adequate for the courses.

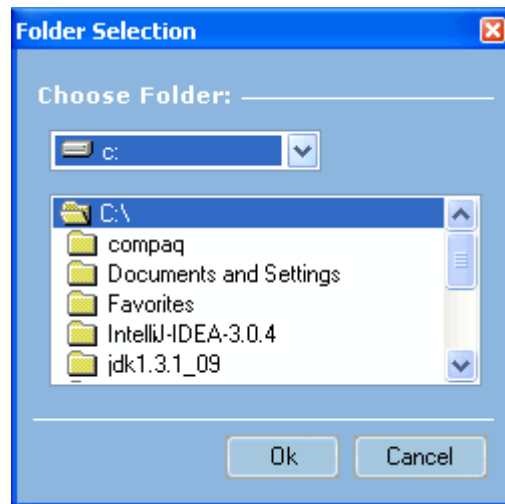
If the user prefers, the tool let the user input directly the value on the text. But the user has, also, the option to click in the “Browse...” button near the textbox to navigate through the file system in order to choose the target file.

When the user clicks on the “Browse...” button the tool presents a form like the below picture:



**Note:** Be very careful with the file that you choose because, if you choose to save the global metadata information, the tool **will overwrite the content without an overwriting alert message**.

- **WZZIP Application Folder** – Text value to the user specify the folder where is located the WZZIP application.  
This setting refers to a special feature of the Authoring Tool that allows the user to create WinZip files that contains a course and all related resources. These files follow the *IMS-CP* indications for interchange files.  
To be able to perform this action the authoring tool needs to have an external application.  
This application is the “WinZip Command Line Support Add-On” and allows the tool to create WinZip files.  
To more information about this application please read the “Create Package Interchange File Option” section of this manual.  
The value that you need to provide is the folder where is located, after installation, the “wzip.exe” file (normally is the folder where is installed the WinZip application).  
If the user prefers, the tool let the user input directly the value on the text. But the user has, also, the option to click in the “Browse...” button near the textbox to navigate through the file system in order to choose the target folder.  
When the user clicks on the “Browse...” button the tool presents a form like the below picture:



**Note:** The user must choose a folder on the folders list by double clicking on the chosen one, otherwise the tool will assume the parent folder.



## 4. Learning Editor Sub Modules

The Learning Editor module is divided into several sub-modules that intend to represent the *IMS-LD* structure and hierarchy.

Besides the File Manipulation Options and the Create Package Interchange File Option, all other options are located in the Learning Editor menu.



**Note:** This menu will be active only when the user creates a new course or loads an existing one.

This menu tries to organize the several *IMS-LD* elements in a similar way as the standard.

This menu has two levels of interaction. For example, the four first options (“*Local Resources*”, “*Title & Metadata*”, “*Learning Objectives*” and “*Pre-Requisites*”) are first level options, this means that when the user chooses one of these options, on the right side of the tool will be presented to him the related form to work.

But the two other options (“*Components*” and “*Method*”) have a sub-menu. This happens because they are not an element but a group of concepts and elements that have something in common.

In the *Components* option we find objects that will be used and reused in the organization of the course.

When the user chooses to click in the “*Components*” button the menu changes to the related sub-menu. That *Components* sub-menu is similar to the following picture.



In the “*Method*” Option we find objects that organize the course in a sequential way.

When the user chooses to click in the “*Method*” button the menu changes to the related sub-menu. This *Method* sub-menu is similar to the following picture.



In the Learning Editor module menus the tool have always a “Learning Editor” button, like the picture below, which redirects the user to the main page of the module.



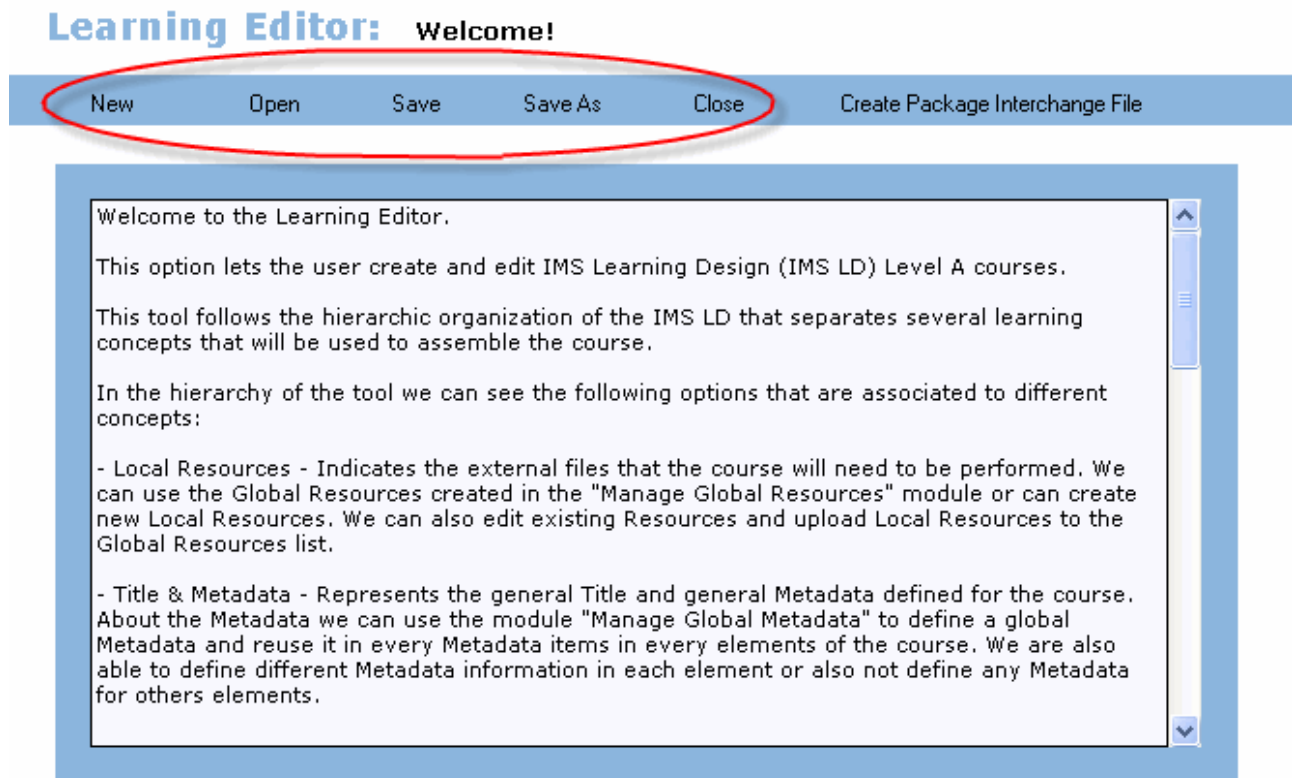
In the following section we define all the sub-modules of this module.

## 4.1 File Manipulation Options

In the main page of the Learning Editor we can find the File Manipulation Options.

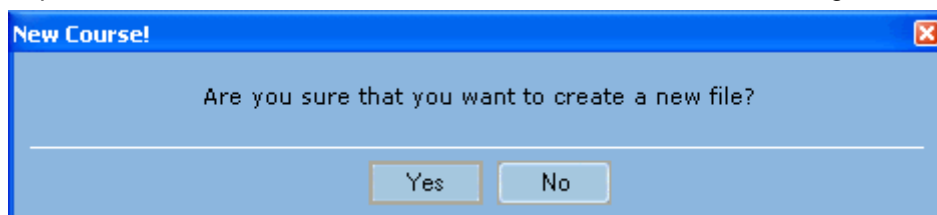
These options are related to the Creation, Loading and Saving of courses.

They are located in the right side of the screen as showed in the picture below:

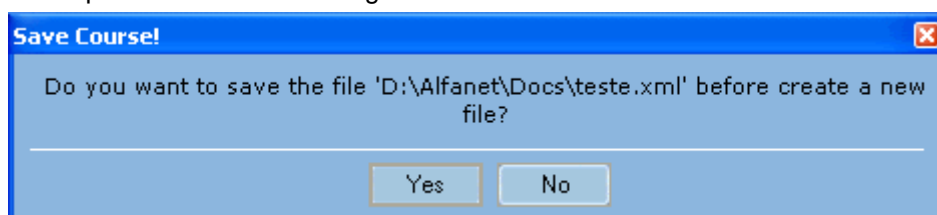


We have the following options:

- **New** – This is the feature that allows the user to create new Courses. When the user chooses this option the tool will ask him for a confirmation, with a message like the below picture, about this action to assure that was not a mistake of clicking.



If the user confirms his option and has already an open file, the tool will ask him if he wants to save the opened file with a message like the one below.



After these steps, finally, the user will be able to create the new course. The tool will present to him a form like the one below.

In this form the user can change the location of the new course by browsing in the file system drives and folders in the “Folder” section of the form.

The default folder will be “Working Folder” defined in the Configuration Options module. If the user chooses another folder to the course that new folder will be the new “Working Folder”.

In the “New Course Title” section the user can choose the name of the new course. This name will be the general title of the course and the name of the physical file with the “.xml” extension.

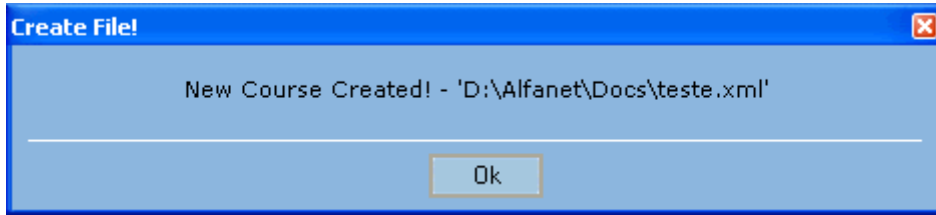
If needed the tool will show an overwriting alert, like the below picture, about a previous file with the same name.

After these steps the tool will load the needed data to start working.

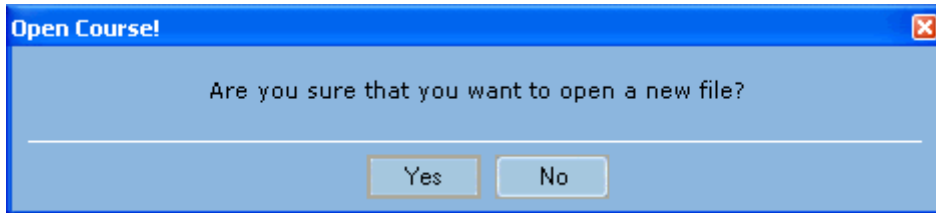
The tool loads the General Resources data (defined in the *Manage Global Resources* module).

The tool asks to the user if we want to use the Global Metadata information (defined in the *Manage Global Metadata* module) as a template to the metadata sections of the new elements of the course. The message is like the below picture:

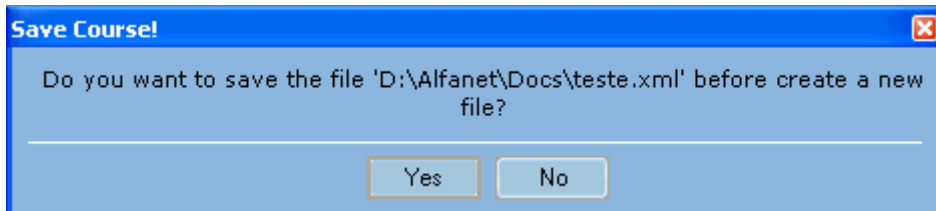
Finally if there was no error in the creation of the new course the tool will show a message like the one below and will activate the Learning Editor menu, the “Save”, “Save As” and “Close” options and the “Create Package Interchange File” option.



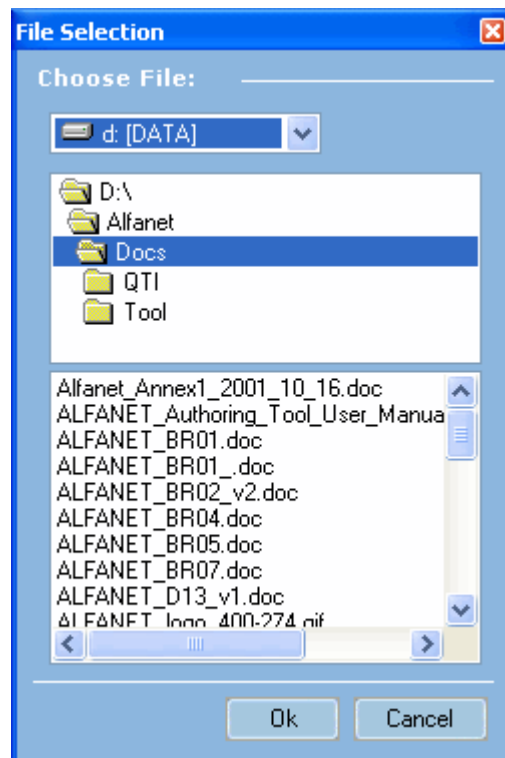
- **Open** – This is the feature that allows the user to open existing Courses and load them to the tool. When the user chooses this option the tool will ask him for a confirmation, with a message like the below picture, about this action to assure that was not a mistake of clicking.



If the user confirms his option and has already an open file, the tool will ask him if he wants to save the opened file with a message like the one below.



After these steps, finally, the user will be able to load the course. The tool will present to him a form like the one below.



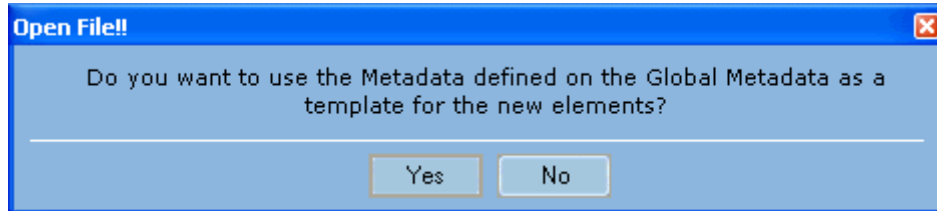
In this form the user can browse the file system drives and folders to choose the target file.

The default folder will be “Working Folder” defined in the Configuration Options module. If the user chooses a file from another folder, that will be the new “Working Folder”.

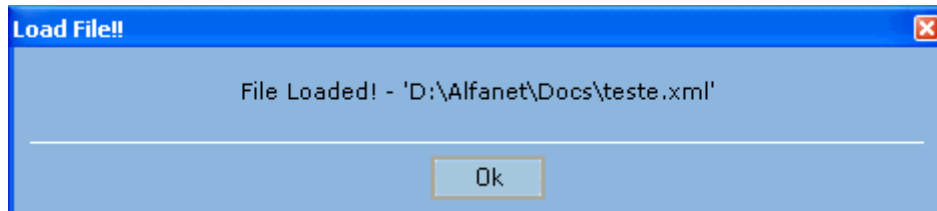
After these steps the tool will load the needed data to start working.

The tool loads the General Resources data (defined in the *Manage Global Resources* module).

The tool asks to the user if he wants to use the Global Metadata information (defined in the *Manage Global Metadata* module) as a template to the metadata sections of the new elements of the course. The message is like the below picture:

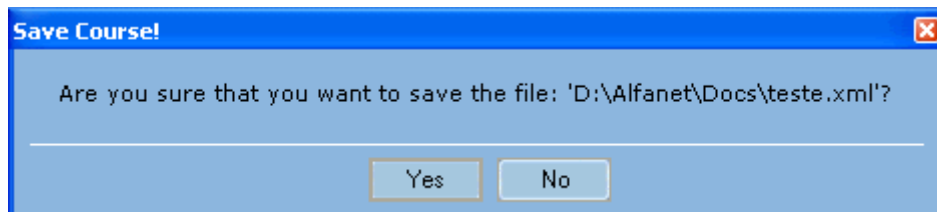


Finally if there was no error loading the course the tool will show a message like the one below and will activate the Learning Editor menu, the “Save”, “Save As” and “Close” options and the “Create Package Interchange File” option.

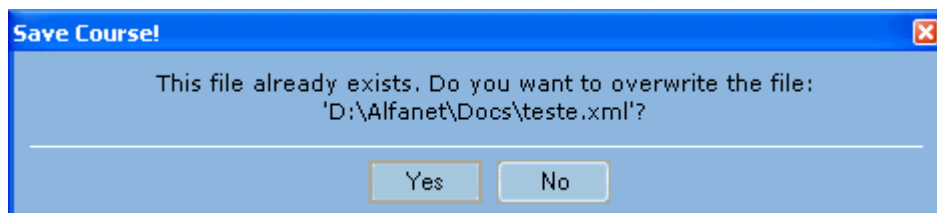


- **Save** – This feature it's only available if there is currently an opened file and allows the user to save the opened Course.

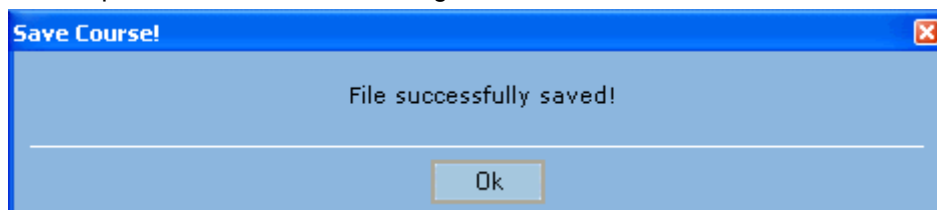
When the user chooses this option the tool will ask him for a confirmation, with a message like the below picture, about this action to assure that was not a mistake of clicking.



If the user confirms his option the tool will present an overwriting alert like the below picture.

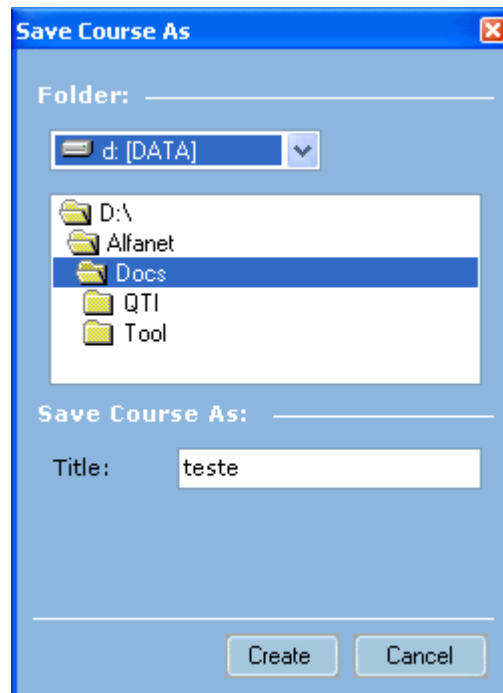


After these steps, finally, the course will be saved and, if there was no problem saving the course, the tool will present to the user a message like the one below.



- **Save As** – This feature it's only available if there is currently an opened file and allows the user to save the opened Course with a different name from the actual one.

When the user chooses this option the tool will presents a form, like the picture below, to let the user decide the new name and location of the file.



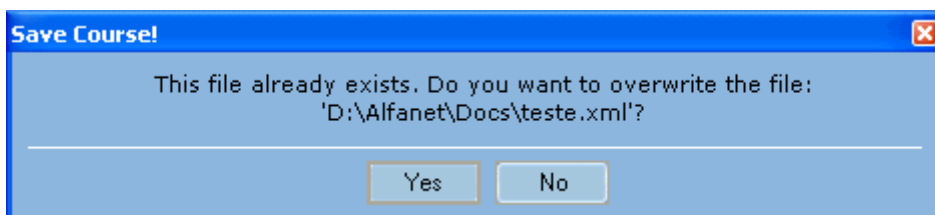
In this form the user can change the location of the new course by browsing in the file system drives and folders in the “Folder” section of the form.

The default folder will be “Working Folder” defined in the Configuration Options module. If the user chooses another folder to the course that new folder will be the new “Working Folder”.

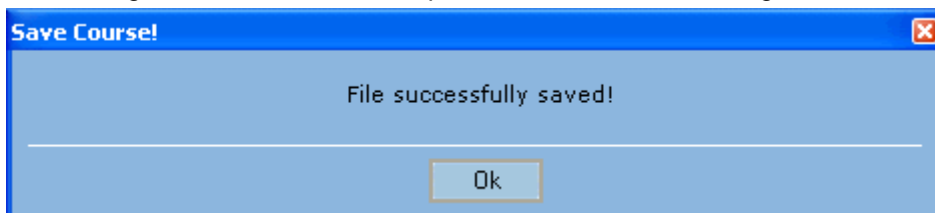
In the “Save Course As” section the user can choose the new name of the course. By default the value is the same name as the opened file.

This title will be the name of the physical file with the “.xml” extension.

If needed, the tool will show an overwriting alert, like the below picture, about a previous file with the same name.

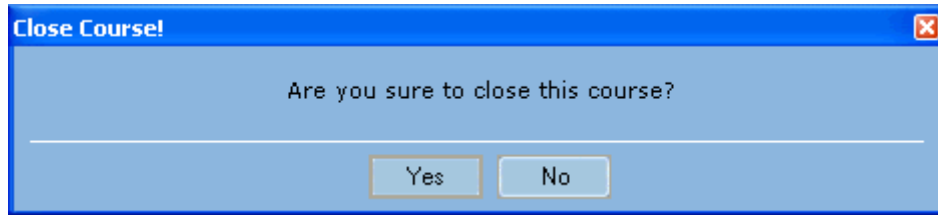


If the user says Yes to the last message, finally, the course will be saved and, if there was no problem saving the course, the tool will present to the user a message like the one below.

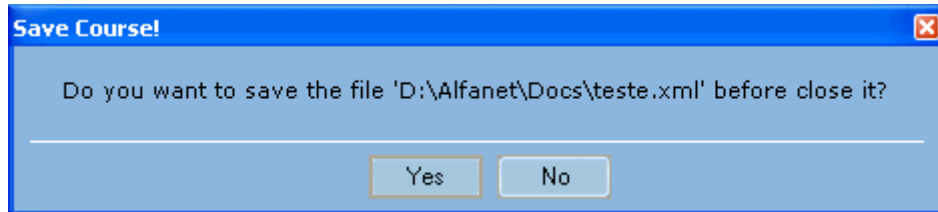


- **Close** – This feature it's only available if there is currently an opened file and allows the user to close the opened Course.

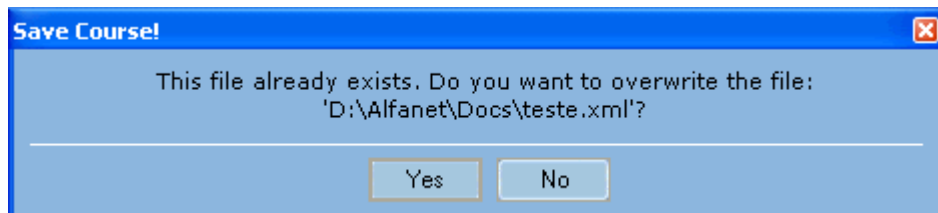
When the user chooses this option the tool will ask him for a confirmation, with a message like the below picture, about this action to assure that was not a mistake of clicking.



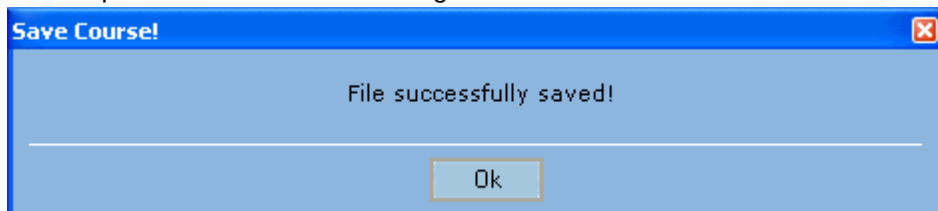
If the user confirms his option the tool will ask to him if he wants to save the opened file before close it.



If the user chooses to save the file, the tool will present an overwriting alert like the below picture.



After these steps, finally, the course will be saved and, if there was no problem saving the course, the tool will present to the user a message like the one below.



After closing the file, this feature will deactivate the Learning Editor menu, the "Save", "Save As" and "Close" options and the "Create Package Interchange File" option.



## 4.2 Create Package Interchange File Option

The Create Package Interchange File option is a feature of the Learning Editor that allows the user to create Package Interchange Files that follows the *IMS-CP* specification.

The Package Interchange File is a single file that aggregates the course definition file and all related resources.

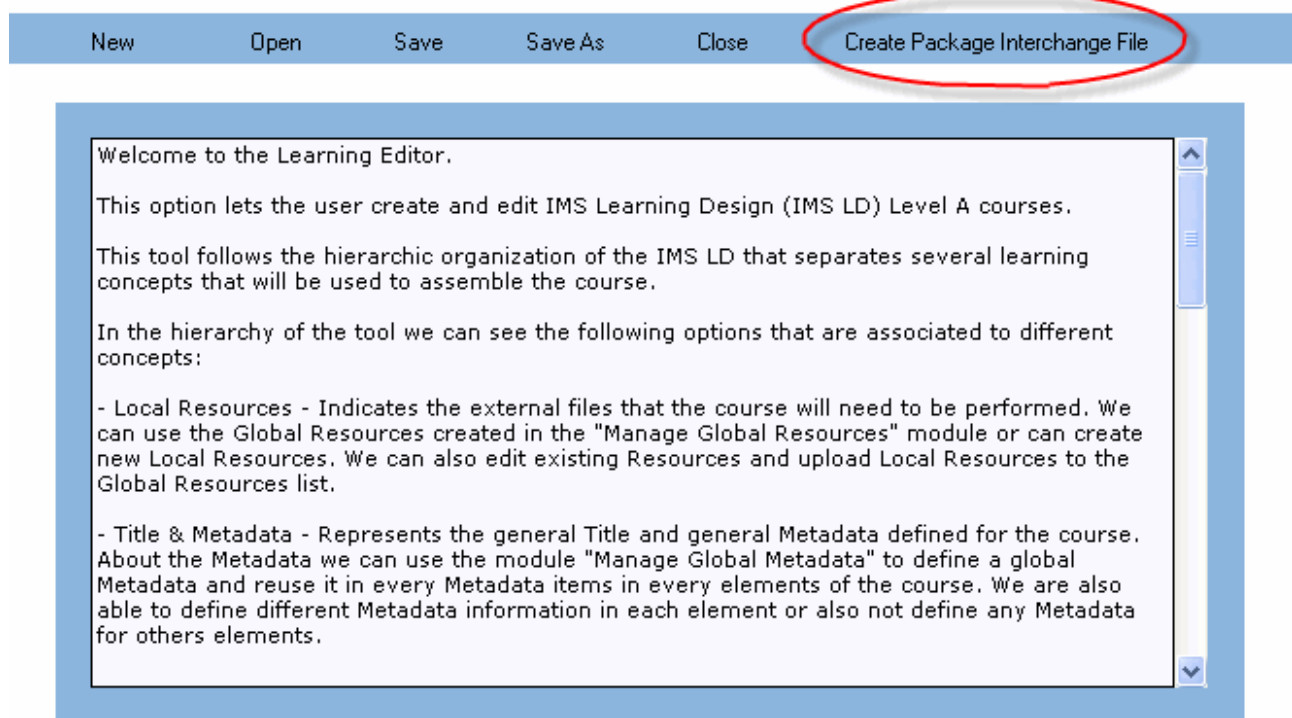
In the *IMS-CP* specification we can find the following description about the Package Interchange File:

*“Package Interchange File – a single file, (e.g., ‘.zip’, ‘.jar’, ‘.cab’) which includes a top-level manifest file named “imsmanifest.xml” and all other physical files as identified by the Manifest. A Package Interchange File is a concise Web delivery format, a means of transporting related, structured information. PKZip v2.04g (.zip) is recommended as the default Package Interchange File format. Any ZIP file format MUST conform to RFC1951.”*

The manifest file referred in the specification can be seen as the open file edited in the Authoring Tool.

This feature is located in the right side of the Learning Editor screen as showed in the picture below:

### Learning Editor: Welcome!



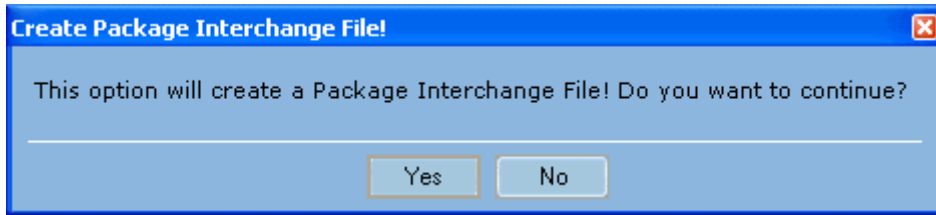
This option is only active when the Authoring Tool has an open course file. If the user didn't open or closed a file, this option will be inactive.

This feature uses an application named "WinZip Command Line Support Add-On".

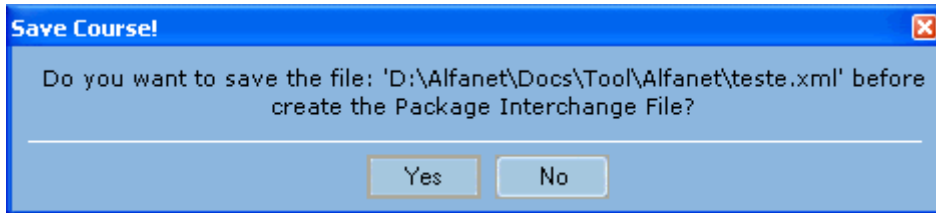
For more information about the "WinZip Command Line Support Add-On" installation and configuration please read the "How to Install and Configure the WinZip Command Line Support Add-On" section of this sub-module.

If there is an open Course and this option is correctly configured the creation of a Package Interchange File follows the following steps.

When the user chooses this option the tool will ask him for a confirmation, with a message like the below picture, about this action to assure that was not a mistake of clicking.

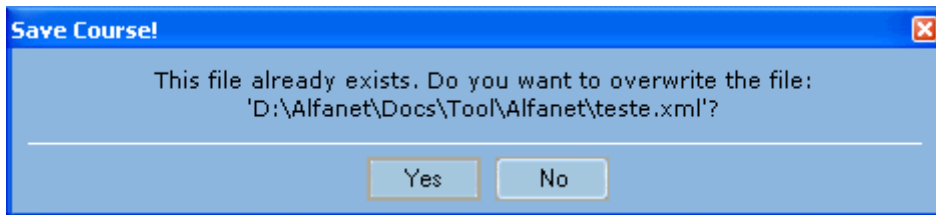


If the user confirms his option the tool will ask to him if he want to saves the last changes to the open file before create the Package Interchange File, like the below picture describes.

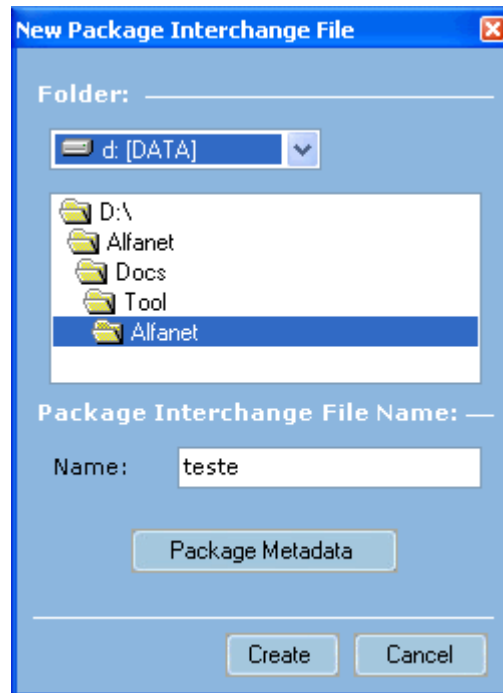


**Note:** The Package Interchange File will assume the physical course file, so if the user says no, the feature will continue, but will assume the saved file and not the last modifications made since the last saving of the file.

If the user confirms the saving of the file, the tool will show him an overwriting message, like the below picture, to make sure that he wants to overwrite the last saved file.



After these steps, finally, the user will be able to create the new Package Interchange File. The tool will present to him a form like the one below.



In this form the user can change the location of the new file by browsing in the file system drives and folders in the “Folder” section of the form.

The default folder will be folder where is located the opened file.

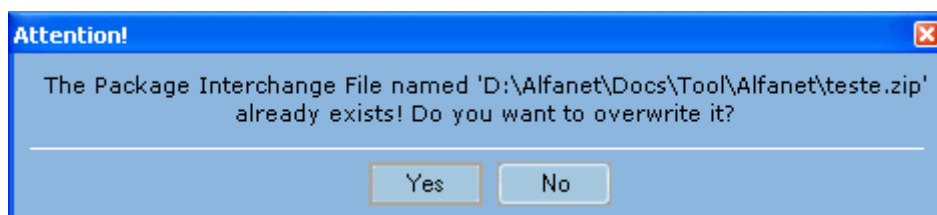
In the “Package Interchange File Name” section the user can choose the name of the new file. This name by default will similar to the general title of the opened file but the user can change this name to another one.

This name will be the name of the physical file with the “.zip” extension.

This form has also a button called “Package Metadata”. This button allows the user to define the Metadata associated with the Package Interchange File.

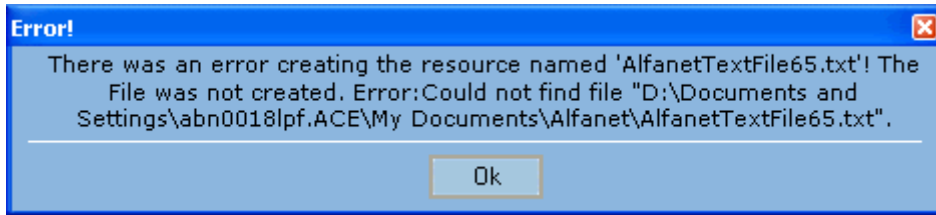
This Metadata can be seen as a metadata for the packaging and not a general metadata for the course. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step.

If needed, the tool will show an overwriting alert, like the below picture, about a previous file with the same name.

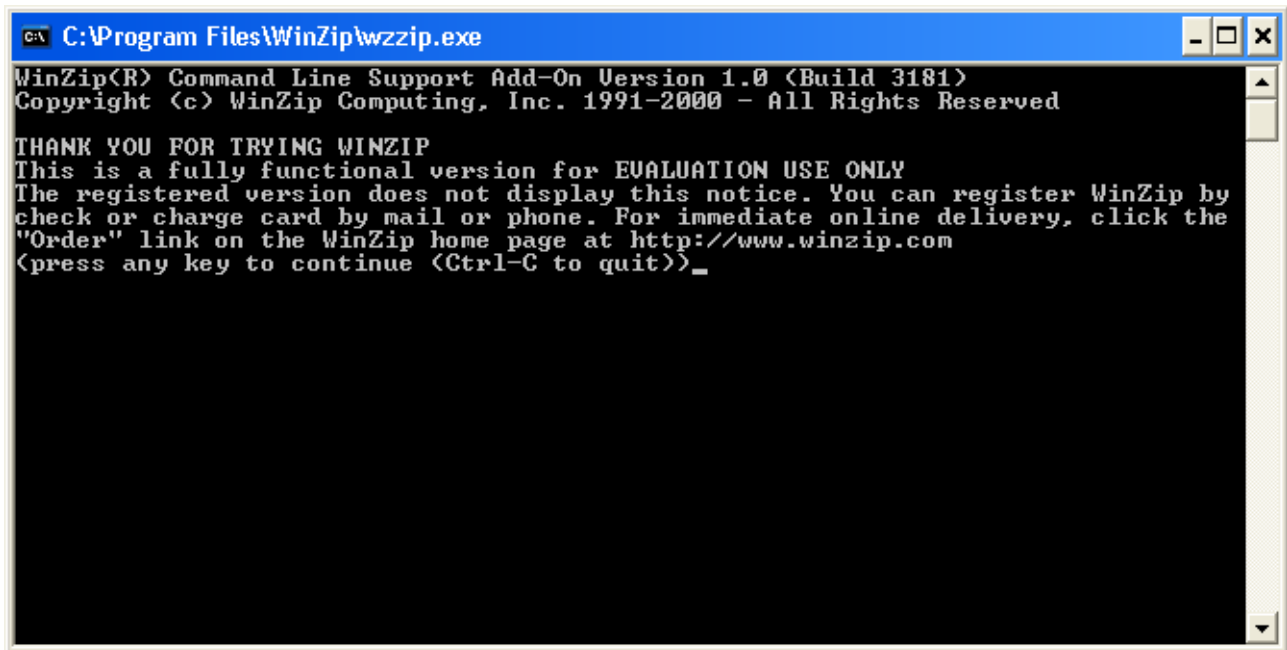


After these steps the tool will process the course file, name it as “imsmanifest.xml” and copy to a temporary folder. The toll will process every resources copying them to the same temporary folder.

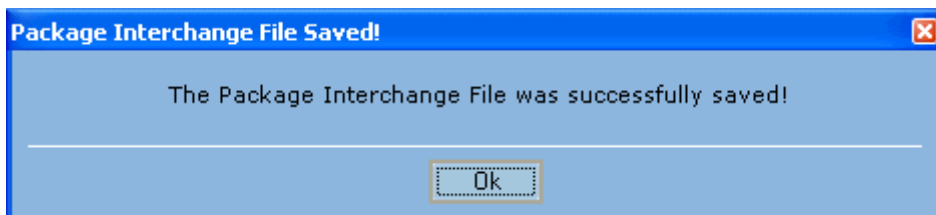
**Note:** One frequent error in this phase is the presence of Resources with references to unavailable files (files that have been moved or deleted). If this fact happens **the tool will abort this procedure** and send a message like the following picture.



If there is no error in the creation of the temporary files, the tool will proceed to the creation of the WinZip file. Probably the tool will present a message, proceeding from the "WinZip Command Line Support Add-On", like the following picture.



The user only needs to press any key and it will be able to see the evolution of the creation of the WinZip file. If there was no problem in the creation of the Package Interchange File the tool will present a success message like the picture below.



#### 4.2.1 How to Install and Configure the WinZip Command Line Support Add-On

The Create Interchange File option is the feature that allows the user to include the course and all Resources in only one file.

To do this the user must have installed on his computer the WinZip 8.0 or higher. To install the WinZip please visit the following page.

<http://www.winzip.com>

Download the latest version and install it.

After installed the WinZip the user also needs to install the 'WinZip Command Line Support Add-On'. To do this, please, visit the following page:

<http://www.winzip.com/wzcline.htm>

Download the latest WinZip Command Line Support Add-On file (it's not recommended to download a beta version!) and install it.

After installed the "WinZip Command Line Support Add-On" please go to the "Configuration Options" module of the Authoring Tool and indicate on the "WZZIP Application Folder" option the folder where you installed the WinZip Command Line Support Add-On (the folder that have the "wzzip.exe" file). Normally is the folder where is installed the WinZip.

## Configuration Options

The screenshot shows a configuration dialog box with the following fields and values:

- Course Project:** (empty)
- Working Project Directory:** D:\Alfanet\Docs\Tool\Alfanet
- Global Resources:** (empty)
- Global Resources Directory:** (empty)
- Metadata:** (empty)
- Global Metadata File:** C:\Program Files\Common Files\System\
- WINZIP:** (empty)
- WZZIP Application Folder:** C:\Program Files\WinZip

The 'WZZIP Application Folder' field and its 'Browse...' button are circled in red.

Only after these steps the Authoring Tool will be able to create Package Interchange files compatible with the *IMS-CP* standard.

## 4.3 Items

The Items are not quietly a module of the Learning Editor but a concept that is present in a lot of objects inside the course.

In the *IMS-LD* specification we can find the following description about the Items:

*“When a component, a learning objective, or a prerequisite needs a resource, an ‘item’ element is used in the similar way as in the organization part of IMS Content Packaging. The learning design provides a semantic context for these items, so that runtime systems can know what to do with the resource. For example in the following case:*

*<learning-objectives><item identifierref="o123"/></learning-objectives> it is clear that the resource with identifier ‘o123’ is a learning objective description. In the case of:*

*<activity-description><item identifierref="o345"/></activity-description> the item is an activity description. A runtime system can position the learning objectives in a certain place in the user-interface, different from the activity-description, and activity-descriptions can be handled differently from other learning content (this can vary from implementation to implementation, within the boundaries of the behavior descriptions provided in this information model).”*

In other words the Items are the object that will encapsulate and refer the Resources. According to the context that the items are used, the user can define for the same resource different kinds of approaches to the content.

The items concept and interface will be describe in just one section of this manual for a better and efficient description.

In other words, in every place that is used, the items will correspond to the same interface that will describe in this section.

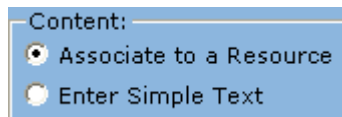
The form that correspond to the items is like the below picture.

- **Title** - Represents the title that the user will give to the item resource. This is a mandatory field, so if this field is not filled the Item can't be created. The tool, if that field is not defined, by default gives an automatic name to the item (normally a sequential name like in the above picture);

**Note:** Despite this value is not the ID of the Item (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some

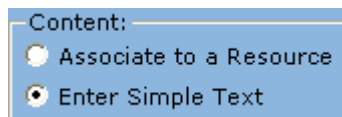
existing item or other kind of element. **According to this, try to put an explicit value on the title of the Item to let you easily remember and identify the different items;**

- **Parameters** – Text value to the parameters to be passed during runtime;
- **Is Visible** – Option field that allows the user to decide if this item will be visible to the end user of the Course;
- **Metadata** – Button that allows the user to define the metadata associated with the Item. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Content** – In this interface the tool let the user to decide if he wants to use an existing resource, create a new one or simply write a text that the tool will convert to a new resource.  
In this last option the tool creates one text file, with the text written by the user, in the working folder and one resource related to that text file.  
If the user wants to use an existing resource or create a new one, he must choose the “Associate to a Resource” option like the below picture.



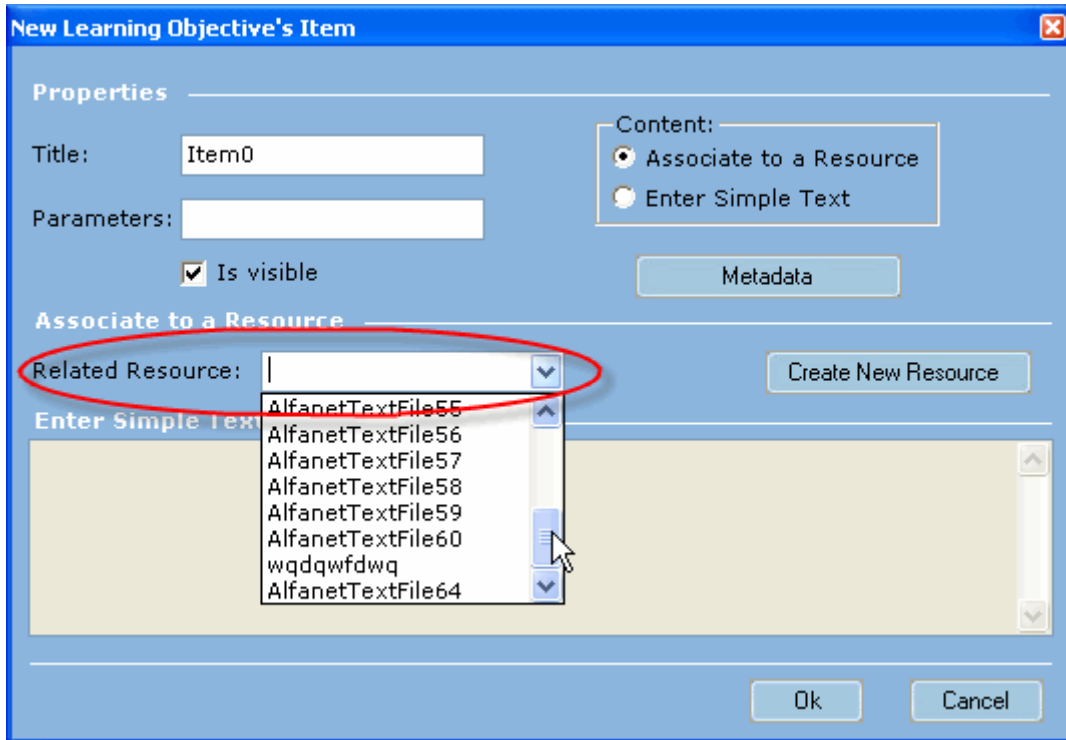
The image shows a blue-bordered box with the label 'Content:'. Below the label are two radio button options. The first option, 'Associate to a Resource', has a filled radio button, indicating it is selected. The second option, 'Enter Simple Text', has an empty radio button.

Otherwise if the user wants to just write a text that will be converted to a new resource, he must choose the “Enter Simple Text” option like the below picture.

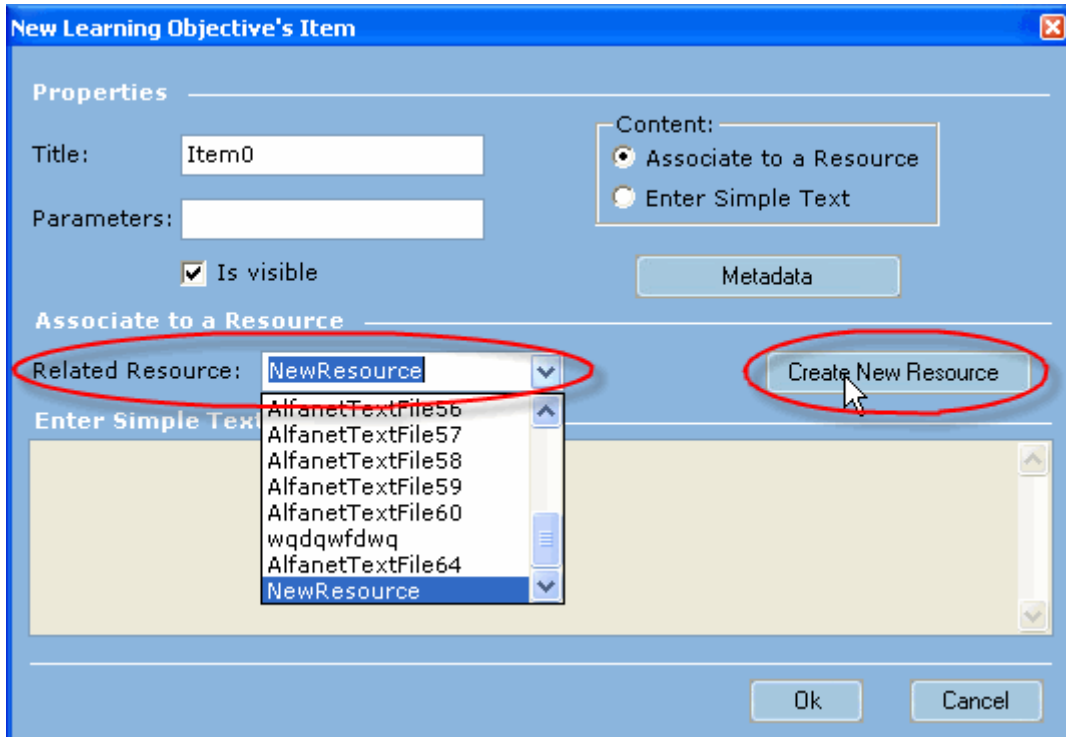


The image shows a blue-bordered box with the label 'Content:'. Below the label are two radio button options. The first option, 'Associate to a Resource', has an empty radio button. The second option, 'Enter Simple Text', has a filled radio button, indicating it is selected.

- **Related Resource** – List of ID’s of the existing resources that the user can choose to relate to this item.  
This field will be active only if the user chooses the “Associate to a Resource” option on the Content field.  
Is this field is active, the user must choose a resource from the list or cancel the operation, otherwise the user can’t advance.



- **Create New Resource** – Button that allows the user to see create a new Local Resource without having to cancel this operation. This field will be active only if the user chooses the “Associate to a Resource” option on the Content field. This option opens a form similar to the option “New Local Resource” of the Local Resources sub-module section. For more information read that section. The resource created in this operation will be chosen resource in the “Related Resource” list like the below picture shows.





- **Enter Simple Text** – Text value where the user can create a text that will be the content of the item.

This field will be active only if the user chooses the “Enter Simple Text” option on the Content field.

Is this field is active, the user must write some text or cancel the operation, otherwise the user can't advance.

The screenshot shows a dialog box titled "New Learning Objective's Item". It has several sections: "Properties" with fields for "Title" (containing "Item0") and "Parameters", and a checked "Is visible" checkbox; "Content" with two radio buttons, "Associate to a Resource" and "Enter Simple Text", where the latter is selected and circled in red; "Associate to a Resource" with a "Related Resource" dropdown set to "NewResource" and an active "Create New Resource" button; and a large text area titled "Enter Simple Text" which is also circled in red and contains the text "This is a simple text!!". At the bottom, there are "OK" and "Cancel" buttons.

This text will be saved automatically by the tool in a text file (.txt) in the working folder.

The tool will create too a resource that refers to that file and will automatically associate this item to that new resource.

If after saving the item the user edits again the item he can see that:

- The “Content” option field maintains the value in “Enter Simple Text”;
- The “Related Resource” list is not active but has the new automatic resource as the chosen resource;
- The “Create New Resource” is not active;
- The “Enter Simple Text” field is active and has the text written in the creation.

**Edit Pre-Requisite's Item**

**Properties**

Title:

Parameters:

Is visible

**Associate to a Resource**

Related Resource:

**Enter Simple Text**

**Content:**

Associate to a Resource

Enter Simple Text

When the user is creating a new Item or editing an existing one the difference will be only that, in the first occasion the form will appear empty except the Title field, which will have a default value. In the edition of the Item the form appears with the item data on the fields.

## 4.4 Local Resources

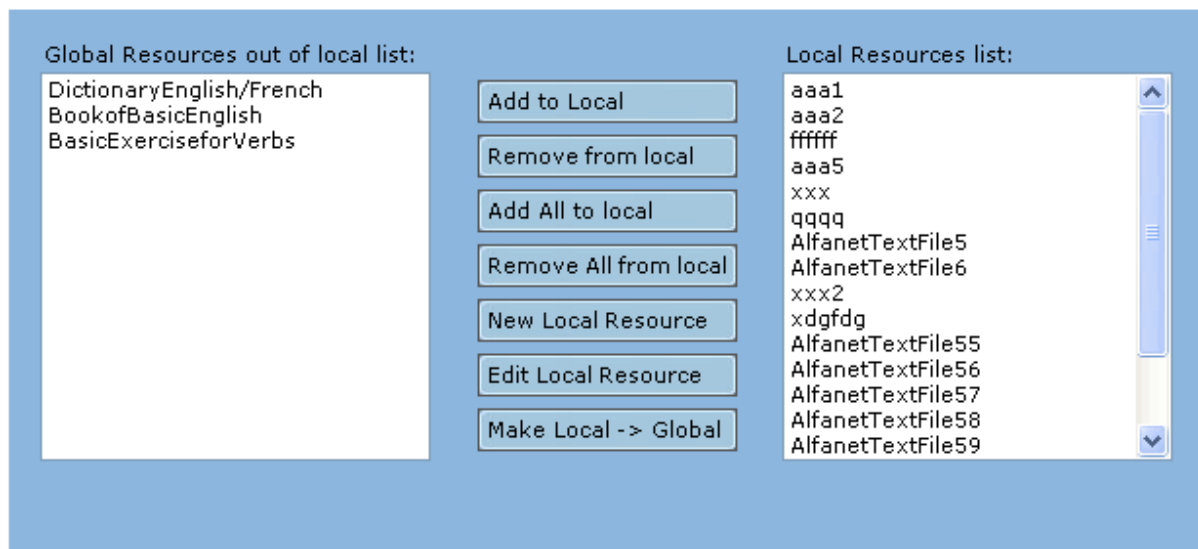
The Local Resources sub-module is the second option of the Learning Editor menu.



This feature allows the user to edit and create the resources that belongs to the opened course. To more information about what is a resource, please read the IMS Content Packaging section of the Introduction and the Manage Global Resources module section.

The below picture shows how is the form available to the user:

### Learning Editor: Local Resources Manager



In the left side of the form we can find the list of ID's of the Global Resources that belongs to the global resources file defined on the "Manage Global Resources" module and is not used on the course.

In the right side of the form we can find the list of ID's of the Resources used in the course that we call Local Resources. This list includes all Resources defined only in this course (called "really" Local Resources) and the Global Resources that are used in the course.

There are seven operations that the user can perform with the resources. Those operations are:

- **Add do Local;**
- **Remove from local;**
- **Add All to local;**
- **Remove All from local;**
- **New Local Resource;**

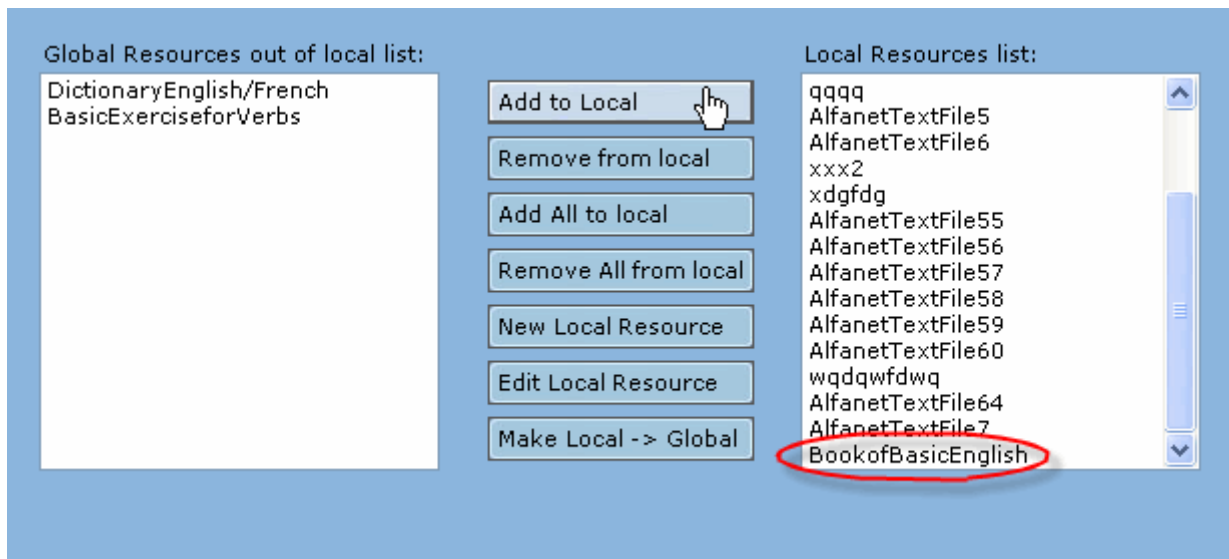
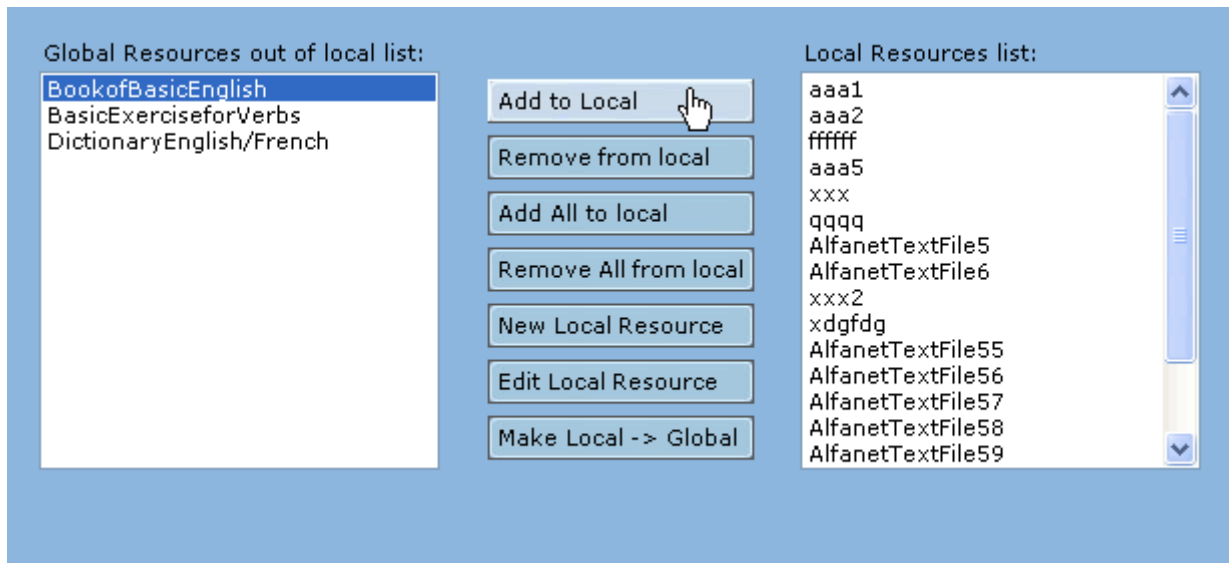
- **Edit Local Resource;**
- **Make Local → Global**

These operations are explained in the following sections.

#### 4.4.1 Add to Local

In this operation the user can choose a Resource from the Global Resources (left list) and add it to the Local Resources (right list).

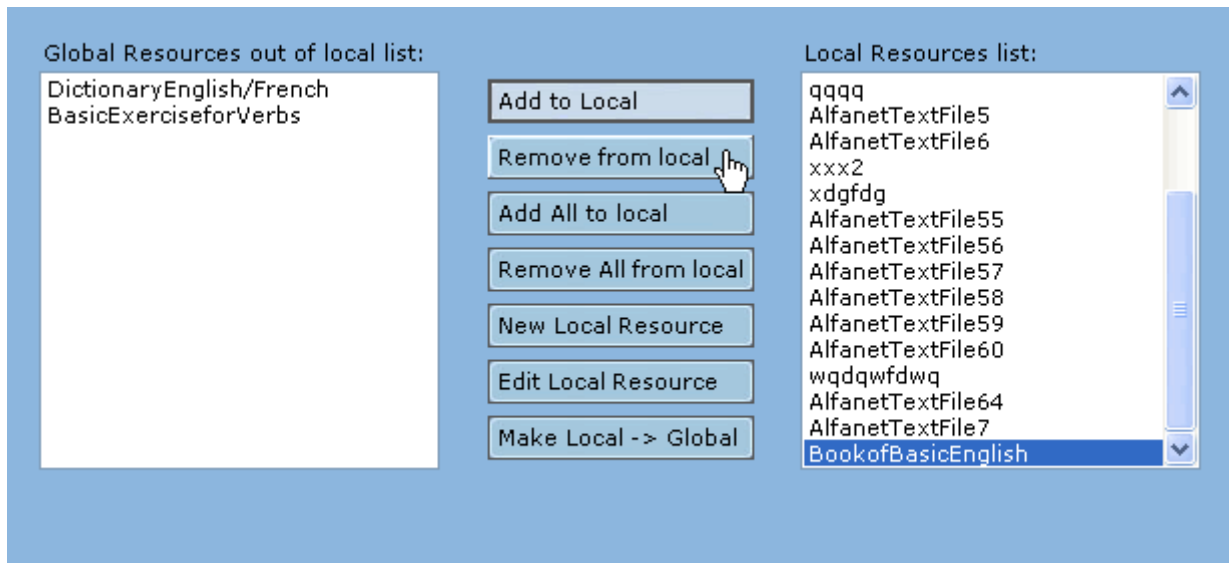
To perform this operation the user only as to choose one or more Resources from the left list and click on the “Add to Local” button like the following pictures represent:



#### 4.4.2 Remove from local

In this operation the user can remove a Resource from the Local Resources list.

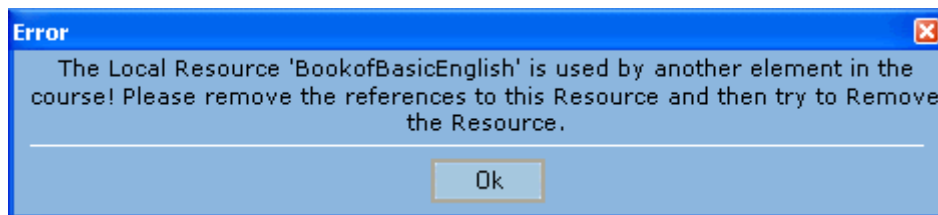
To perform this operation the user only as to choose one or more resource from the right list and click on the “Remove from local” button like the following picture represents:



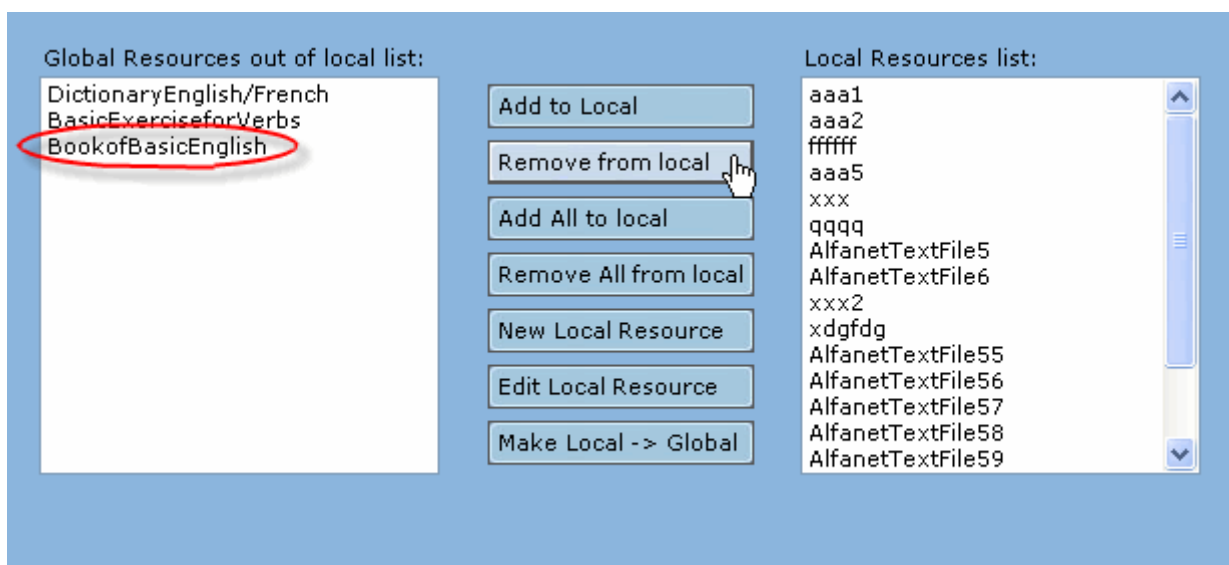
If the Resource is referenced in an element inside the course it's impossible to remove the resource, because it would cause referential problems inside the course.

To solve this problem the user will need to locate and remove all the references to the chosen resource by changing or deleting the items that refer this resource.

In this case the tool will show a message like the one below.

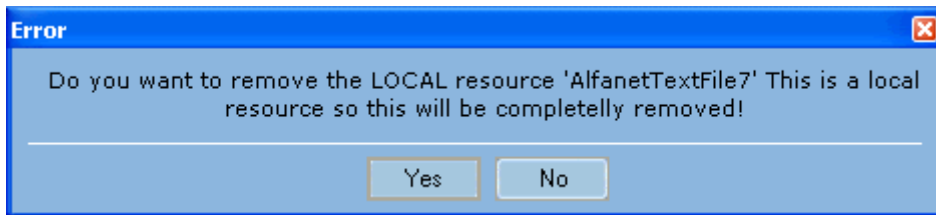


If the chosen Resource is not referenced by any element inside the course and **belongs to the Global Resources**, the Resource will be removed by the right list and will appear in the left list.



Otherwise if the chosen Resource is not referenced by any element inside the course and **doesn't belongs to the Global Resources** (this means that this is a “really” Local Resource), the Resource will be **permanently** removed.

In this case the tool will ask to the user if he wants to permanently remove the Resource like described in the following picture.



If the user chooses to permanently remove the Resource it will disappear from any list and will no longer exist.

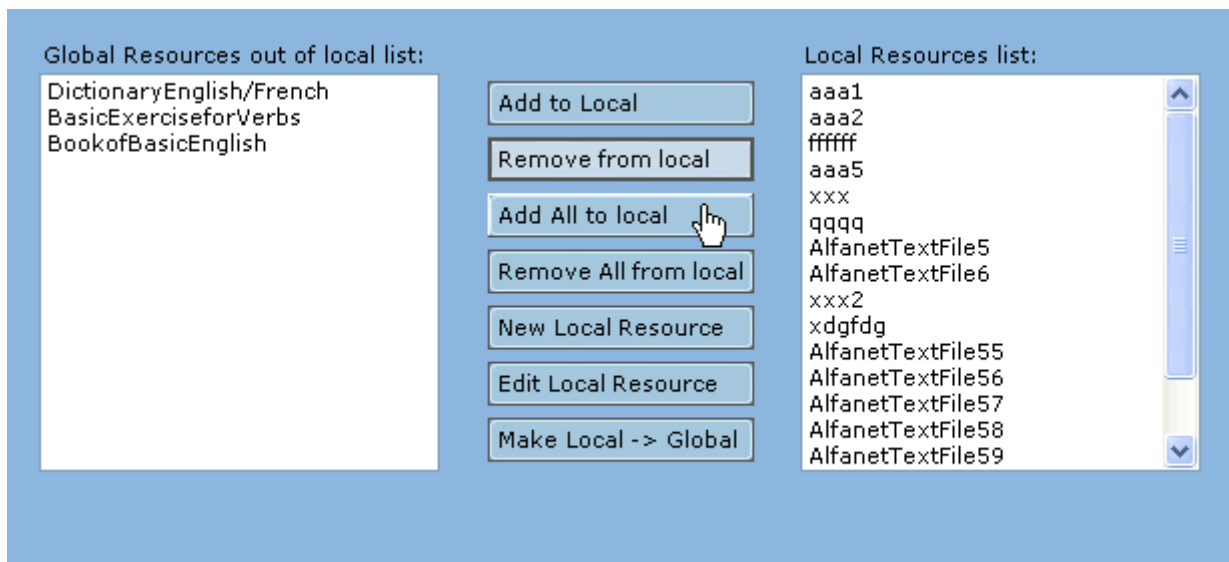
**Note:** The only way to recover the resource is to reopen the file without saving, losing every changes made since the last saving.

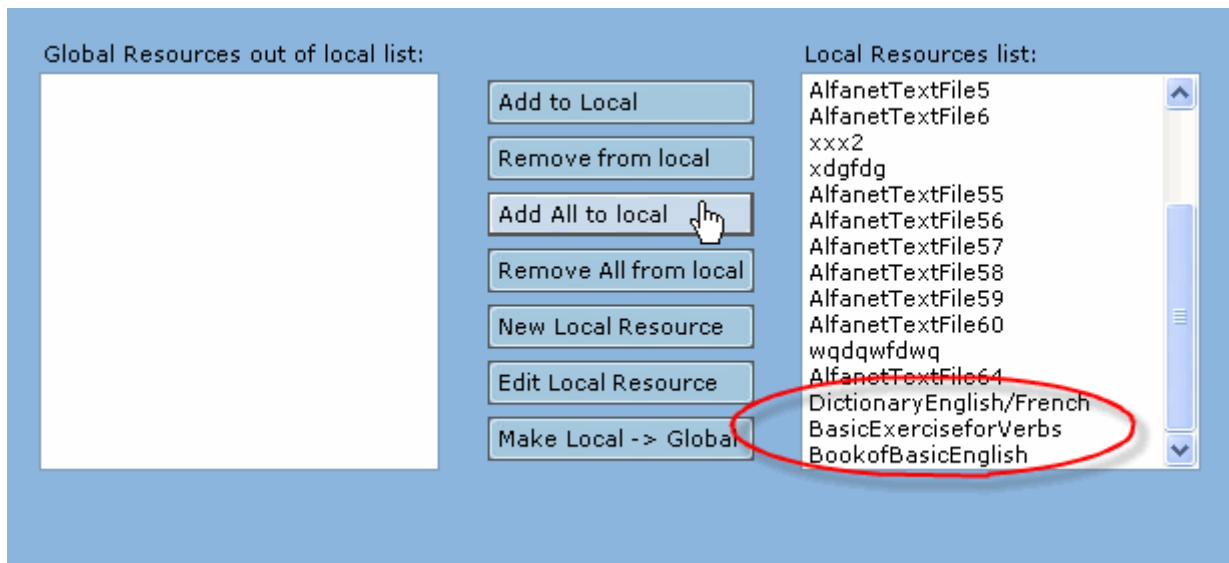
#### 4.4.3 Add all to local

This operation has the same logic as the “Add to Local” option but the logic is applied to all resources in the left list.

This can be seen as an easy way to import all the Global Resources to the course.

To perform this operation the user only has to click on the “Add all to local” button like the following pictures represent:





**Note:** As you can see from this moment the left list will be empty because all the Global Resources are used in the Course and are on the right list.

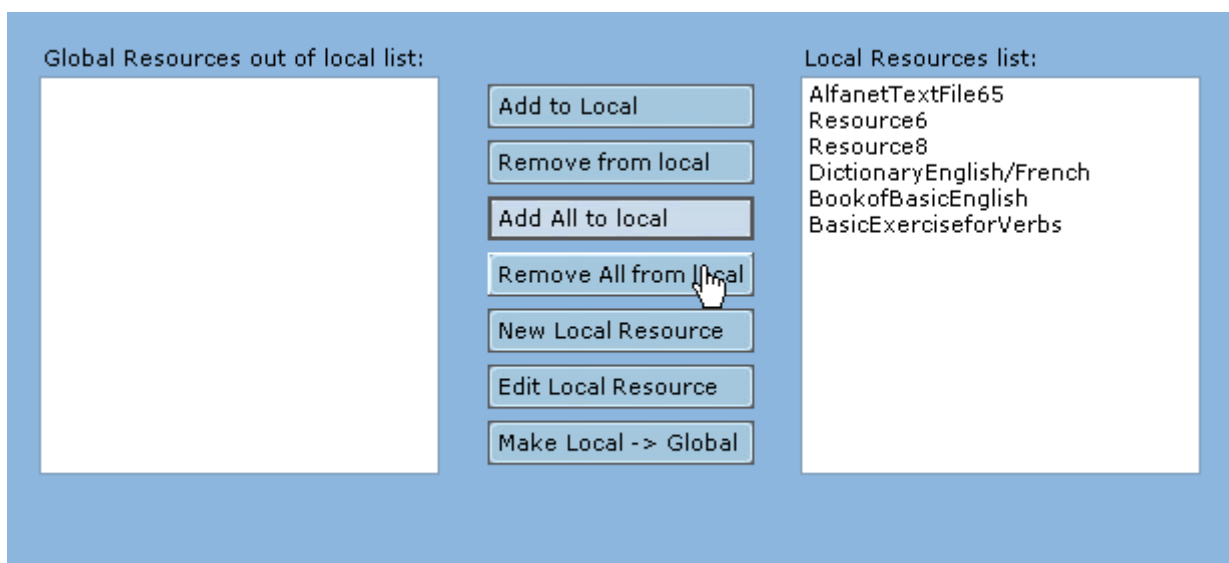
#### 4.4.4 Remove all from local

This operation has the same logic as the “Remove from local” option but the logic it’s applied to all resource in the right list.

This can be seen as an easy way to delete all the Resources of the course.

**Note:** If one of the Resources is not possible to remove or is a “really” Local Resource, the tool will not remove or ask to remove the Local Resource, as described in the “Remove from local” section, and abort the removal of the resources.

To perform this operation the user only has to click on the “Remove all from local” button like the following pictures represent:



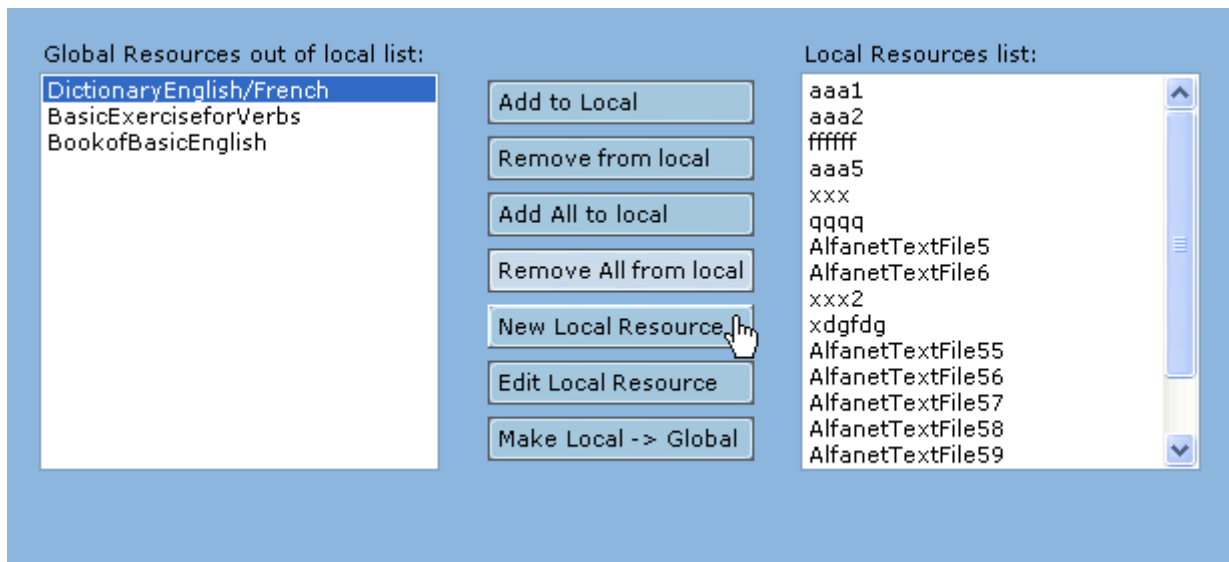


**Note:** If the tool could remove every Local resources the right list will be empty, because all the Local Resources are deleted, and the left list will have all the Global Resources defined.

#### 4.4.5 New Local Resource

In this operation the user can create a new Local Resource for this Course.

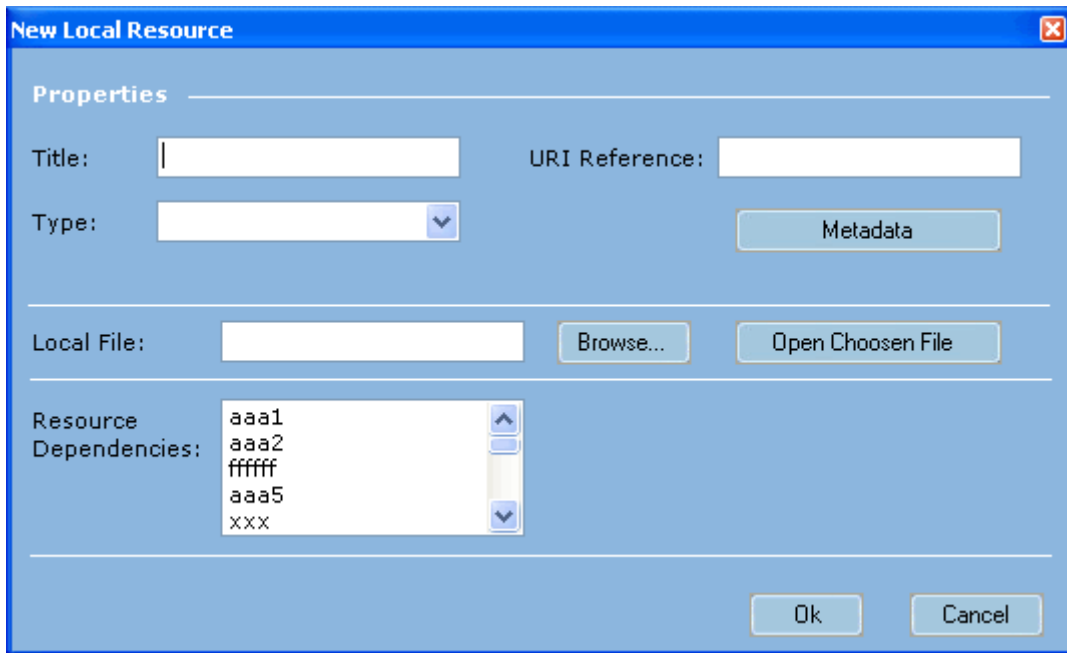
To perform this operation the user only has to click on the “New Local Resource” button like the following picture represents:



**Note:** The Resources created in this option will be “really” Local Resources that means that they will be defined only in this course.

When the user clicks on this option the tool will present to him a form to define the new resource data like the picture below.

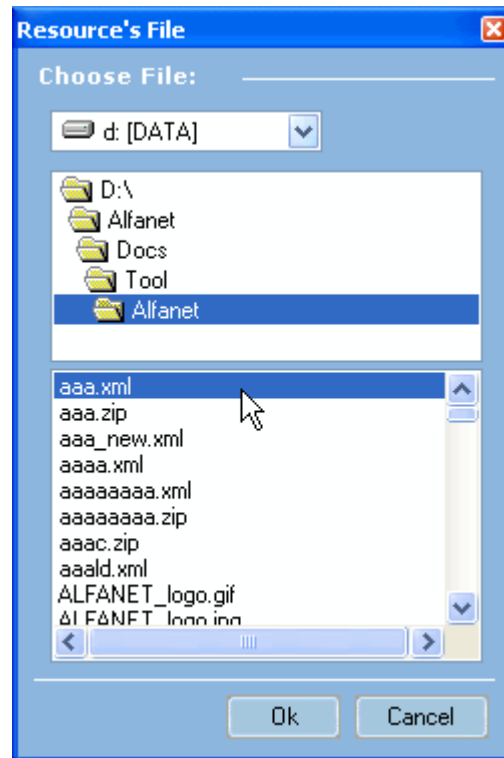




- **Title** - Represents the title that the user will give to the resource. This is a mandatory field, if this field is not filled the resource can't be created;

**Note:** Despite this value is not the ID of the Resource (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing resource or other kind of element. **According to this, try to put an explicit value on the title of the resource to let you easily remember and identify the different resources;**

- **URI Reference** - Reference to an external file that can't be transform in a local file, e.g. a web site;
- **Type** – Type of resource. A resource can be “webcontent” or “imsldcontent”. The “webcontent” resource means that this is a file that a browser can show. The “imsldcontent” resource means that this is other kind of file. This is a mandatory field, if this field is not filled the resource can't be created;
- **Metadata** – Button that allows the user to define the metadata associated with the Resource. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Local File** – Physical location of the file that the resource represents;
- **Browse** – Button that allows the user to navigate through his file system to locate the target file. The form presented to the user is like the below picture. The user can navigate through the file system to choose the file that we want. By default if the Local File is not filled this window will open on the Working Folder. The user must choose a file from the list or cancel the operation, otherwise the user can't advance.

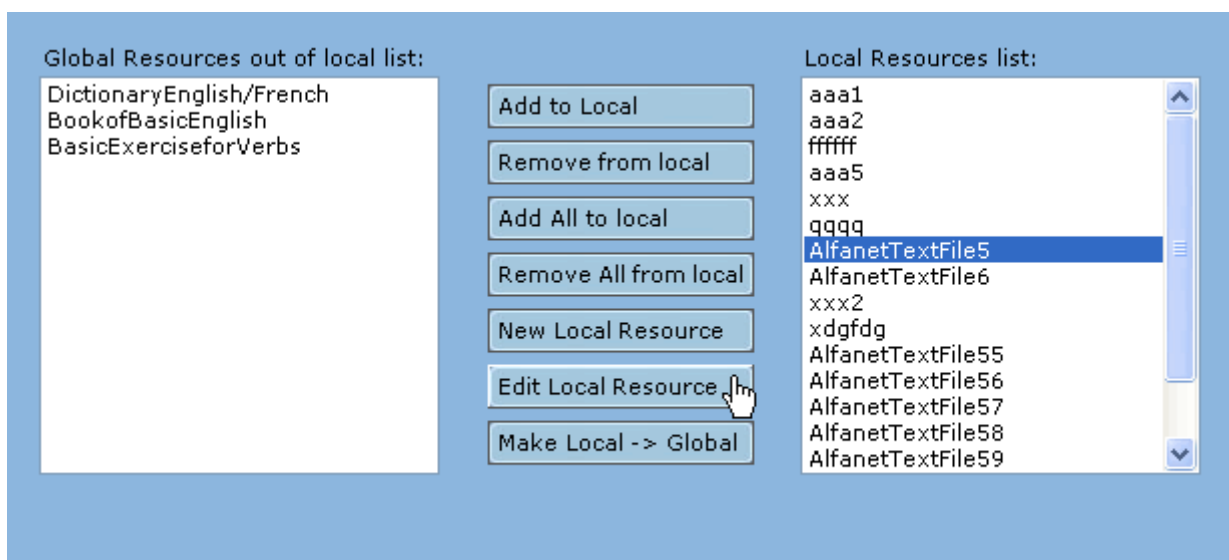


- **Open Chosen File** – Button that allows the user to see the chosen file with the default editor/reader application associated to that kind of file on Windows. If there is no application associated to the kind of the file this option will not work;
- **Resource Dependencies** – This is a list of the existing local resources and the user can choose other resources that this new resource is depending of. Only the selected resources will be considered as dependencies to the new resource.

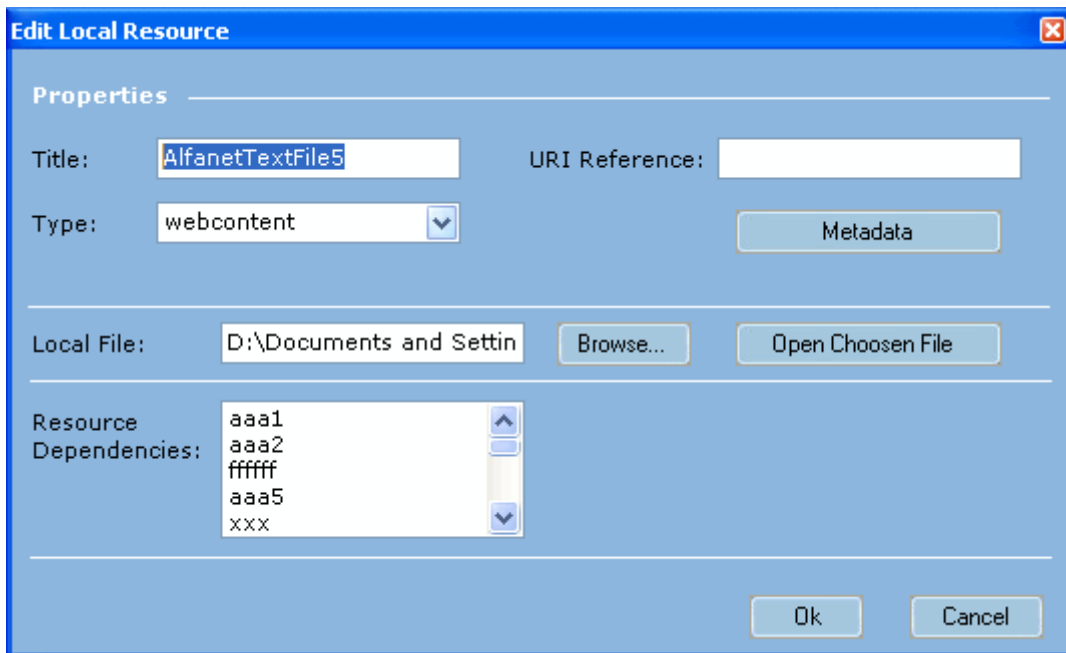
#### 4.4.6 Edit Local Resource

In this operation the user can edit an existing Local Resource on this Course.

To perform this operation the user only as to choose one resource from the Local Resources list (right list) and click on the “Edit Local Resource” button like the following picture represents:



After this step will be presented to the user a form similar to the “New Local Resource” option form but will appear filled with the resource data.



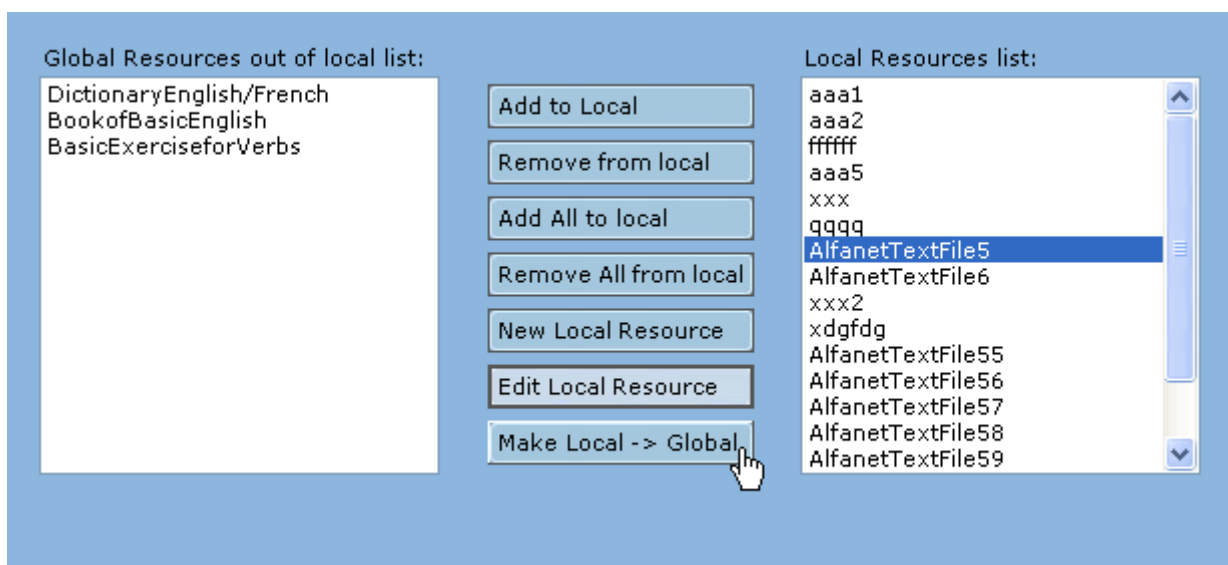
**Note:** If the selected resource is a Global Resource used in the Course, the changes made to the resource will only affect locally the instance inside the Course.

By other words if you change a resource, that belongs to the Global Resources, in this option, the changes will not make effect in the Global Resources file. So if you want to reuse that resource in other course, the resource copied to that new course will not be identical to the resource with the changes of this option.

#### 4.4.7 Make Local → Global

In this operation the user can transform a “really” Local Resource to a Global Resource adding it to the Global Resources file.

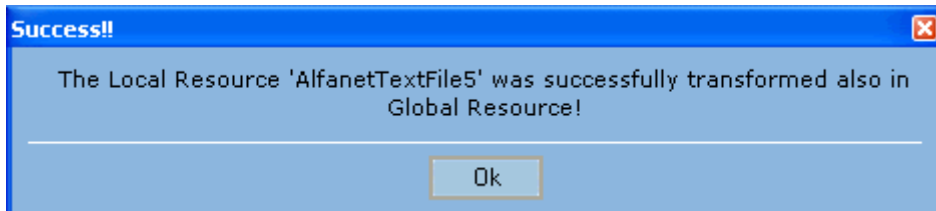
To perform this operation the user only as to choose one of the Local Resources from the right list and click on the “Make Local → Global” button like the following picture represents:



If the selected file is already a Global Resource the tool will send a message to the user, like the picture below, saying that this is already Global Resource and will do nothing.



Otherwise if the resource is not a Global Resource and there was no problem in the transformation the tool will send the following message of success.



## 4.5 Title & Metadata

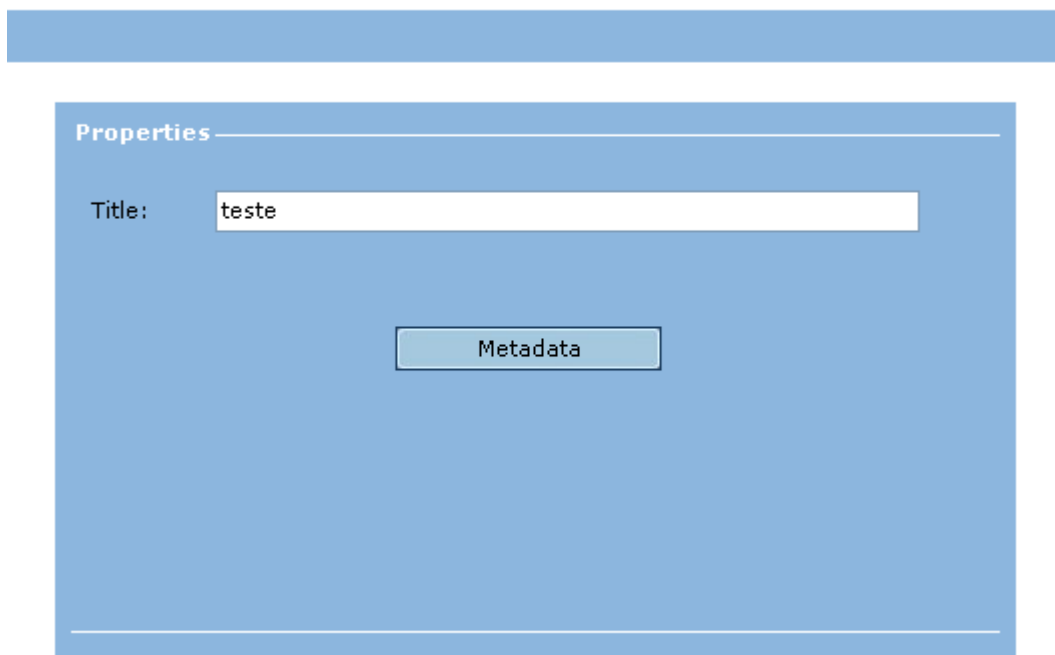
The Title & Metadata sub-module is the third option of the Learning Editor menu.



The Title & Metadata sub-module is the option that allows the user to edit the General Name of the Course (not the name of the file) and also the General Metadata of the Course.

The below picture shows how is the form available to the user:

### Learning Editor: Course Title & Course Metadata

A screenshot of a web form titled "Learning Editor: Course Title & Course Metadata". The form has a blue header bar. Below the header, there is a section titled "Properties" with a horizontal line underneath. Inside this section, there is a label "Title:" followed by a text input field containing the word "teste". Below the input field, there is a button labeled "Metadata".

- **Title** – This is the general title of the course;
- **Metadata** – Button that allows the user to define the general metadata associated with the Course. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

## 4.6 Learning Objectives

The Learning Objectives sub-module is the fourth option of the Learning Editor menu.



The Learning Objectives sub-module is the option that allows the user to edit the learning objectives of the course (unit of learning and course are the same concept).

In the *IMS-LD* specification we can find the following description about what are the Learning Objectives:

*“The learning objectives are the overall learning objectives to be attained by learners who complete the unit of learning. Learning objectives can be specified on several levels of detail. In IMS Learning Design, designers can choose to specify learning objectives at two levels, each with advantages and disadvantages. First, it is possible to define the learning objectives at the global level of the unit of learning. Second, it is possible to specify learning objectives for every single activity in the learning design. Designers can follow several approaches:*

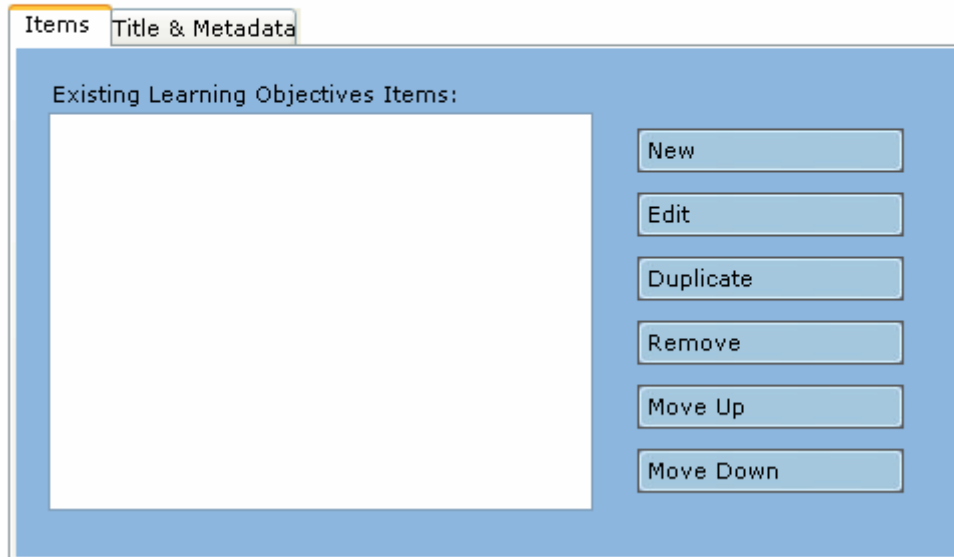
- *define the learning objectives only at the level of the unit of learning as a whole, not indicating the sub-objectives of the individual learning activities or what they add to the overall objectives.*
- *define the learning objectives only per learning-activity and not globally for the unit of learning. The learning objective for the unit of learning is nothing more or less than the list of all the learning objectives specified in the different learning activities.*
- *define the learning objectives on both levels: the learning objectives at the unit of learning level can be described more abstractly than those at the activity level.*

(...)

*Learning objectives and prerequisites can be described using the IMS Reusable Definition of Competency or Educational Objective (RDCEO) format, but can also refer to simple resources (e.g., a text) with a description of the learning objective.”*

The below picture shows how is the form available to the user:

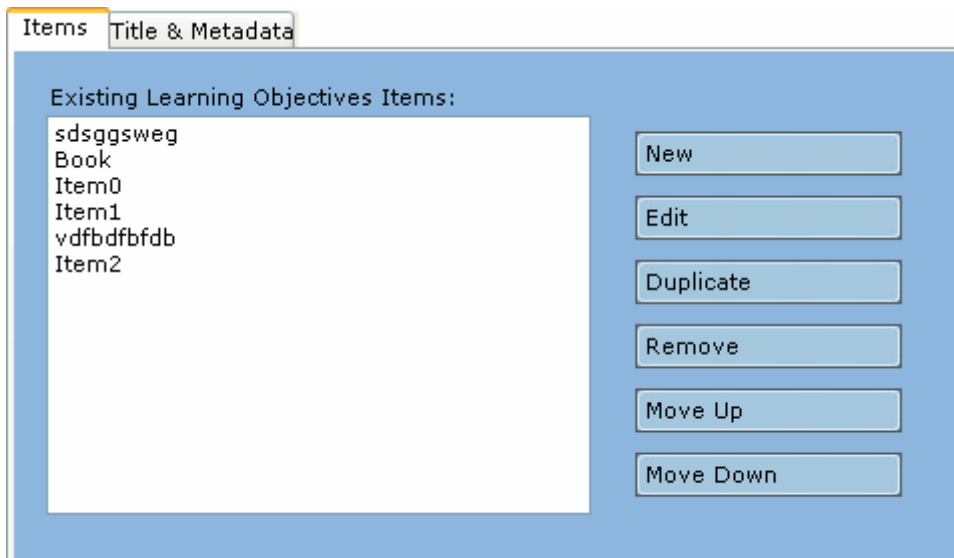
## Learning Editor: Learning Objectives Manager



In the Learning Objectives sub-module we have two sections of data available to the user fill. The first section is called of “Items” and the second section of “Title & Metadata”.

### 4.6.1 Items

The items are the objects that related to the resources and represents, in this case, the contents of the Learning Objectives.



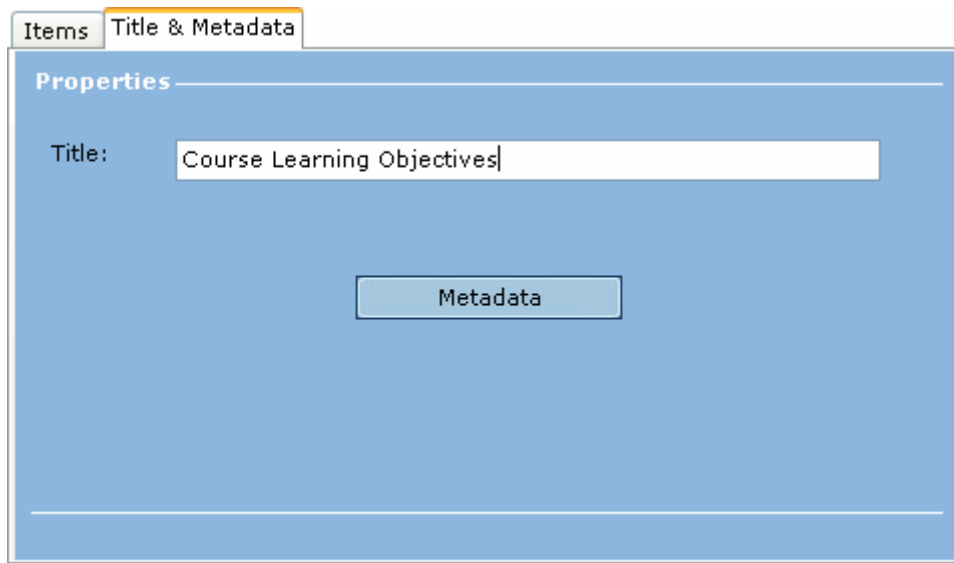
**Note:** For more information about the Items concept please read the Items section of the Learning Editor sub-modules section.

In this section the user can:

- Create **New** Items;
- **Edit** existing Items;
- **Duplicate** existing Items;
- **Remove** existing Items;
- **Move Up** existing Items;
- **Move Down** existing items.

## 4.6.2 Title & Metadata

The Learning Objectives can have a specific Title and Metadata so, the tool has a second section in this sub-module dedicated to those subjects that is called “Title & Metadata”.



The screenshot shows a software interface with two tabs: 'Items' and 'Title & Metadata'. The 'Title & Metadata' tab is selected and highlighted. Below the tabs is a blue panel with the heading 'Properties'. Under 'Properties', there is a 'Title:' label followed by a text input field containing the text 'Course Learning Objectives'. Below the input field is a button labeled 'Metadata'.

In this section the fields are:

- **Title** – This is the Title of the general Learning Objectives of the Course;
- **Metadata** – Button that allows the user to define the Metadata associated with the general Learning Objectives of the Course. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;



## 4.7 Pre-Requisites

The Pre-Requisites sub-module is the fifth option of the Learning Editor menu.



The Pre-Requisites sub-module is the option that allows the user to edit the requisites that the learners need to achieve before starting the course (unit of learning and course are the same concept).

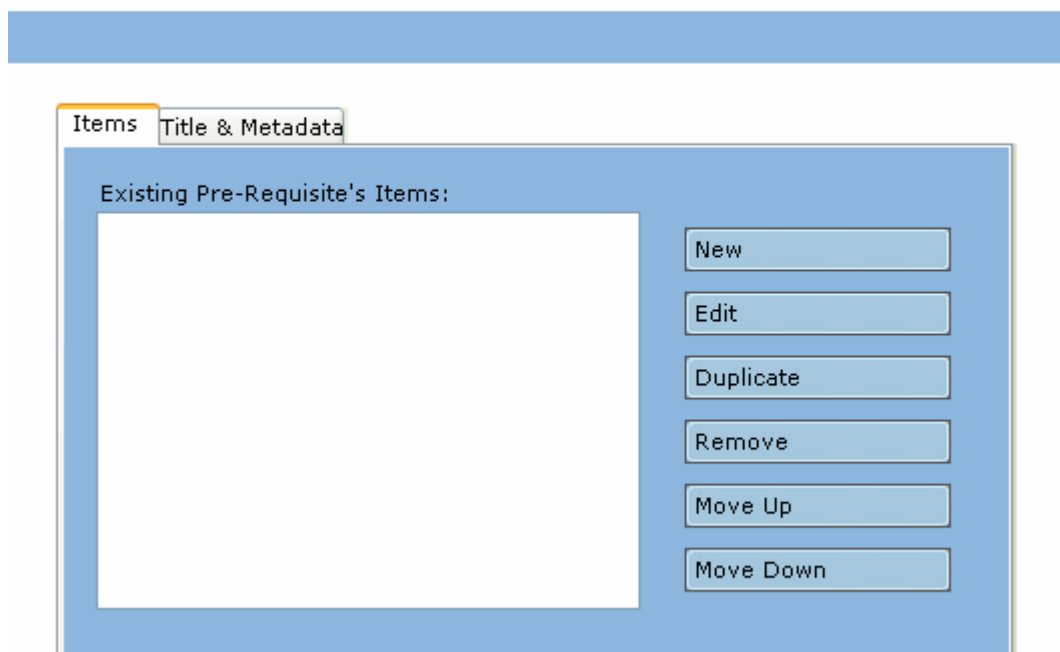
In the *IMS-LD* specification we can find the following description about what are the Pre-Requisites:

*"The prerequisites specify the overall entry requirements for learners for doing the unit of learning. As with learning objectives, the prerequisites can be provided at the level of the unit of learning and/or for individual learning activities."*

*Learning objectives and prerequisites can be described using the IMS Reusable Definition of Competency or Educational Objective (RDCEO) format, but can also refer to simple resources (e.g., a text) with a description of the learning objective."*

The below picture shows how is the form available to the user:

### Learning Editor: Pre-Requisites Manager

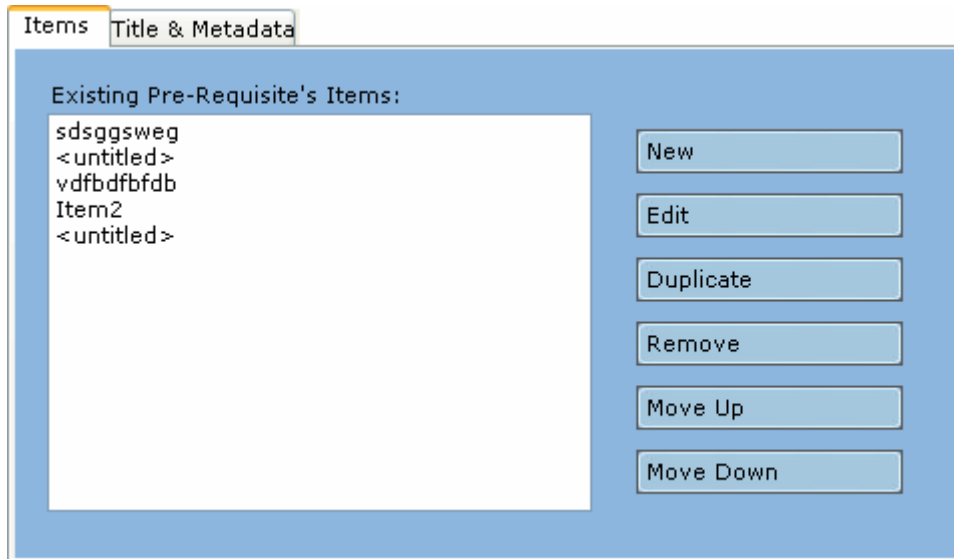
The screenshot shows the "Pre-Requisites Manager" interface. At the top, there is a blue header bar. Below it, there are two tabs: "Items" (which is active) and "Title & Metadata". The main content area is titled "Existing Pre-Requisite's Items:" and contains a large empty white box. To the right of this box is a vertical stack of six buttons: "New", "Edit", "Duplicate", "Remove", "Move Up", and "Move Down".

In the Pre-Requisites sub-module we have two sections of data available to the user fill.

The first section is the section is called of "Items" and the second section of "Title & Metadata".

### 4.7.1 Items

The items are the objects that related to the resources and represents, in this case, the contents of the Pre-Requisites.



**Note:** For more information about the Items concept please read the Items section of the Learning Editor sub-modules section.

In this section the user can:

- Create **New** Items;
- **Edit** existing Items;
- **Duplicate** existing Items;
- **Remove** existing Items;
- **Move Up** existing Items;
- **Move Down** existing items.

### 4.7.2 Title & Metadata

The Learning Objectives can have a specific Title and Metadata so, the tool has a second section in this sub-module dedicated to those subjects that is called "Title & Metadata".

The image shows a software interface with two tabs: 'Items' and 'Title & Metadata'. The 'Title & Metadata' tab is active. Below the tabs is a blue panel with the heading 'Properties'. Inside this panel, there is a 'Title:' label followed by a text input field containing the text 'Course Pre-Requisites'. Below the input field is a button labeled 'Metadata'. The interface is clean and uses a light blue color scheme.

In this section the fields are:

- **Title** – This is the Title of the general Pre-Requisites of the Course;
- **Metadata** – Button that allows the user to define the Metadata associated with the general Pre-Requisites of the Course. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

## 4.8 Components

The Components sub-module can be seen as a group of concepts that will be used and reused in the organization of the course.

In the Components option we can find concepts such as Roles, Activities and Environments.

In the *IMS-LD* specification we can find the following description about Components:

*“These are the declarations of the different components that provide the ‘building blocks’ for the method section of the learning design. In Learning Design Level A these are: roles, activities, and environments In Learning Design Levels B and C these are: roles, properties, activities, and environments. The components are declared separately from the method to avoid duplication in the method when using the same component more than once. The component and the method sections can be compared to a cooking recipe: the components are the list of ingredients and the method is the preparation instructions.”*

The Components sub-module is the sixth option of the Learning Editor menu:



The Components sub-module is like the following picture.

**Learning Editor: Components Manager**

« Main Menu

**Learning Editor**

- Roles
- Environments
- Activities
- Properties

Welcome to the Components Manager.

In this option you are able to manipulate objects of the concepts that will be used and reused in the organization of the course. The concepts are the following ones:

- Roles - Indicates every kind of player that will interact with the course. We have two kinds of Roles: "Learner" and "Staff". The Learner Roles define the different types of learners that the course will have. The Staff Roles define every type of support players needed to the execution of the course.
- Environments - Indicates the support Environments that are needed to execute the activities of the course. Inside of the Environments we can define or configure Services or Learning Objects.
- Services - The Service can be a Conference between several course players, Send a Mail to a specific kind of player or create an Index/Search Service.
- Learning Objects - The Learning Objects can be any kind of Item or Resource that can help and support the Learner in the course.
- Activities - We can define three kinds of activities in the course, Learning Activities, Support

When the user clicks in the “Components” button the tool shows a screen with a brief text to help and explain to the user every concepts that belongs to the Components sub-module.

The tool also changes the Learning Editor menu to the Components menu that is like the below picture.



In this menu we can:

- Accede to the tool Main Menu by the “<< Main Menu” link;
- Go back to the Learning Editor menu by the “Learning Editor” button;
- Accede to the concepts that are involved on the Components clicking on the related buttons (Roles, Environments, Activities and Properties).

**Note:** *The Properties concept belongs to the Level B of the IMS-LD specification so the target of this link will be inactive because this tool version supports only Level A of the IMS-LD specification.*

The following sections will explain in more detail the concepts that belong to the Level A of the Components (Roles, Environments and Activities).

## 4.9 Roles

The Roles sub-module is the second option of the Components menu.



In the *IMS-LD* specification we can find the following description about Roles:

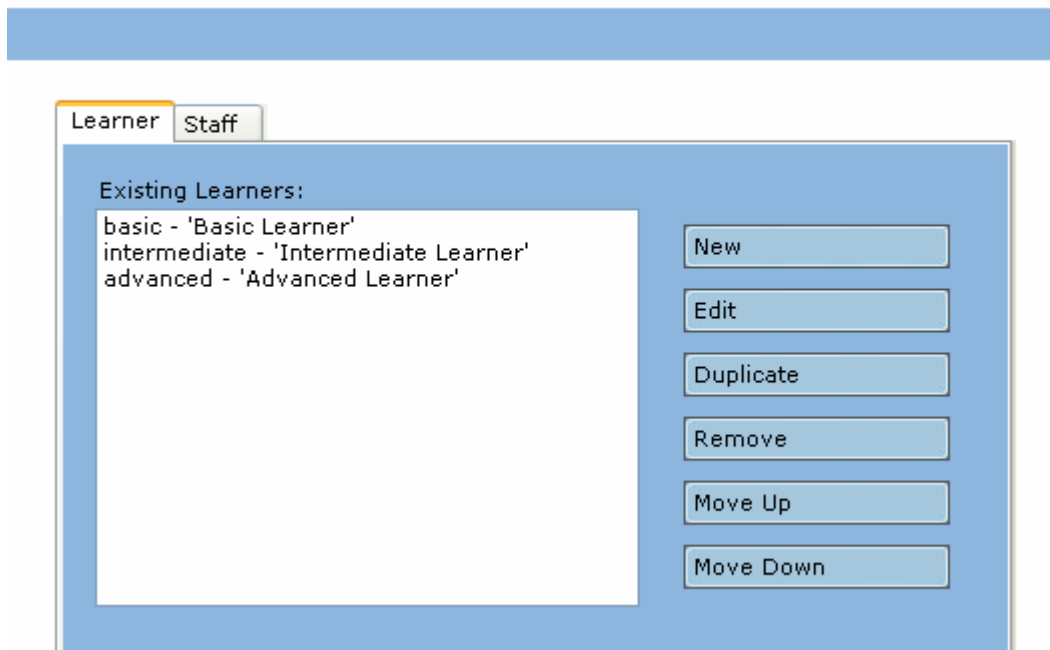
*“Roles allow the type of participant in a unit of learning to be specified. There are two basic Role types: Learner and Staff. These however can be sub-typed to allow learners to play different roles in certain types of learning activity such as task-based, role-play and simulations. Similarly support staff can be sub-typed and given more specialized roles, such as Tutor, Teaching Assistant, Mentor, etc. Roles thus lay the basis for multi-user models of learning.*

*The name a role is given depends on the pedagogy and setting used. In some instances a learner is called a ‘student’ in others a ‘participants’. The names of staff roles are even more variant, e.g., teacher, trainer, tutor, facilitator, mentor, assessor. Every role can have its own ‘title’ which provides the name for it.*

*At runtime more than one user can be assigned to the same role, however restrictions can be set on the maximum and minimum number for each role. In this sense roles can be used for grouping purposes.”*

The below picture shows how is the form available to the user:

### Learning Editor: Roles Manager

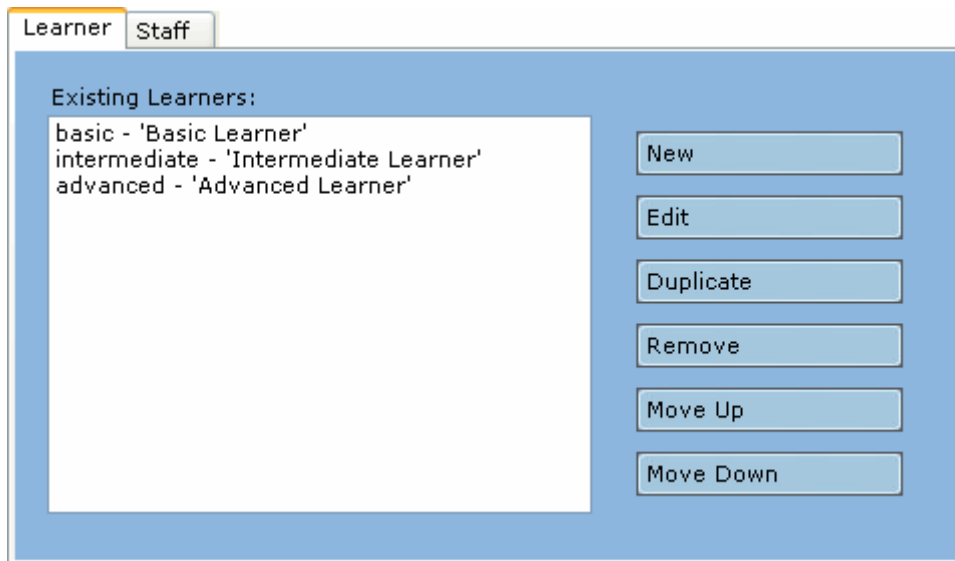


In the Roles sub-module we have two kinds of Roles. Those kinds are Learner and Staff.

## 4.9.1 Learner Roles

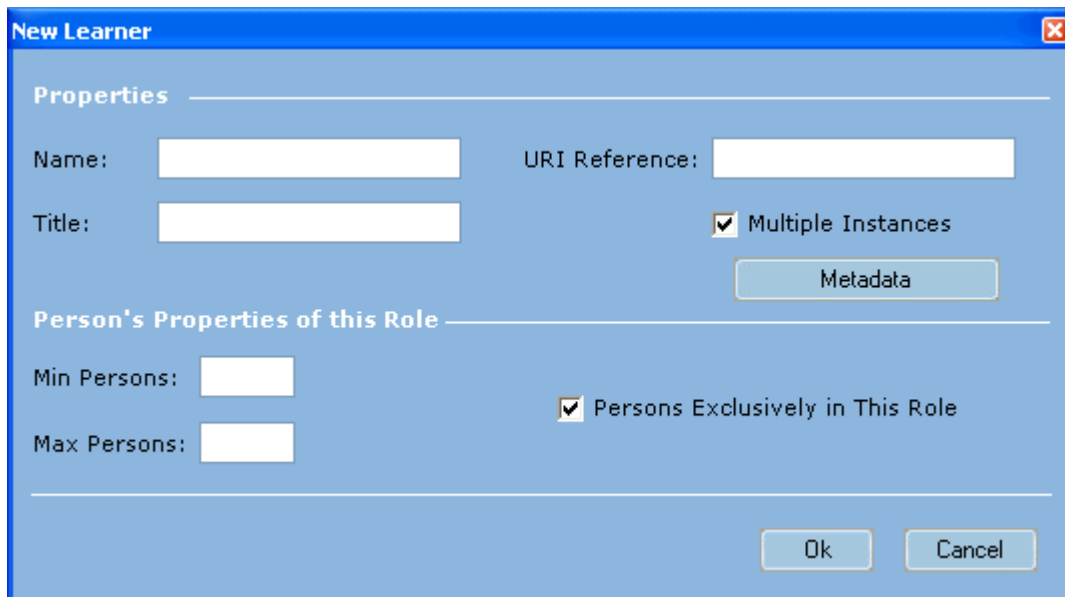
The Learner Roles can be seen as the Roles of the target persons of the course. They can be defined as student, participant, learner, etc...

The section that allows to manipulate the Learner Roles is like the below picture.



This feature allows the user to:

- Create **New** Learner Roles – In this option the tool will present to the user a form, like the one below, to let the user fill the new Learner Role data.



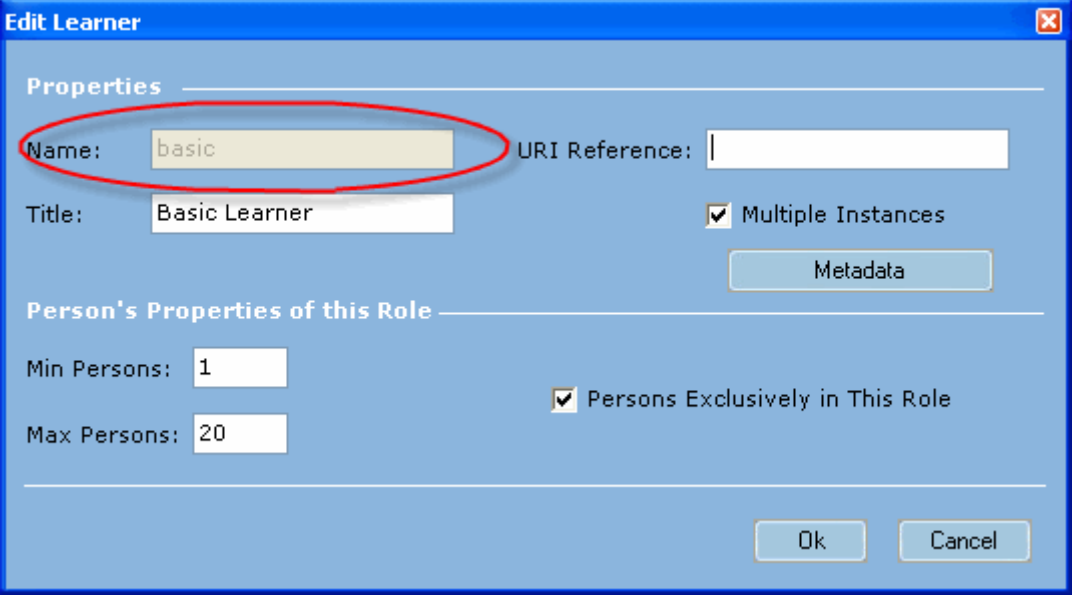
The form consists in the following items:

- **Name** – Text value that represents the ID of the Role. This is a mandatory field, if this field is not filled the resource can't be created;

**Note:** *This is a very important value because this is the field that will identify the Learner Role in every elements and lists. Try to put an explicit value on the Name of the Role to let you easily remember and identify the different Roles;*

- **URI Reference** – Text value that allows the user to refer an URI link to this role;

- **Title** – Text value that defines a title description for the Role;
  - **Multiple Instances** – Option value that defines if this role can have multiple instances or can only have one and only one instance;
  - **Metadata** – Button that allows the user to define the Metadata associated with this Role of the Course. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
  - **Min Persons** – Numeric value that defines the minimum number of persons in that role before start running the course;
  - **Max Persons** – Numeric value that defines the maximum number of persons in that role before start running the course. If these two values are empty, the role can be considered as unlimited;
  - **Persons Exclusively in This Role** – Option value that defines if the persons in this Role should be only in this Role or can belong to more than one Role. This is due to the use of sub-roles hierarchies.
- **Edit** existing Learner Roles – In this option the tool will present to the user a form similar to the previous option but with the chosen Learner Role data filled. The user can change all fields of that Learner Role except to the Name value, as you can see in the below picture;



The screenshot shows a dialog box titled "Edit Learner" with a blue header and a close button in the top right corner. The dialog is divided into two main sections: "Properties" and "Person's Properties of this Role".

**Properties section:**

- Name:** A text input field containing "basic", which is circled in red.
- URI Reference:** An empty text input field.
- Title:** A text input field containing "Basic Learner".
- Multiple Instances:** A checked checkbox.
- Metadata:** A button.

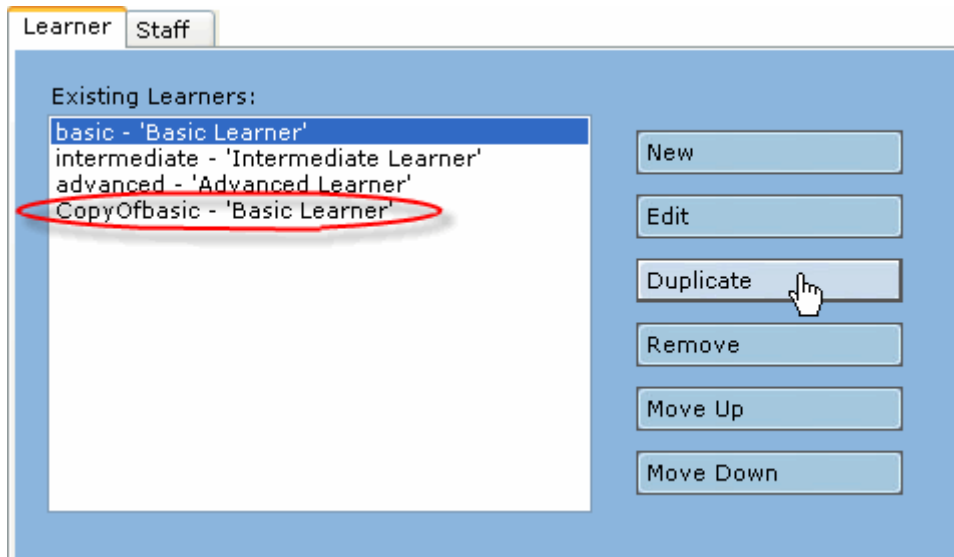
**Person's Properties of this Role section:**

- Min Persons:** A text input field containing "1".
- Max Persons:** A text input field containing "20".
- Persons Exclusively in This Role:** A checked checkbox.

At the bottom of the dialog, there are two buttons: "Ok" and "Cancel".

- **Duplicate** existing Learner Roles – In this option the user can select an existing Learner Role and create a copy of that Role. The only different value will be the Name of the Role, like you can see in the below picture;



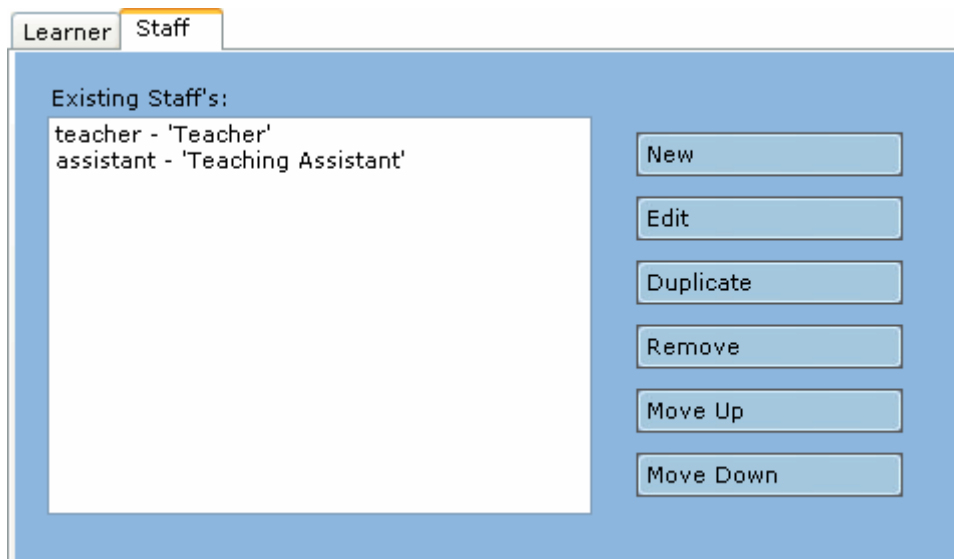


- **Remove** existing Learner Roles – In this option the user can select and delete one existing Learner Role;
- **Move Up** existing Learner Roles – In this option the user can select and move up one position the selected Learner Role in the list order;
- **Move Down** existing Learner Roles – In this option the user can select and move down one position the selected Learner Role in the list order.

## 4.9.2 Staff Roles

The Staff Roles can be seen as the Roles of the persons needed to support the execution of the course. They can be defined as teacher, trainer, tutor, facilitator, mentor, assessor, etc...

The section that allows to manipulate the Staff Roles is like the below picture.



This feature allows the user to:

- Create **New** Staff Roles – In this option the tool will present to the user a form, like the one below, to let the user fill the new Staff Role data.

The form consists in the following items:

- **Name** – Text value that represents the ID of the Role. This is a mandatory field, if this field is not filled the resource can't be created;

**Note: This is a very important value because this is the field that will identify the Staff Role in every elements and lists. Try to put an explicit value on the Name of the Role to let you easily remember and identify the different Roles;**

- **URI Reference** – Text value that allows the user to refer an URI link to this role;
  - **Title** – Text value that defines a title description for the Role;
  - **Multiple Instances** – Option value that defines if this role can have multiple instances or can only have one and only one instance;
  - **Metadata** – Button that allows the user to define the Metadata associated with this Role of the Course. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
  - **Min Persons** – Numeric value that defines the minimum number of persons in that role before start running the course;
  - **Max Persons** – Numeric value that defines the maximum number of persons in that role before start running the course. If these two values are empty the role can be considered as unlimited;
  - **Persons Exclusively in This Role** – Option value that defines if the persons in this Role should be only in this Role or can belong to more than one Role. This is due to the use of sub-roles hierarchies.
- **Edit** existing Staff Roles – In this option the tool will present to the user a form similar to the previous option but with the chosen Staff Role data filled. The user can change all fields of that Staff Role except to the Name value, as you can see in the below picture;

**Edit Staff**

**Properties**

Name:  URI Reference:

Title:   Multiple Instances

**Person's Properties of this Role**

Min Persons:   Persons Exclusively in This Role

Max Persons:

- **Duplicate** existing Learner Roles – In this option the user can select an existing Staff Role and create a copy of that Role. The only different value will be the Name of the Role, like you can see in the below picture;

**Learner** **Staff**

Existing Staff's:

- teacher - 'Teacher'
- assistant - 'Teaching Assistant'
- CopyOfteacher - 'Teacher'

- **Remove** existing Staff Roles – In this option the user can select and delete one existing Staff Role;
- **Move Up** existing Staff Roles – In this option the user can select and move up one position the selected Staff Role in the list order;
- **Move Down** existing Staff Roles – In this option the user can select and move down one position the selected Staff Role in the list order.

## 4.10 Environments

The Environment concept can be seen as a definition of objects needed by the activities to run properly. The Environments consist in two kinds of objects with different purposes.

Those objects can be Services or Learning Objects.

The Services are separated in several concepts such as services that support the establishment of conferences between the course Roles (Conference Service), the sending of e-mails to the course Roles (Send Mail Service), or to define index searching system inside the course (Index/Search Service).

The Learning Objects can be seen as objects or collection of objects to help the end user to execute and achieve the activity learning objectives. We can consider a good example for the learning objects a collection of electronic books or an electronic library.

In the *IMS-LD* specification we can find the following description about Environments:

*“Activities take place in a so-called ‘environment’, which is a structured collection of learning objects, services, and sub-environments. The relationship between an activity and an environment can be derived from the linguistic description of the activities. Most nouns in the activity imply the availability of learning objects in the environment; references to other persons imply the availability of communication services; some verbs imply the availability of supportive services or tools. For instance the activity: ‘read the problem and discuss solutions with your peers’ refers to environment components: ‘the problem’ which must be available for reading; and ‘peers’ which must be available to communicate with (including communication means).”*

The environments sub-module is the third option in the Components menu.



When the user clicks in this option the Components menu is changed to the Environments menu. This menu is like the following picture.

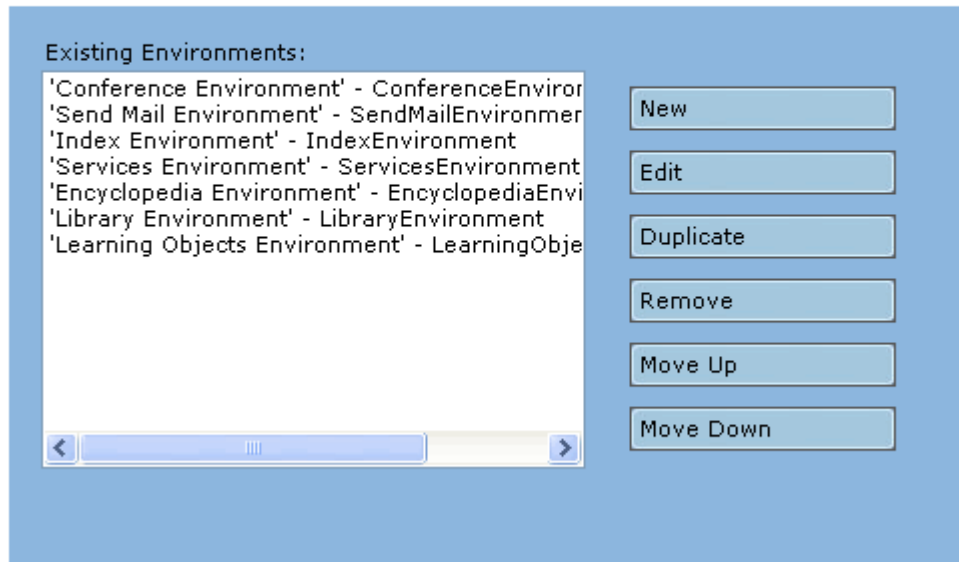


In this menu the user finds:

- A button to return to the Components menu and page (“Back to Components” button);
- A button to create and edit the environments (“Environments” button);
- A button to create and edit Services (“Services” button);
- A button to create and edit Learning Objects (“Learning Objects” button).

The Environments form available to the user is like the following picture.

## Learning Editor: Environments Manager



In this feature the user can:

- Create **New** Environments – In this option the tool will present to the user a form, like the one below, to let the user fill the new Environment data.

The 'New Environment' dialog box has a title bar with a close button. It features two tabs: 'Properties' and 'Content'. The 'Content' tab is active, showing a 'Title:' text box and a 'Type Of Environment Content:' section with three radio button options: 'Service(s)' (selected), 'Reference to Other Environment(s)', and 'Learning Object(s)'. A 'Metadata' button is located below the content area. At the bottom right are 'Ok' and 'Cancel' buttons.

The form consists in the following items:

- **Title** – Represents the title that the user will give to the Environment. This is a mandatory field, if this field is not filled the Environment can't be created;

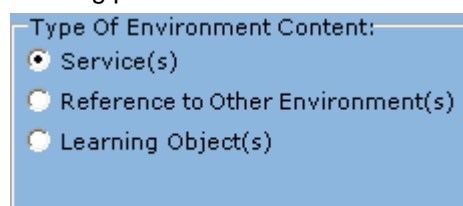
**Note:** Despite this value is not the ID of the Environment (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing environment or other kind of element. **According to this, try to put an explicit value on the title of the environment to let you easily remember and identify the different environments;**

- **Metadata** – Button that allows the user to define the Metadata associated with this Environment. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

- **Type of Environment Content** – Option value that defines which kind of content this Environment uses.

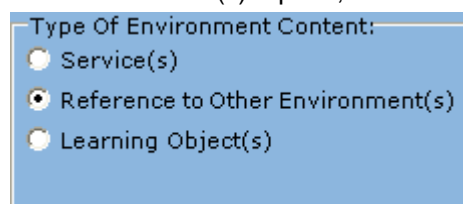
The Environment can have Services, Learning Objects or other Environments (sub-environments).

If the user decides to include services in the environment he must choose the “Service(s)” option, like the following picture shows.



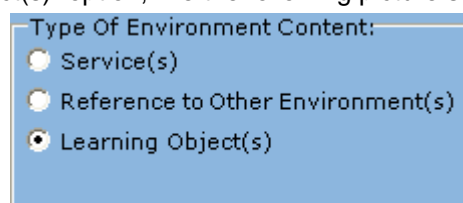
In the Content section, when this option is selected, will appear the list of every Services defined in this Course.

If the user decides to include sub-environments in the environment he must choose the “Reference to Other Environment(s)” option, like the following picture shows.



In the Content section, when this option is selected, will appear the list of every Environments defined in this Course.

If the user decides to include learning objects in the environment he must choose the “Learning Object(s)” option, like the following picture shows.



In the Content section, when this option is selected, will appear the list of every Learning Objects defined in this Course.

- **Content** – Section that has the related list according to the choice in the “Type Of Environment Content” field.

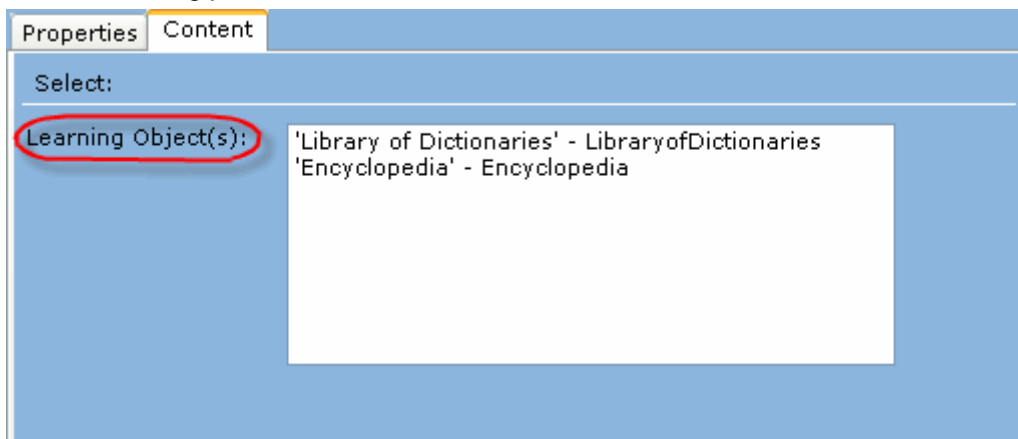
If the user select to use Services in this Environment the Content list will have the list of every Services defined in the Course and will appear like the following picture.



If the user select to use References to Other Environments in this Environment the Content list will have the list of every Environments defined in the Course and will appear like the following picture.



At last, if the user select to use Learning Objects in this Environment the Content list will have the list of every Learning Objects defined in the Course and will appear like the following picture.

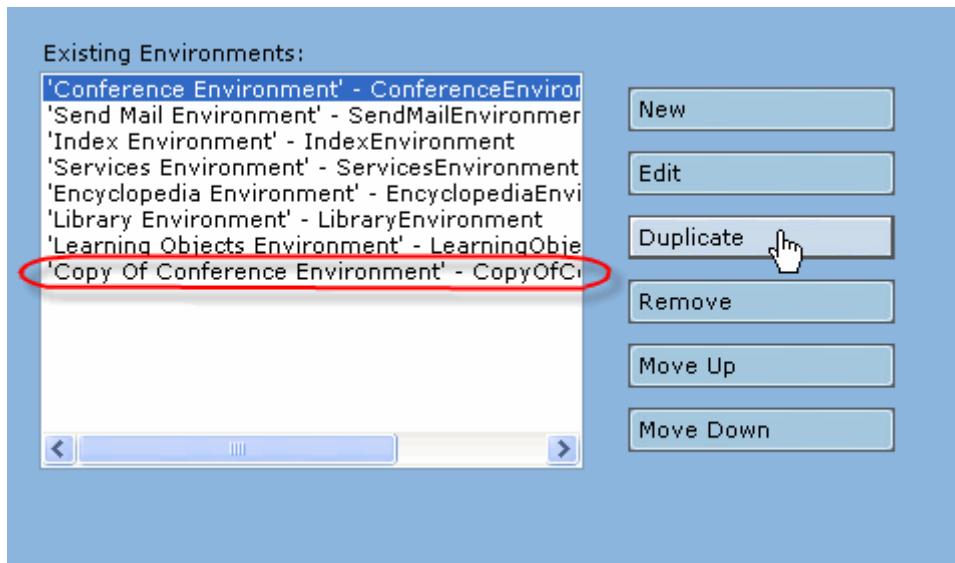


The Content list is a mandatory field. This means that, whatever option the user chooses in the "Type Of Environment Content" field, he must select at least one element of the Content list.

If there isn't any element on the Content list, the user must choose another type of content that has defined elements or cancel the operation

**Note:** Because the Content list is a mandatory field, if there isn't any defined Service, Learning Object or Environment it's impossible to create an Environment. **According to this, it's recommended that the user define in a first moment the Services and/or Learning Objects and only after that step define the Environments of the course.**

- **Edit** existing Environments – In this option the tool will present to the user a form similar to the previous option but with the chosen Environment data filled. The user can change all fields of that Environment;
- **Duplicate** existing Environments – In this option the user can select an existing Environment and create a copy of that Environment. The only different values will be the ID and Title of the Environment, like you can see in the below picture;

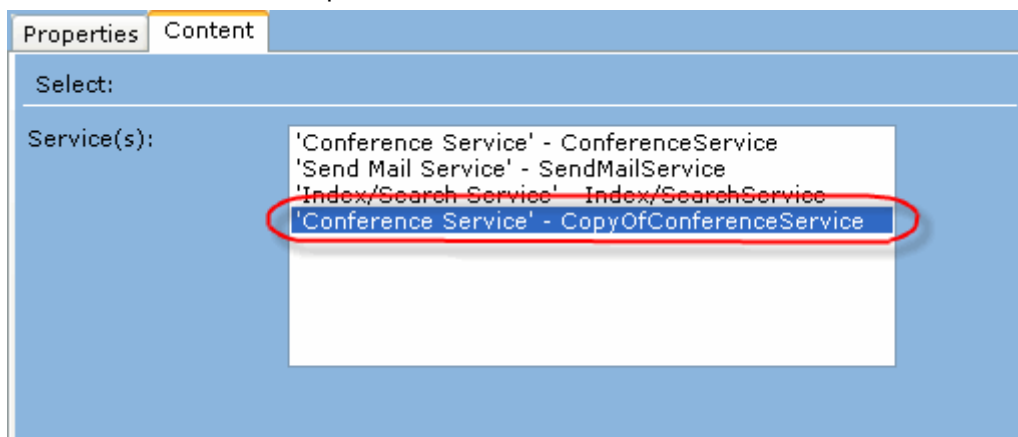


When the user duplicates an Environment all its content will be copied to.

In other words if the user duplicates an Environment that has Services or Learning Objects, those objects will be copied too.

You can see an example on the following picture.

In the above picture we copied an Environment that has in the content a Conference Service. If we go to the Services list we will find a new service (copy of the original). This new copied service will be the content of the copied environment.



**Note:** This fact doesn't happen if the user selects sub-environments to the content of the Environment.

- **Remove** existing Environments – In this option the user can select and delete one existing Environment;



- **Move Up** existing Environments – In this option the user can select and move up one position the selected Environment in the list order;
- **Move Down** existing Environments – In this option the user can select and move down one position the selected Environment in the list order.

### 4.10.1 Services

The Services are Environments that specify and define facilities to configure three specific kinds of environments to run the activities.

In the *IMS-LD* specification we can find the following description about Services:

*“Besides resources which can be defined at design time, there are numerous so-called ‘service facilities’ used during the teaching and learning, for instance, a discussion forum or some other communication facility. Service facilities are resources that cannot be given a URL at design time. They have to be instantiated by a local runtime service. This is because, if a service facility is bound at design time, then that specific service would have to be used by all users of all instances of the learning design. When what is needed is an instance of the service that is unique to the runtime instance of the learning design and its assigned users, (e.g., if a chat forum is to be dedicated to the use of a specific group of learners and support staff associated with an particular instance of a learning design), then this has to be created and the local URL assigned after the instance of the design has been set up and the group of learners and staff associated with it. For this to work, it requires a well defined set of service types, which are known to the runtime service, such as chat, discussion forum, announcement channel etc. These are now commonly found in learning management systems. In a learning design, the use and setup of such a service is declared at an abstract level, so that a runtime facility (or a human) can setup the necessary facility according to the requirements. In the learning design specification, the abstract declaration of a service facility is called a ‘service’. The instantiation of a service is called a ‘service facility’.*

*Current service types are: send-mail, conference, monitor (level B), and index search. The selection of services to be included needs to be driven by the community. We therefore decided to start with the most widely implemented and used services in online learning environments.”*

- **Send-Mail Service**

About the Send-Mail Service we can find the following description in the *IMS-LD* specification:

*“One of the services in any online learning facility is the ability to send and receive mail. This is done with an e-mail client. However, in learning situations it is often needed to know all the e-mail addresses of your fellow students and teachers when sending to groups. This information is available in the runtime system. In order to help users to send e-mail to other users in the same run of the unit-of-learning, the declaration of a send-mail facility is included in the services part. When a send mail service is included as part of a user’s environment, the runtime system must provide a facility to edit a message and attachments and to send them to a selected list of e-mail addresses of users in the run of the unit-of-learning. This can either be to all the users in a specific role or to individuals selected from a role. The users receive the messages in their regular inbox in the e-mail client.”*

- **Conference Service**

About the Conference Service we can find the following description in the *IMS-LD* specification:

*“A typical communications service is a conference. The conference service, in addition to a title and metadata, specifies four conference system roles: participant, observer, conference-manager, and moderator. These contain references to roles in the learning design. When the learning design roles have been assigned players, this information can be used to automatically set up the dedicated conference space. It is not defined in this Learning Design Specification what permissions the conference roles should have, so this is left up to the implementer. However, these conference roles have commonly understood meanings and learning designers would have the expectation that implementers would stay within this range.*

*The conference service can be divided into three subtypes: synchronous conferences (like chats and audio/video conferences), asynchronous conferences (like newsgroups, forums), and announcements (one to many asynchronous conferences).”*

- **Index-Search Service**

About the Conference Service we can find the following description in the *IMS-LD* specification:

*“The Index-Search service enables a unit-of-learning to be indexed and searches to be then made across it. In addition to title and metadata elements, it includes an index element and a search element.*

*The index element is a wrapper for indexing aspects, used to set up a search service.*

The index is made in the background (not visible to users). The visibility is determined with the search element.

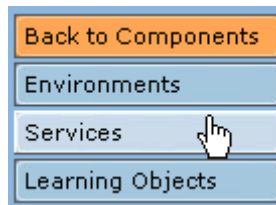
The functionality of the index is dependent on the search element:

- when search is free-text-search, then the index is made on the resource pointed at in the index (i.e., the underlying html texts).
- when search is index-with/without-reference, then an index is only made of the elements which share the same class, including underlying items. This has the form of a table of contents.

The search element specifies how a user can access the indexed entities. There are three possibilities:

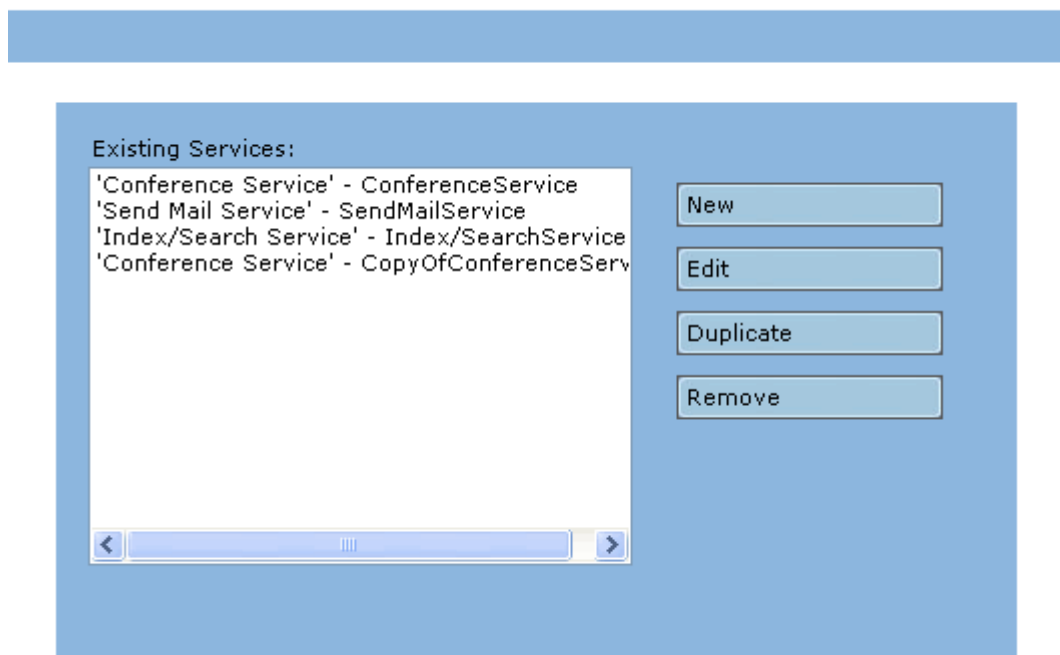
- the user gets a free text search dialog, where he can search the index in a free text format (this also means that the index has to be build for free text retrieval). The syntax for free text retrieval is implementation dependent (e.g., the format found in search engines like Google or AltaVista).
- the user is presented a text index (table of content) with (hyper-) linked (or on other media e.g., page numbers) references to the source.
- the user is presented a text index (table of content) without (hyper-) linked references. This provides e.g. information about the structure of the unit of learning.”

The Services sub-module is the third option of the Environments Menu.



The Services form available to the user is like the following picture.

## Learning Editor: Services Manager



In this feature the user can:

- Create **New Services** – In this option the tool will present to the user a form, like the one below, to let the user fill the new Service data.

The form consists in the following items:

- **Title** – Represents the title that the user will give to the Service. This is a mandatory field, if this field is not filled the Service can't be created;

**Note:** Despite this value is not the ID of the Service (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing service or other kind of element. **According to this, try to put an explicit value on the title of the service to let you easily remember and identify the different services;**

- **Parameters** – Text value to the parameters to be passed during runtime;
- **Class** – Text value to the class name assigned to this service;
- **Is Visible** – Option value to define if this service will be visible to the end user of the course;
- **Type Of Service** – Option value to define which kind of Service is this Service. According to this choice the content of the second section will change such as the section name. We can see three examples of those changes in the following pictures.

Properties **Conference**

Title:

Parameters:

Class:

Is Visible

Type Of Service:

Conference

Send mail

Index/Search

Monitor

Properties **Send Mail**

Title:

Parameters:

Class:

Is Visible

Type Of Service:

Conference

Send mail

Index/Search

Monitor

Properties **Index/Search**

Title:

Parameters:

Class:

Is Visible

Type Of Service:

Conference

Send mail

Index/Search

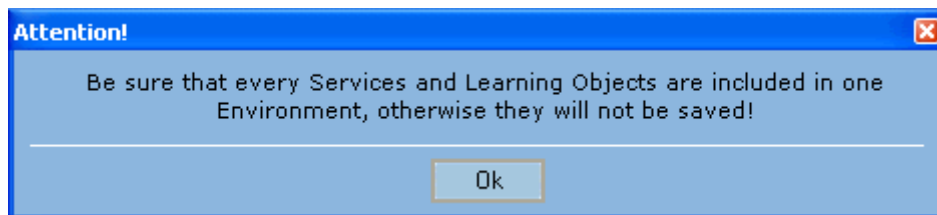
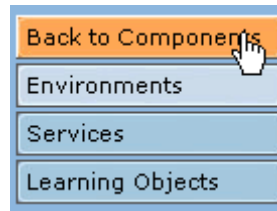
Monitor

- **Edit** existing Services – In this option the tool will present to the user a form similar to the previous option but with the chosen Service data filled. The user can change all fields of that Service;
- **Duplicate** existing Services – In this option the user can select an existing Service and create a copy of that Service. The only different values will be the ID and Title of the Service;
- **Remove** existing Services – In this option the user can select and delete one existing Service;

**Note: When you define a Service you need to associate it to an Environment otherwise that Service will not be saved!!!**

*This is due to the fact that Services and Learning Objects are inner concepts of the Environments, but the tool organizes the Services and the Learning Objects separately from the Environments to make the Environment's interface simpler.*

*According to this fact, when the user is leaving the Environments menu, the Authoring Tool launches an alert message to the user. The process is described in the below pictures.*



In the following sections we will explain the content of the second section of the Services form. This section is different according to the choice made by the user in the "Type Of Service" field.

#### 4.10.1.1 Conference

When the user chooses to create or edit a Conference Service the tool will present the second section of the Service form to the user like the below picture.

 A screenshot of the 'Conference' form. The form has two tabs: 'Properties' and 'Conference'. The 'Conference' tab is active. On the left, there is a 'Conference Type' section with three radio buttons: 'Synchronous' (selected), 'Asynchronous', and 'Announcement'. Below this are two buttons: 'Item' and 'Metadata'. On the right, there are four fields: 'Participant(s)', 'Observer(s)', 'Manager:', and 'Moderator:'. Each of these fields has a list of roles: 'teacher - 'Teacher'', 'assistant - 'Teaching Assistant'', and 'basic - 'Basic Learner''. The 'Participant(s)' and 'Observer(s)' fields have expandable arrows. The 'Manager:' and 'Moderator:' fields have dropdown arrows.

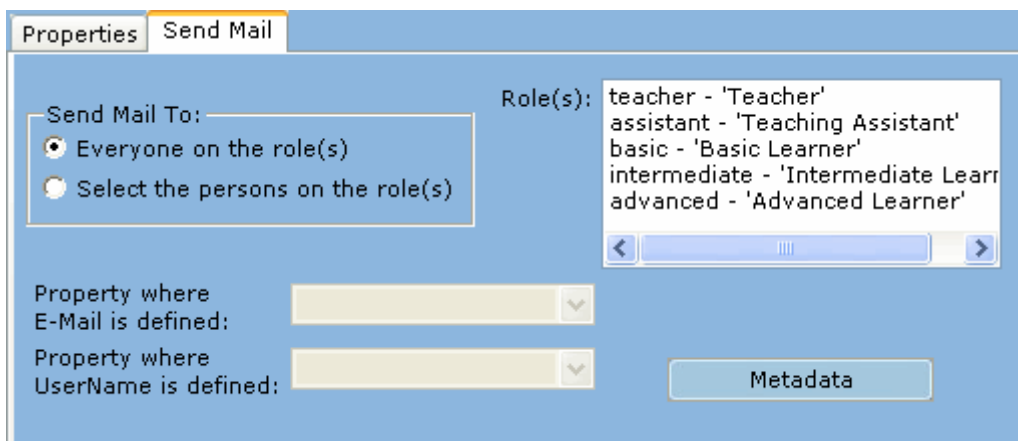
- **Conference Type** – Option value to define the type of conference this service will be. The options are "Synchronous", "Asynchronous" and "Announcement".  
 The "Synchronous" type of conference means that the participants can interact in a synchronous way, e.g. chat.  
 The "Asynchronous" type of conference means that the participants can interact in an asynchronous way, e.g. a lesson give by a tutor with possibility of questions by the learners.  
 The "Announcement" type of conference means that the participants have an interaction in a single way, e.g. a lesson give by a tutor without interaction by the learners.
- **Participant(s)** – List where the user will select the Roles that will participate in the conference.  
 The list is of every Roles defined in the Course.  
 The selected Roles will be the participants of the conference.

The user must select at least one role to advance because this field is a mandatory field;

- **Observer(s)** – List where the user will select the Roles that will be observers in the conference. The list is of every Roles defined in the Course. The selected Roles will be the observers of the conference;
- **Manager** – List where the user will select the Role that will manage the conference. The list is of every Roles defined in the Course. The selected Role will be the manager of the conference;
- **Moderator** – List where the user will select the Role that will moderate the conference. The list is of every Roles defined in the Course. The selected Role will be the moderator of the conference;
- **Item** – Button that let the user define an Item to associate this conference to a Resource. For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;
- **Metadata** – Button that allows the user to define the Metadata associated with this Conference Service. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

#### 4.10.1.2 Send Mail

When the user chooses to create or edit a Send Mail Service the tool will present the second section of the Service form to the user like the below picture.



- **Send Mail To** – Option value to define if the mail will be send to “Everyone on the role(s)” selected by the user or if the runtime system will allow to the end user of this service to “Select the persons on the role(s)” which will receive the mail related to this Service;
- **Role(s)** – List where the user will select the Roles that will receive the mail related to this Service. The list is of every Roles defined in the Course. The user must select at least one role to advance because this field is a mandatory field;
- **Metadata** – Button that allows the user to define the Metadata associated with this Send Mail Service. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

**Note:** Despite the “Property where E-Mail is defined” and the “Property where Username is defined” fields are visible they are inactive. This is due to the fact that the properties belong exclusively to the Level B of the IMS-LD specification.

#### 4.10.1.3 Index/Search

When the user chooses to create or edit a Index/Search Service the tool will present the second section of the Service form to the user like the below picture.

- **Search Type** – Option value to define the type of search that this Service allows. The options are “Free Text”, “Index with reference” and “Index without reference”.

The “Free Text” type means that the search uses a free text search index.

The “Index with reference” type means that the search uses page numbers or hyperlinks (depending on the publication medium used).

The “Index without reference” type means that the search is a list of terms without page numbers or hyperlinks.
- **Class to Index** – Text value to the user identifies which class defined in the course will the index of the search be make on.

If this option is selected, the user must fill this field because, this field turns mandatory when is selected;
- **Element to Index** – List where the user will select the ID of one of the elements defined in this course which the index of the search is making on.

The list is of every objects with an ID defined inside the Course.

If this option is selected, the user must fill this field because, this field turns mandatory when is selected;
- **Type of Element to Index** – List where the user will select the type of elements defined in this course which the index of the search is make on.

The list is of every concepts related to the *IMS-LD* standard.

If this option is selected, the user must fill this field because, this field turns mandatory when is selected;



## 4.10.2 Learning Objects

The Learning Objects are Environments that define objects or collections of objects that can be associated with specific activities or acts to help the learner to achieve the learning objectives of that activity and the entire course.

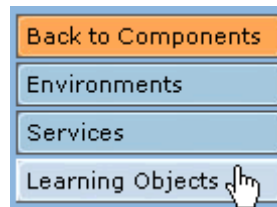
In the *IMS-LD* specification we can find the following description about Learning Objects:

*“Learning objects are defined here, as any reproducible and addressable digital or non-digital resource used to perform learning activities or support activities.*

*(...)*

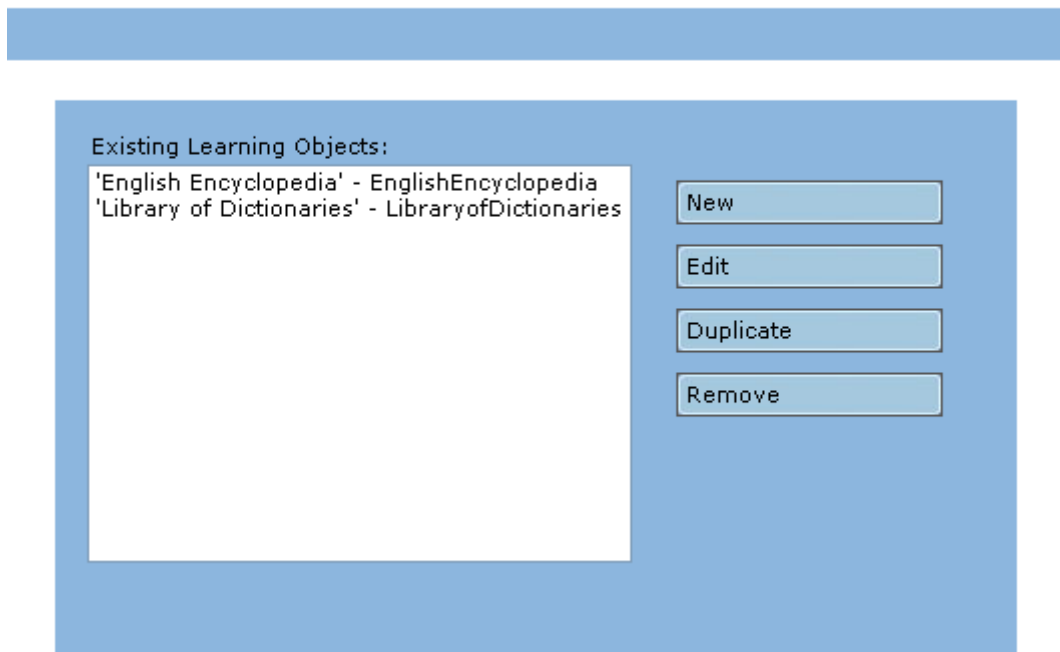
*Examples are: web pages, text books, productivity tools (text processors, editors, calculators,), instruments (microscope, etc.), and test items. A classification of different types of learning objects can be found in the LOM specification (element 5.2 Learning Resource Type makes a distinction between: exercise, simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment, and lecture). A Learning Object may reference any of these types. However this assumes a runtime system that will be able to handle them.”*

The Learning Objects sub-module is the fourth option of the Environments Menu.



The Learning Objects form available to the user is like the following picture.

### Learning Editor: Learning Objects Manager



In this feature the user can:

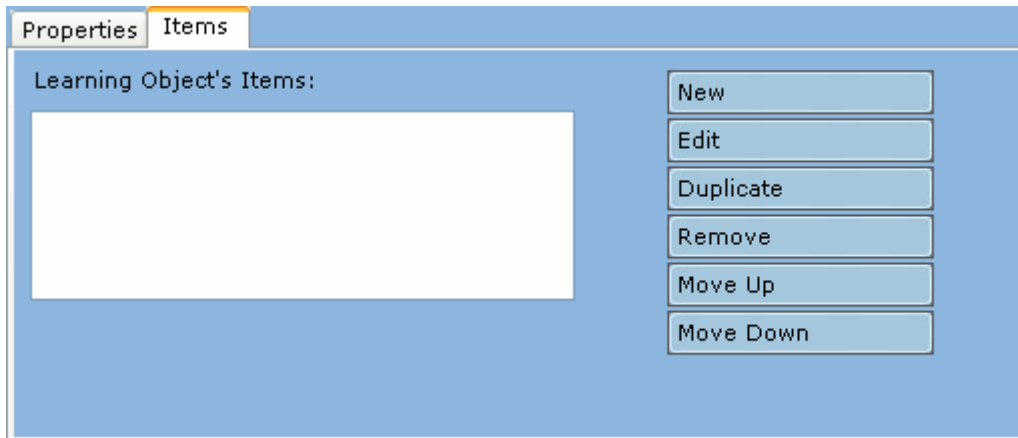
- Create **New Learning Objects** – In this option the tool will present to the user a form, like the one below, to let the user fill the new Learning Object data.

The form consists in the following items:

- **Title** – Represents the title that the user will give to the Learning Object. This is a mandatory field, if this field is not filled the Learning Object can't be created;

**Note:** Despite this value is not the ID of the Learning Object (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing learning object or other kind of element. **According to this, try to put an explicit value on the title of the learning object to let you easily remember and identify the different learning objects;**

- **Type** – Option value that defines the type of this learning object. The options can be “knowledge-object”, “tool-object” and “test-object”;
- **Class** – Text value to the class name assigned to this learning object;
- **Parameters** – Text value to the parameters to be passed during runtime;
- **Is Visible** – Option value to define if this learning object will be visible to the end user of the course;
- **Metadata** – Button that allows the user to define the Metadata associated with this Learning Object. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Items** – Section that includes the list of Items associated to this learning object. These Items will be the objects that relate the learning object with the resources that define the contents of the learning object. For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;



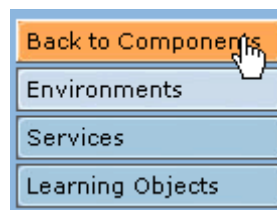
In this section the user can:

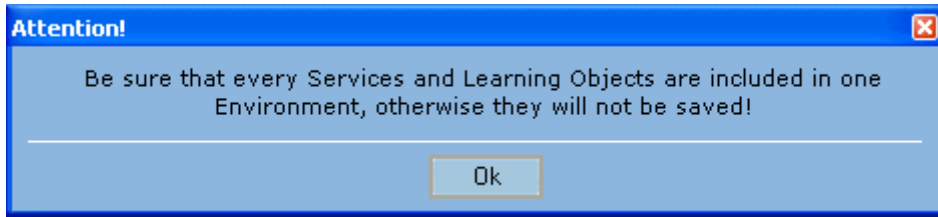
- Create **New** Items;
  - **Edit** existing Items;
  - **Duplicate** existing Items;
  - **Remove** existing Items;
  - **Move Up** existing Items;
  - **Move Down** existing items.
- **Edit** existing Learning Objects – In this option the tool will present to the user a form similar to the previous option but with the chosen Learning Object data filled.  
The user can change all fields of that Learning Object;
  - **Duplicate** existing Learning Objects – In this option the user can select an existing Learning Object and create a copy of that Learning Object.  
The only different values will be the ID and Title of the Learning Object;
  - **Remove** existing Learning Objects – In this option the user can select and delete one existing Learning Objects;

**Note: When you define a Learning Object you need to associate it to an Environment otherwise that Learning Object will not be saved!!!**

**This is due to the fact that Services and Learning Objects are inner concepts of the Environments, but the tool organizes the Services and the Learning Objects separately from the Environments to make the Environment's interface simpler.**

**According to this fact, when the user is leaving the Environments menu, the Authoring Tool launches an alert message to the user. The process is described in the below pictures.**





## 4.11 Activities

The Activity concept can be seen as the steps that are needed to perform the course. The Activities consist in three kinds of objects with different purposes.

Those objects can be Learning Activities, Support Activities and Activity Structures.

In the *IMS-LD* specification we can find the following description about Activities:

*“Activities are one of the core structural elements of the ‘learning workflow’ model for learning design. They form the link between the roles and the learning objects and services in the learning environment. They describe the activities a role has to undertake within a specified environment composed of learning objects and services. They also specify their termination conditions and the actions to be taken on termination. There are two basic types of activities: learning activities and support activities. A learning activity is directed at attaining a learning objective per individual user. Any user performs a learning activity only once (until completion). A support activity is meant to facilitate a role performing one or more learning activities. More than one person can be assigned to a role in runtime. In practice this means that a support activity has to be performed as many times as there are users in the supported role.*

*Activities can be aggregated into an activity-structure which provides the mechanisms to structure activities and referenced units of learning into a sequence or a user-selection.*

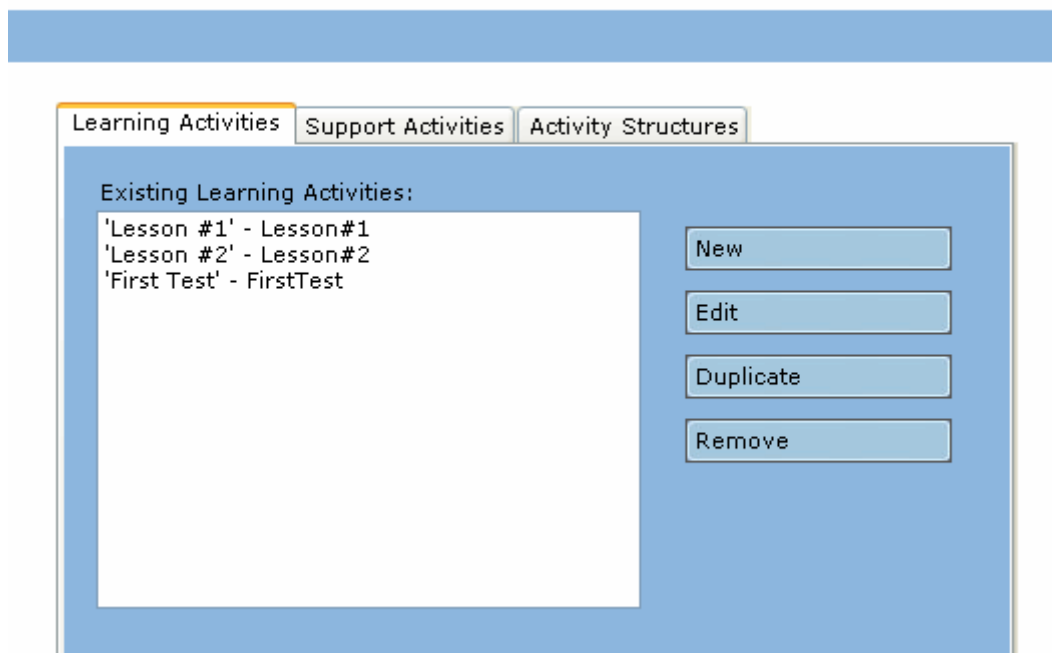
*An activity references the environment in which the activity must be executed. For the execution of any activity, a user needs, at a minimum, an activity description and, optionally, an environment with the learning objects and services needed to perform the activity.”*

The Activities sub-module is the fourth option in the Components menu.



The Activities form available to the user is like the following picture.

### Learning Editor: Activities Manager

The screenshot shows the "Learning Editor: Activities Manager" interface. At the top, there are three tabs: "Learning Activities", "Support Activities", and "Activity Structures". The "Learning Activities" tab is selected. Below the tabs, there is a list of existing learning activities: "'Lesson #1' - Lesson#1", "'Lesson #2' - Lesson#2", and "'First Test' - FirstTest". To the right of the list, there are four buttons: "New", "Edit", "Duplicate", and "Remove".

This form consists in three sections each one corresponding to a type of Activities.

### 4.11.1 Learning Activities

The first type of activities that we have is the Learning Activities.

These are the core of the Learning Design.

Is in these activities that we define the learning episodes that can be available to the learners.

In the *IMS-LD* specification we can find the following description about Learning Activities:

*“A Learning Activity consists of a single activity-description and several optional elements. The activity-description is the actual cue given to the user (rendered in the user-interface) to describe the activity to be performed by the user. In most cases the activity-description is a text (of type webcontent). In other cases it can be an audio-file (webcontent), a video file or any other cue to the user. Whatever forms it takes, the activity-description is referenced via an <item> element, derived from Content Packaging, referencing a resource element in the content package.*

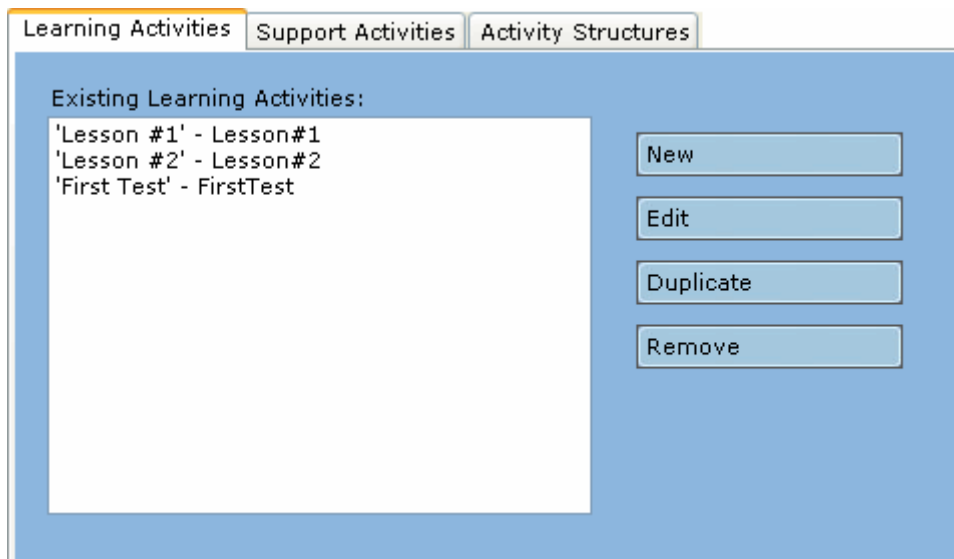
*In addition to Environment reference(s), the other optional elements include:*

*title, IMS metadata, learning-objectives, prerequisites (see above), and the new elements: complete-activity (which specifies when an activity is completed, in Level A either by user-choice, or on reaching a time-limit, and extended by Level B with when-property-value-is-set), and on-completion (which specifies actions that are to be executed on completion of the activity).*

*In level A, on-completion contains only one element, feedback-description, which references content to be displayed to the user when they finish.*

*on-completion is further extended at level B by the change-property-value element and at Level C by the notification element).”*

The Learning Activities section of the Activities form available to the user is like the following picture.



In this section the user can:

- Create **New** Learning Activities – In this option the tool will present to the user a form to let the user fill the new Learning Activity data.  
Since this form needs an extensive explanation see the “Learning Activity Form” section of this sub-module;
- **Edit** existing Learning Activities – In this option the tool will present to the user a form similar to the previous option but with the chosen Learning Activity data filled.  
The user can change all fields of that Learning Activity;

- **Duplicate** existing Learning Activities – In this option the user can select an existing Learning Activity and create a copy of that Learning Activity.  
The only different values will be the ID and Title of the Learning Activity;
- **Remove** existing Learning Activities – In this option the user can select and delete one existing Learning Activities;

#### 4.11.1.1 Learning Activity Form

The Learning Activity form available to the user is like the below picture.

The screenshot shows a dialog box titled "New Learning Activity" with a blue header and a close button in the top right. The main area is divided into several sections. At the top, there are tabs: "Properties" (selected), "Learning Objectives", "Pre-Requisites", "Description", "Final Feedback", and "Pi". Below the tabs, there are input fields for "Title:" and "Parameters:". To the right of these is a dropdown menu for "Environment(s):" with a list of options: "'Conference Environm...", "'Send Mail Environmen...", "'Index Environment' -", and "'Services Environment...". Below the input fields is a section titled "Completed Activity Options:" containing four radio button options: "None of Below options" (selected), "Activity has a Time Limit" (with H, M, S input boxes), "User Can say when activity is completed", and "Complete when set a Property Value" (with a "Define" button). To the right of this section is a checked checkbox "Is Visible" and a "Metadata" button. At the bottom of the dialog are "Ok" and "Cancel" buttons.

This form is divided in seven sections, which two of them (“Property Change” and “Notifications”) are related to Level B or C of *IMS-LD* specification. So they are inactive and don’t fall in the scope of this Authoring Tool version.

The sections that belong to Level A and are active are:

- **Properties** – These are the main properties of the Learning Activity.

This is a duplicate of the screenshot above, showing the "New Learning Activity" dialog box with the "Properties" tab selected. It displays the same fields for Title, Parameters, Environment(s), Completed Activity Options, Is Visible checkbox, Metadata button, and Ok/Cancel buttons.

The properties fields are:

- **Title** – Represents the title that the user will give to the Learning Activity. This is a mandatory field, if this field is not filled the Learning Activity can’t be created;

**Note:** Despite this value is not the ID of the Learning Activity (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing learning activity or other kind of element. **According to this, try to put an explicit value on the title of the learning activity to let you easily remember and identify the different learning activities;**

- **Environment(s)** – List to the user select which environments are needed for this activity could be performed.  
The list is of every Environments defined in this Course;
- **Parameters** – Text value to the parameters to be passed during runtime;
- **Completed Activity Options** – Option value to the user defines how can be considered the activity completed.  
The options are “Activity has a Time Limit” and “User Can say when activity is completed”.  
In the “Activity has a Time Limit” option the activity will considered completed after the time defined on this value(sum of defined hours, minutes and seconds), **relative to the start of the course**. If this option is selected the user must insert a time value on the numeric fields (hours, minutes and seconds) in front of the option.  
In the “User Can say when activity is completed” option the activity may be considered completed when the end user (typically learner) decides in that way.
- **Is Visible** – Option value to define if this Learning Activity will be visible to the end user of the course;
- **Metadata** – Button that allows the user to define the Metadata associated with this Learning Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

- **Learning Objectives** – This is a list of Items that correspond to the specific Learning Objectives of this Activity.

To see more information about the concept of Learning Objective please read the “Learning Objectives” sub-module of the “Learning Editor sub-modules” section.

The Learning Objectives fields are:

- **Activity's Learning Objectives Items** – List of Items that represents the Learning Objectives of this Learning Activity.



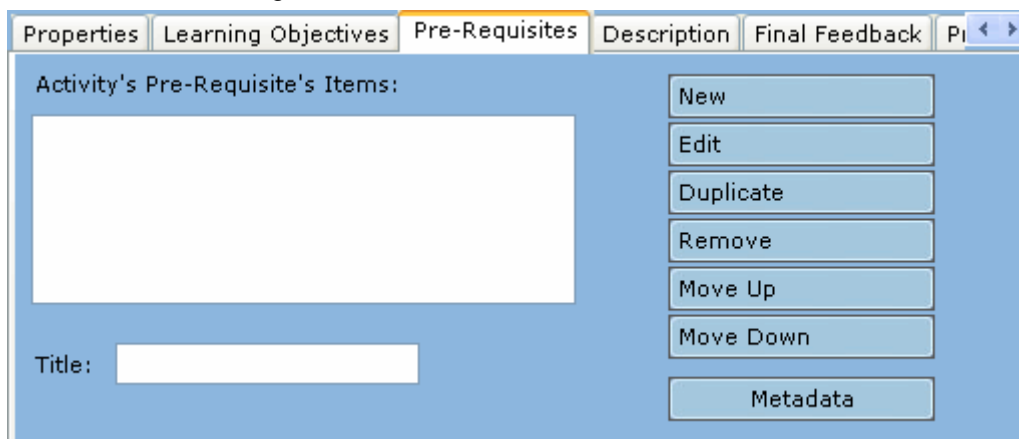
This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:

- Create **New** Items;
- **Edit** existing Items;
- **Duplicate** existing Items;
- **Remove** existing Items;
- **Move Up** existing Items;
- **Move Down** existing Items;

For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;

- **Title** – Text value to define a specific title for the Learning Objectives of this Learning Activity.
- **Metadata** – Button that allows the user to define the Metadata associated with the Learning Objectives of this Learning Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Pre-Requisites** – This is a list of Items that correspond to the specific Pre-Requisites of this Activity.

To see more information about the concept of pre-requisite please read the “Pre-Requisites” sub-module of the “Learning Editor sub-modules” section.



The screenshot shows a software interface with several tabs: Properties, Learning Objectives, Pre-Requisites (which is selected and highlighted in orange), Description, Final Feedback, and Pi. The Pre-Requisites tab contains a large empty rectangular area labeled "Activity's Pre-Requisite's Items:". To the right of this area is a vertical stack of seven buttons: "New", "Edit", "Duplicate", "Remove", "Move Up", "Move Down", and "Metadata". Below the main area, there is a "Title:" label followed by an empty text input field.

The Pre-Requisites fields are:

- **Activity's Pre-Requisite's Items** – List of Items that represents the Pre-Requisites of this Learning Activity.

This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:

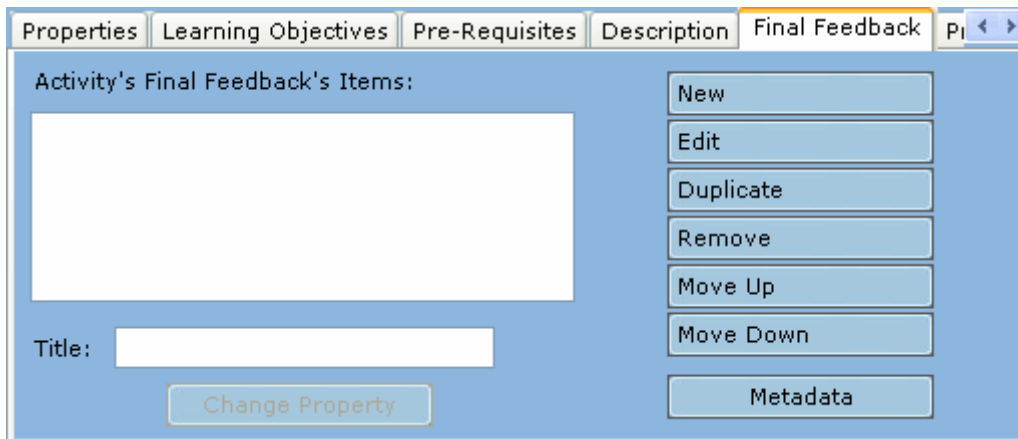
- Create **New** Items;
- **Edit** existing Items;
- **Duplicate** existing Items;
- **Remove** existing Items;
- **Move Up** existing Items;
- **Move Down** existing Items;

For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;

- **Title** – Text value to define a specific title for the Pre-Requisites of this Learning Activity.
- **Metadata** – Button that allows the user to define the Metadata associated with the Pre-Requisites of this Learning Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Description** – This is a list of Items that will describe to the end user the activity to be performed by him.  
The user must introduce at least one Item, otherwise the Learning Activity can't be created because this is a mandatory section.

The Description fields are:

- **Activity's Description's Items** – List of Items that represents the Description of this Learning Activity.  
This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:
  - Create **New** Items;
  - **Edit** existing Items;
  - **Duplicate** existing Items;
  - **Remove** existing Items;
  - **Move Up** existing Items;
  - **Move Down** existing Items;
 For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;
- **Title** – Text value to define a specific title for the Description of this Learning Activity.
- **Metadata** – Button that allows the user to define the Metadata associated with the Description of this Learning Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Final Feedback** – This is a list of Items that will be shown to the end user when activity is completed.



The Final Feedback fields are:

- **Activity's Final Feedback's Items** – List of Items that represents the Final Feedback of this Learning Activity.  
This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:
  - **Create New** Items;
  - **Edit** existing Items;
  - **Duplicate** existing Items;
  - **Remove** existing Items;
  - **Move Up** existing Items;
  - **Move Down** existing Items;For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;
- **Title** – Text value to define a specific title for the Final Feedback of this Learning Activity.
- **Metadata** – Button that allows the user to define the Metadata associated with the Final Feedback of this Learning Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

### 4.11.2 Support Activities

The second type of activities that we have is the Support Activities.

These are the activities that will support the execution of the learning activities and the flow of the course. A good example on a Support Activity can be the assessment of a test.

In the *IMS-LD* specification we can find the following description about Support Activities:

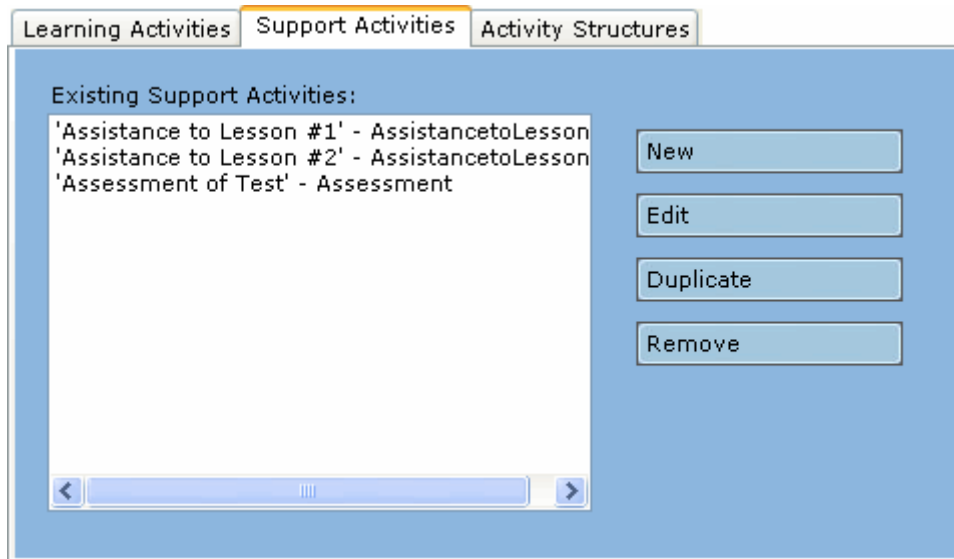
*“A Support Activity consists mostly of the same elements as a Learning Activity, but without the learning-objectives, prerequisites, and with a role-ref element added. The role-ref element indicates who will be supported by this activity.*

*The supported role can have more than one person in the role. This means in practice that the support activity has to be repeated for every user in the supported role before it is completed. This is a key difference from learning activities, which is only performed once. Example: a staff role has the support activity to grade reports made by persons in the learner role named ‘student’. Every person being a student creates his/her own report. The tutor grades every report (repeating the ‘grade report’ support activity).*

(...)

*A support activity can optionally be connected to a role. This means that the activity is repeated for every member in the supported role (learner/staff). Support activities are typically performed by staff members (e.g. tutors) to support learners. In some pedagogical models however learners can support learners (peer to peer teaching). It is also possible that staff members support staff members. When the optional role-ref element is set, it is expected that the support activity will act for every single user in the specified role(s). That is: the same support activity is repeated for every user in the role(s).”*

The Support Activities section of the Activities form available to the user is like the following picture.



In this section the user can:

- Create **New** Support Activities – In this option the tool will present to the user a form to let the user fill the new Support Activity data. Since this form needs an extensive explanation see the “Support Activity Form” section of this sub-module;
- **Edit** existing Support Activities – In this option the tool will present to the user a form similar to the previous option but with the chosen Support Activity data filled.  
The user can change all fields of that Support Activity;
- **Duplicate** existing Support Activities – In this option the user can select an existing Support Activity and create a copy of that Support Activity.  
The only different values will be the ID and Title of the Support Activity;

- **Remove** existing Support Activities – In this option the user can select and delete one existing Support Activities;

#### 4.11.2.1 Support Activity Form

The Support Activity form available to the user is like the below picture.

This form is divided in five sections, which two of them (“Property Change” and “Notifications”) are related to Level B or C of *IMS-LD* specification. So they are inactive and don’t fall in the scope of this Authoring Tool version.

The sections that belong to Level A and are active are:

- **Properties** – These are the main properties of the Support Activity.

The properties fields are:

- **Title** – Represents the title that the user will give to the Support Activity. This is a mandatory field, if this field is not filled the Support Activity can’t be created;

**Note:** Despite this value is not the ID of the Support Activity (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing support activity or other kind of element. **According to this, try to put an explicit**

**value on the title of the support activity to let you easily remember and identify the different support activities;**

- **Role(s)** – List to the user select which Roles will be supported by this activity. The activity is repeated for every member in the supported role (learner/staff). The list is of every Roles defined in this Course;
- **Parameters** – Text value to the parameters to be passed during runtime;
- **Environment(s)** – List to the user select which environments are needed for this activity could be performed. The list is of every Environments defined in this Course;
- **Completed Activity Options** – Option value to the user defines how can be considered the activity completed.  
The options are “Activity has a Time Limit” and “User Can say when activity is completed”.  
In the “Activity has a Time Limit” option the activity will considered completed after the time defined on this value(sum of defined hours, minutes and seconds), **relative to the start of the course**. If this option is selected the user must insert a time value on the numeric fields (hours, minutes and seconds) in front of the option.  
In the “User Can say when activity is completed” option the activity may be considered completed when the end user (typically learner) decides in that way.
- **Is Visible** – Option value to define if this Support Activity will be visible to the end user of the course;
- **Metadata** – Button that allows the user to define the Metadata associated with this Support Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Description** – This is a list of Items that will describe to the end user the activity to be performed by him.  
The user must introduce at least one Item, otherwise the Support Activity can't be created because this is a mandatory section.

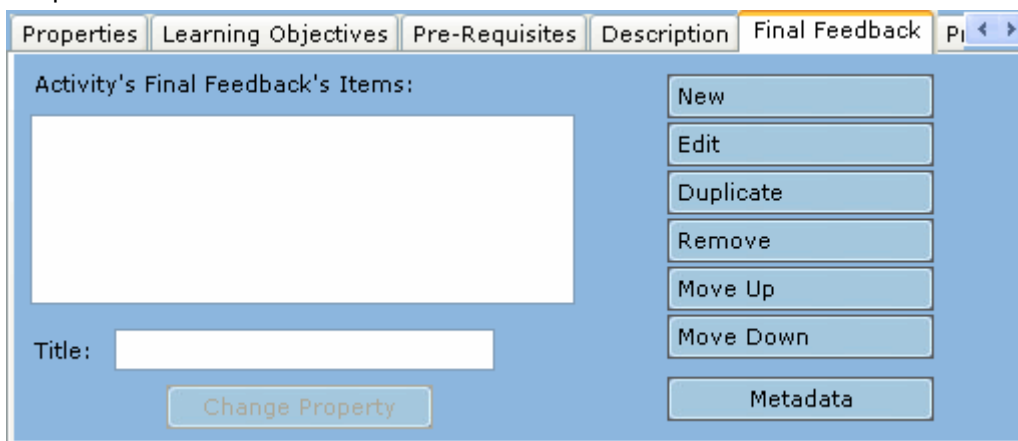
The Description fields are:

- **Activity's Description's Items** – List of Items that represents the Description of this Support Activity.  
This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:

- Create **New** Items;
- **Edit** existing Items;
- **Duplicate** existing Items;
- **Remove** existing Items;
- **Move Up** existing Items;
- **Move Down** existing Items;

For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;

- **Title** – Text value to define a specific title for the Description of this Support Activity.
- **Metadata** – Button that allows the user to define the Metadata associated with the Description of this Support Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Final Feedback** – This is a list of Items that will be shown to the end user when activity is completed.



The Final Feedback fields are:

- **Activity's Final Feedback's Items** – List of Items that represents the Final Feedback of this Support Activity.

This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:

- Create **New** Items;
- **Edit** existing Items;
- **Duplicate** existing Items;
- **Remove** existing Items;
- **Move Up** existing Items;
- **Move Down** existing Items;

For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;

- **Title** – Text value to define a specific title for the Final Feedback of this Support Activity.
- **Metadata** – Button that allows the user to define the Metadata associated with the Final Feedback of this Support Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

### 4.11.3 Activity Structures

The third and last type of activities that we have is the Activity Structures.

These activities are groups of activities. We can see the Activity Structures as a subjective activity that is composed by several minor activities.

In the Activity Structures the user can include any kind of activities.

In the *IMS-LD* specification we can find the following description about Support Activities:

*“An Activity-structure in turn consists of references to one or more of:*

- *A Learning Activity*
- *A Support Activity*
- *An (sub) Activity-structure*

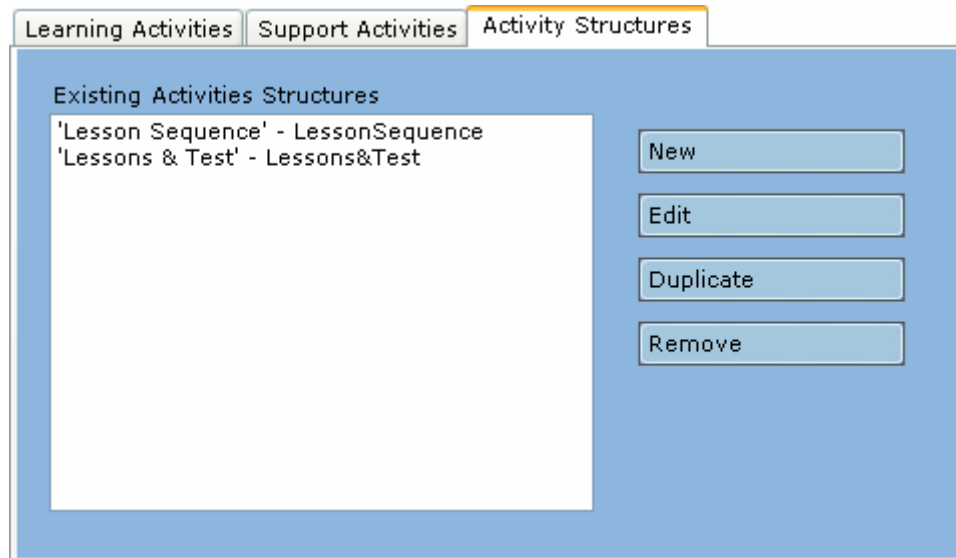
(...)

*Like the single Activities, an Activity-structure may reference one or more environments. This allows for learning design models where a series of different activities are performed within the same environment. When an activity-structure references one or more environments, then these will overrule the environments specified within the referenced activities.*

*The environments will not be inherited between hierarchical levels of the Activity-structure, allowing environments to be omitted as well. As a consequence, for each hierarchical level of the Activity-structure the appropriate reference to an environment has to be made and possibly repeated.*

*A structure may contain information. This provides a CP Organization/Item structure which provides links to resources which contain further information about the activity-structure.”*

The Activity Structures section of the Activities form available to the user is like the following picture.



In this section the user can:

- Create **New** Activity Structures – In this option the tool will present to the user a form to let the user fill the new Activity Structure data. Since this form needs an extensive explanation see the “Activity Structure Form” section of this sub-module;
- **Edit** existing Activity Structures – In this option the tool will present to the user a form similar to the previous option but with the chosen Activity Structure data filled. The user can change all fields of that Activity Structure;



- **Duplicate** existing Activity Structures – In this option the user can select an existing Activity Structure and create a copy of that Activity Structure. The only different values will be the ID and Title of the Activity Structure;
- **Remove** existing Activity Structures – In this option the user can select and delete one existing Activity Structures;

#### 4.11.3.1 Activity Structure Form

The Activity Structure form available to the user is like the below picture.

This form is divided in five sections. Those sections are:

- **Properties** – These are the main properties of the Activity Structure.

The properties fields are:

- **Title** – Represents the title that the user will give to the Activity Structure. This is a mandatory field, if this field is not filled the Activity Structure can't be created;

**Note:** Despite this value is not the ID of the Activity Structure (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing activity structure or other kind of element. **According to this, try to put an explicit**

**value on the title of the activity structure to let you easily remember and identify the different activity structures;**

- **Environment(s)** – List to the user select which environments are needed for this activity could be performed. The list is of every Environments defined in this Course;
- **Metadata** – Button that allows the user to define the Metadata associated with this Support Activity. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Number of Activities to consider structure completed** – Numeric value that, if filled, define the number of completed activities inside the structure are needed to considered the structure completed.  
If this value is not defined the structure will only be completed when all the activities are completed;
- **Choose Activities Presentation** – Option value to the user defines how the activities will be presented to the end user.  
The options are “Show Activities Sequentially” and “Let the User Choose the Order”.  
In the “Show Activities Sequentially” option the activities will be shown to the user one after another by the order defined in the other Activity Structure’s section.  
In the “Let the User Choose the Order” option the run time system will present some kind of menu where the user will choose the order of the activities that we wants to perform.
- **Choose Activities Order** – Option value to the user defines how the order of the activities is.  
The options are “As Is” and “In Visibility Order”.  
In the “As Is” option the activities will be ordered like defined in the other sections.  
In the “In Visibility Order” option the activities will be ordered by the visibility.
- **Information** – This is a list of Items that will give some additional information to the end user about this structure.

The screenshot shows a software interface with a blue header and a white content area. The header has five tabs: 'Properties', 'Information' (highlighted in orange), 'Learning Activities', 'Support Activities', and 'Activity Structures'. Below the tabs, the main area is titled 'Activity Structure Information's Item'. On the left, there is a large empty white rectangular box. To the right of this box is a vertical stack of six buttons: 'New', 'Edit', 'Duplicate', 'Remove', 'Move Up', and 'Move Down'. Below the large box, there is a 'Title:' label followed by a white text input field. At the bottom right of the main area, there is a 'Metadata' button.

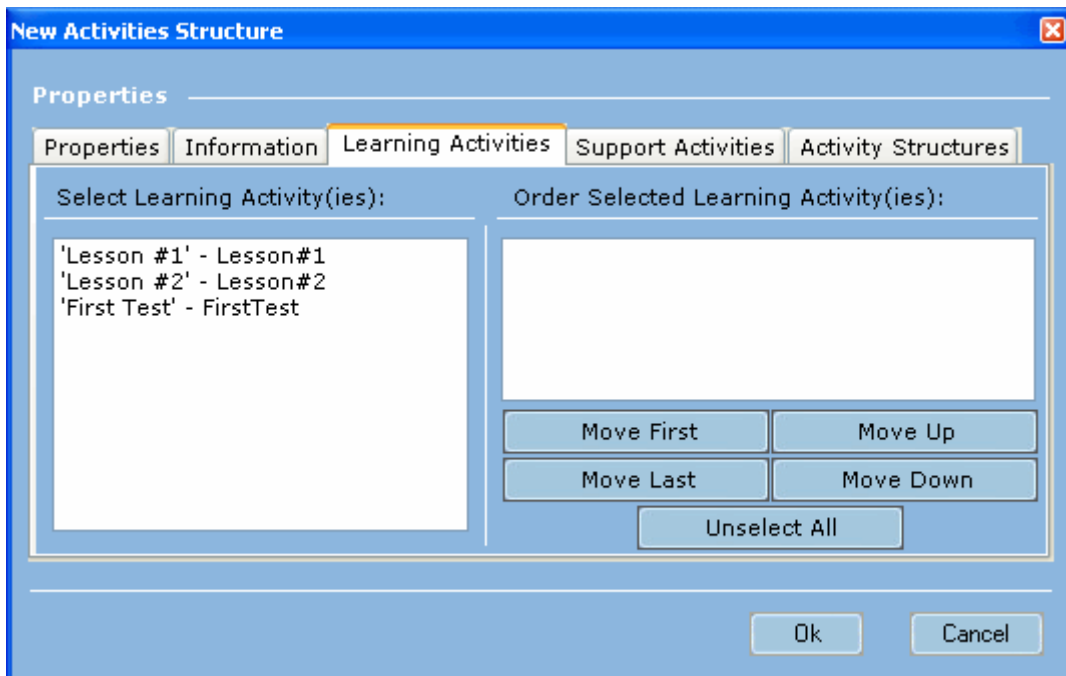
The Information fields are:

- **Activity Structure Information’s Items** – List of Items that represents the Information of this Activity Structure.  
This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:
  - Create **New** Items;

- **Edit** existing Items;
- **Duplicate** existing Items;
- **Remove** existing Items;
- **Move Up** existing Items;
- **Move Down** existing Items;

For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;

- **Title** – Text value to define a specific title for the Information of this Activity Structure.
- **Metadata** – Button that allows the user to define the Metadata associated with the Information of this Activity Structure. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Learning Activities** – This is the section where the user can select which Learning Activities will belong to this Activity Structure.



The Learning Activities section has two lists.

The first list (on the left side) is the list of every Learning Activities defined in the course. The user must select the Learning Activities that he wants to use on this Activity Structure.

The second list (on the right side) is the list of the selected Learning Activities.

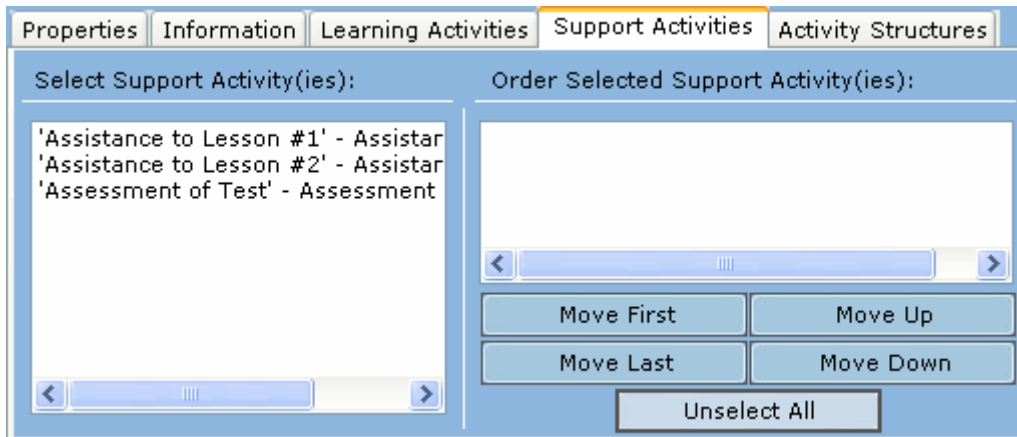
This list is only active is the user select the “Show Activities Sequentially” option of the “Choose Activities Presentation” field of the “Properties” section.

When the user select activities on the first list the tool will refresh the second list and then, the user can manipulate the order of the chosen activities.

To manipulate the second list the form has five buttons that can execute the ordinance operations. Those operations are:

- **Move First** – Move the selected activity to the first position;
- **Move Last** – Move the selected activity to the last position;
- **Move Up** – Move the selected activity one position upper;
- **Move Down** – Move the selected activity one position lower;
- **Unselect All** – Unselect all elements of the left list emptying the right list.

- **Support Activities** – This is the section where the user can select which Support Activities will belong to this Activity Structure.



The Support Activities section has two lists.

The first list (on the left side) is the list of every Support Activities defined in the course. The user must select the Support Activities that he wants to use on this Activity Structure.

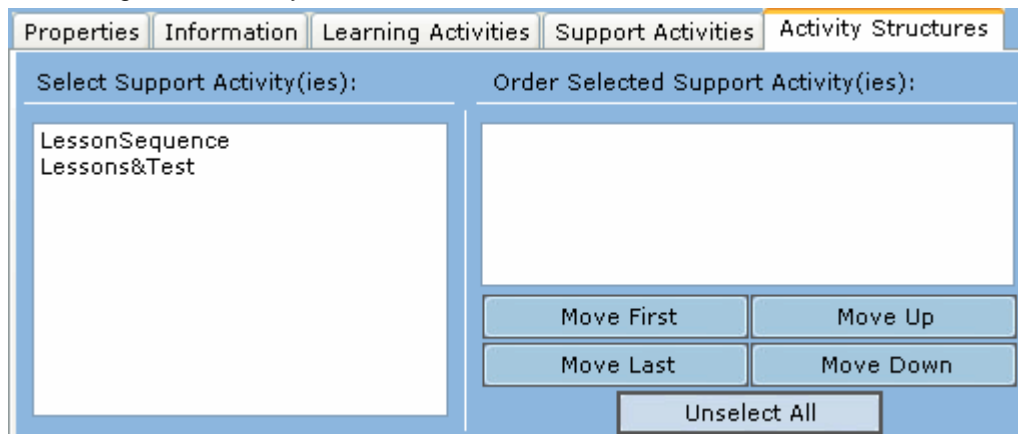
The second list (on the right side) is the list of the selected Support Activities.

This list is only active is the user select the “Show Activities Sequentially” option of the “Choose Activities Presentation” field of the “Properties” section.

When the user select activities on the first list the tool will refresh the second list and then, the user can manipulate the order of the chosen activities.

To manipulate the second list the form has five buttons that can execute the ordinance operations. Those operations are:

- **Move First** – Move the selected activity to the first position;
  - **Move Last** – Move the selected activity to the last position;
  - **Move Up** – Move the selected activity one position upper;
  - **Move Down** – Move the selected activity one position lower;
  - **Unselect All** – Unselect all elements of the left list emptying the right list.
- **Activity Structures** – This is the section where the user can select which (sub) Activity Structures will belong to this Activity Structure.



The Activity Structures section has two lists.

The first list (on the left side) is the list of every Activity Structures defined in the course. The user must select the Activity Structures that he wants to use on this Activity Structure.

The second list (on the right side) is the list of the selected Activity Structures.

This list is only active if the user selects the "Show Activities Sequentially" option of the "Choose Activities Presentation" field of the "Properties" section.

When the user selects activities on the first list, the tool will refresh the second list, and then, the user can manipulate the order of the chosen activities.

To manipulate the second list, the form has five buttons that can execute the ordinance operations. Those operations are:

- **Move First** – Move the selected activity to the first position;
- **Move Last** – Move the selected activity to the last position;
- **Move Up** – Move the selected activity one position up;
- **Move Down** – Move the selected activity one position down;
- **Unselect All** – Unselect all elements of the left list, emptying the right list.

## 4.12 Method

The Method sub-module can be seen as a group of concepts that will be compound and organize the execution sequence of the course. In the Method option we can find concepts such as Plays, Acts and Parts.

In the *IMS-LD* specification we can find the following description about Method:

*“This and its sub-elements control the behavior of the unit of learning as a whole, coordinating the activities of the players in the various roles and their use of resources.*

*This creates a ‘learning flow’, similar to the coordination of activities in a workflow groupware system (but not to the passage of documents in a document-oriented workflow system).*

*Functionally, a method is made up of one or more play elements. Play elements are functionally independent and run in parallel, so each play element has to be instantiated when the unit of learning is first initialized.*

*The terminology used to describe the various sub-parts of a Method, Play, Act, Role, and Role-part are drawn from the metaphor of theatre, with the Environment equivalent to the stage set and stage props. The Method, Plays, Acts, and Role-parts are all nested within each other, providing three levels within a Method.*

*At the top level, the method consists of the 2 elements, play and complete-unit-of-learning. The latter, as described in the Completion Rules section that follows, holds both the condition on which the unit-of-learning is completed and optional actions to be taken when it is completed.*

*On initialization, all play elements are made active for the members of the roles that participate in it.*

*Play elements unfold in a series of one or more acts, which are always run in sequence. No act is made visible to the role/s that play a part in it until the previous act has completed. This factor can be used to synchronize the activities of those playing roles within the play.*

*The activities associated with the roles within a play may have complex sequencing, so it is possible to have a ‘one-act’ play which has internally complex sequencing, as long as the activities of the players do not need synchronizing (e.g., when the use of a learning resource is completed, all participants move to a discussion forum).*

*(...)*

*Although within an act there may be complex sequencing, and there may be only one active act, if there is more than one act within a play, the acts are run in sequence.*

*Thus acts can be used as synchronization points, either waiting for all players to finish before starting the next, or forcing an end when a certain number of players have finished, when terminated by a support staff member or under some other condition. All role participants in the next act may then start together at the same time, subject to their being simultaneously logged into the runtime service. When the last act is completed, the unit of learning is also complete.*

*An act brings together one or more role-parts. This is the mechanism that allows more than one role to perform at the same time. Therefore role-parts within an act are always run in parallel.”*

Role-parts and Parts are the same concept.

*(...)*

*Role-parts enable several users, playing the same or different roles, to participate in the same act. Each role-part associates exactly one role with exactly one type of activity (including the performance of another unit-of-learning and activity-structures), or with one environment (equivalent to an organization in Content Packaging). Multiple role-parts within one act are performed concurrently.*

*The same role can be associated with different activities or environments in different role-parts, and the same activity or environment can be associated with different roles in different role-parts. However, the same role may only be referenced once in the same act. If multiple activities or environments need to be associated for the same role an activity-structure or wrapper environment should be used.*

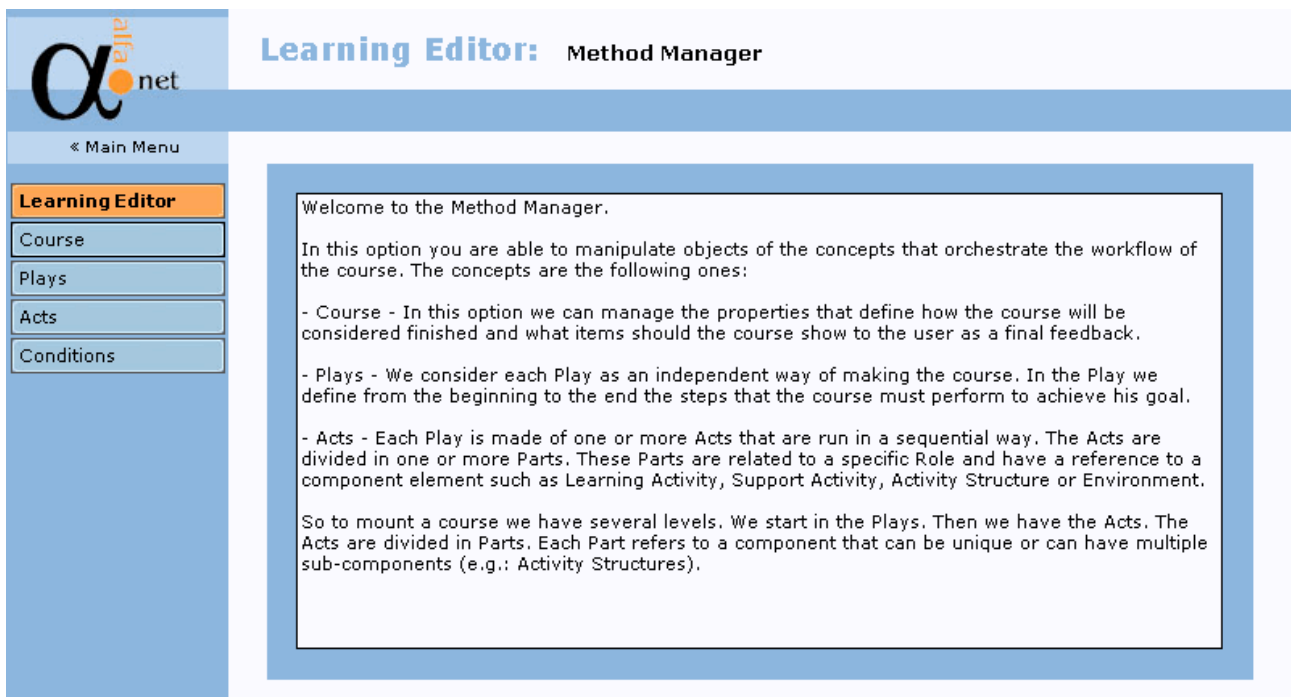
*When an act within a play is activated, all the role-parts in the act go ‘on-stage’ or become live. Depending on the implementation, players of the roles referenced by the role-parts may then have their associated activities (or environments) made visible in their ‘activity-tree’ and any content associated with the activity made accessible.*

*However, if an activity or items isvisible attribute is set to ‘false’, the link in the activity-tree might be made visible, but the content not made accessible.”*

The Method sub-module is the seventh option of the Learning Editor menu:



The Method sub-module is like the following picture.



When the user clicks in the “Method” button the tool shows a screen with a brief text to help the user and explain every concepts that belongs to the Method sub-module.

The tool also changes the Learning Editor menu to the Method menu that is like the below picture.



In this menu we can:

- Accede to the tool Main Menu by the “<< Main Menu” link;

- Go back to the Learning Editor menu by the “Learning Editor” button;
- Accede to the concepts that are involved on the Method clicking on the related buttons (Course, Plays, Acts and Conditions).

**Note:** *The Conditions concept belongs to the Level B or C of the IMS-LD specification so the target of this link will be inactive because this tool version supports only Level A of the IMS-LD specification.*

The following sections will explain in more detail the concepts that belong to the Level A of the Method (Course, Plays and Acts).



## 4.13 Course Completion

The Course Completion sub-module is the second option of the Method menu.

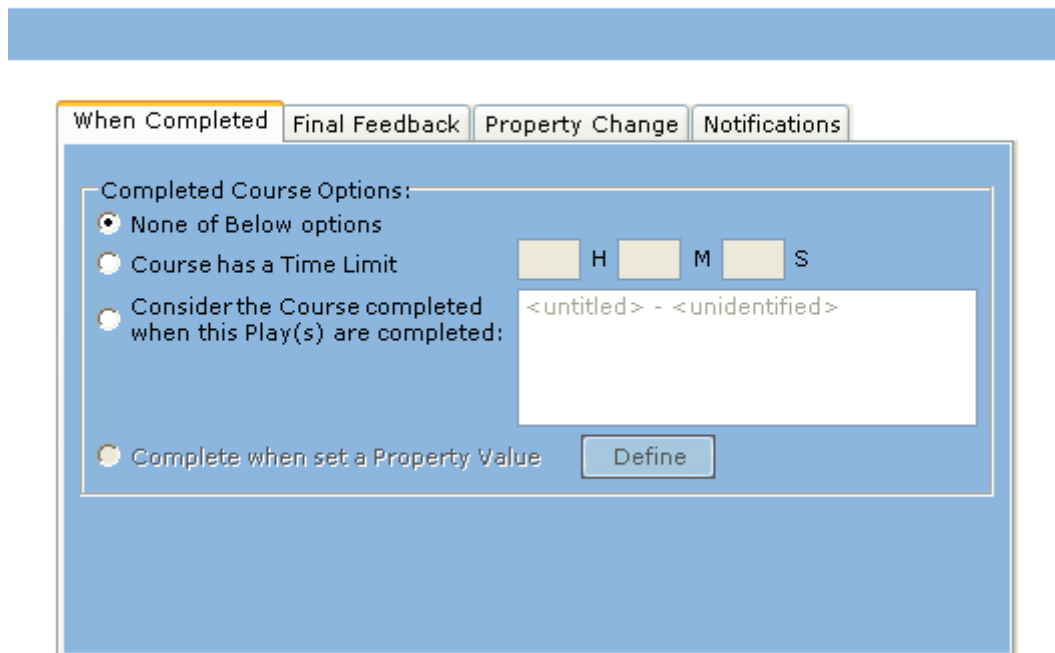


This module consists in the information to consider the Course completed and the information to produce when the Course is finished.

This module refers to the concept "Course" of the "Method" module.

The below picture shows how is the form available to the user:

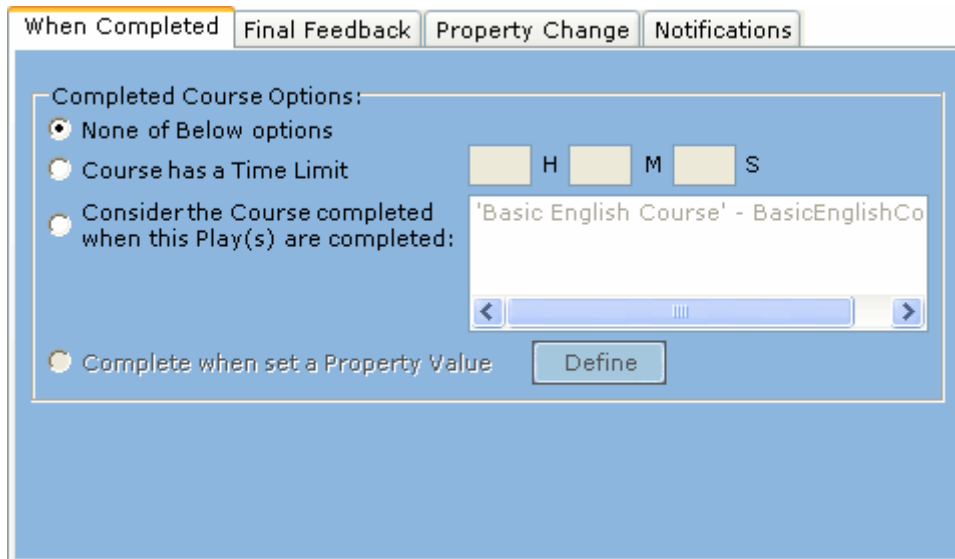
### Learning Editor: Course Completion Manager

The screenshot shows a software interface titled "Learning Editor: Course Completion Manager". It features a blue header bar and a main content area with four tabs: "When Completed", "Final Feedback", "Property Change", and "Notifications". The "When Completed" tab is active. Below the tabs, there is a section titled "Completed Course Options:" with four radio button options: "None of Below options" (selected), "Course has a Time Limit", "Consider the Course completed when this Play(s) are completed:", and "Complete when set a Property Value". To the right of the "Course has a Time Limit" option are three input fields labeled "H", "M", and "S". Below the "Consider the Course completed when this Play(s) are completed:" option is a text area containing the text "<untitled> - <unidentified>". Below the "Complete when set a Property Value" option is a "Define" button.

This form is divided in four sections, which two of them ("Property Change" and "Notifications") are related to Level B or C of *IMS-LD* specification. So they are inactive and don't fall in the scope of this Authoring Tool version.

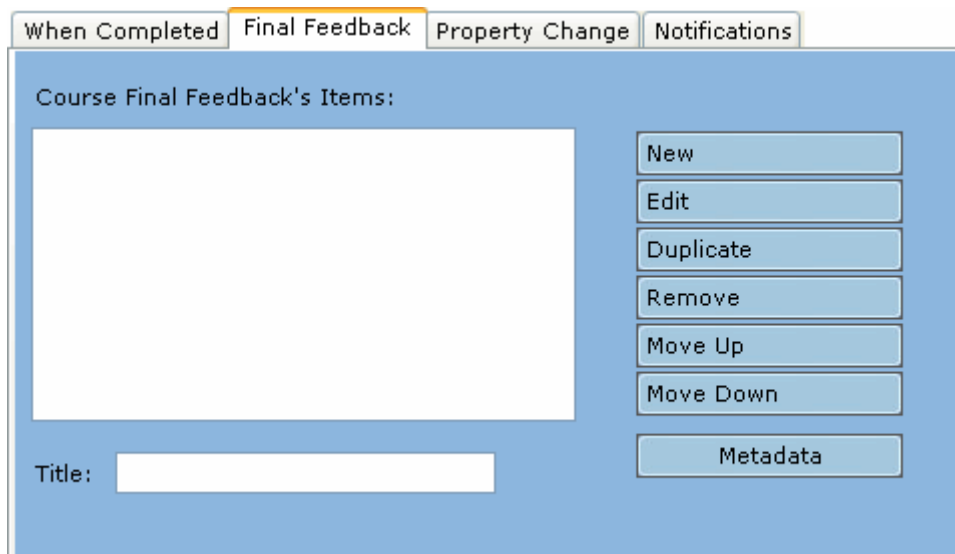
The sections that belong to Level A and are active are:

- **When Completed** – This section consists in an option value where the user decides when does this course can be consider complete.



The options are:

- **Course has a Time limit** – In this option the user defines that the course has a time limit to be executed and will be considered completed after that time.  
 The time is defined has a sum of hours ('H'), minutes ('M') and seconds ('S') defined in numeric fields in front of the option.  
 If this option is selected the user must insert a time value on that numeric fields;
- **Consider the Course completed when this play(s) are completed** – In this option the user defines that the course will be considered completed when a certain Play(s) is completed.  
 This is defined in the list in front of the option. The user will select which Play(s) correspond to this option. The list is of every plays defined in the Course.  
 If the user selects more than one Play this can be seen that all those selected Plays need to be completed to consider the course completed.  
 If this option is selected the user must select at least one Play from the list;
- **Final Feedback** – This is a list of Items that will be shown to the end user when the Course is completed.



The Final Feedback fields are:

- **Course Final Feedback's Items** – List of Items that represents the Final Feedback of this Course.

This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:

- Create **New** Items;
- **Edit** existing Items;
- **Duplicate** existing Items;
- **Remove** existing Items;
- **Move Up** existing Items;
- **Move Down** existing Items;

For more information about the Items concept please read the "Items" section of the "Learning Editor sub-modules" section;

- **Title** – Text value to define a specific title for the Final Feedback of this Course.
- **Metadata** – Button that allows the user to define the Metadata associated with the Final Feedback of this Course. Please see "Metadata Information Categories" section of the "Manage Global Metadata" module to see more information how to do this step;

## 4.14 Plays

The Plays sub-module is the second option of the Method menu.

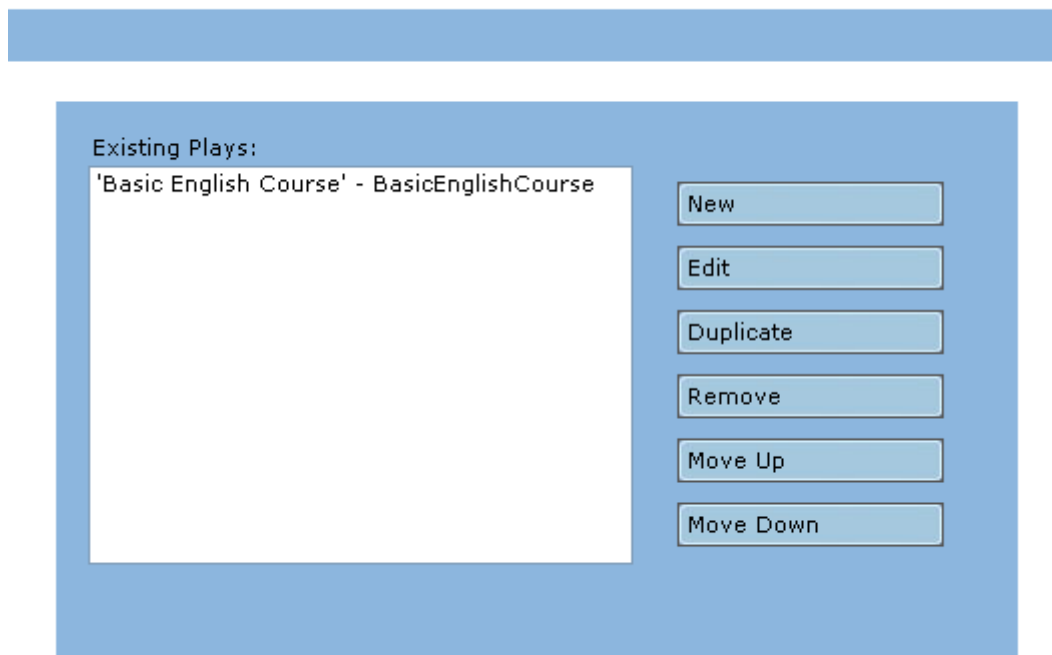


In the *IMS-LD* specification we can find the following description about Plays:

*“The core part of the learning design is represented in the ‘play’. A play specifies the actual learning design, the teaching-learning process, referring to the components declared earlier. In the play, it is specified which roles perform what activities in what order. When reading a learning design one basically reads the play. This is true for human readers as well as machines. Components not referenced in the play are not shown in the runtime system. A play is modeled according to a theatrical play with acts and role-parts. In general: a play consists of a sequence of acts. In each act, different activities are set for different roles and are performed in parallel. When an act is completed, the next act starts until the completion requirements for the learning design are met.”*

The below picture shows how is the form available to the user:

### Learning Editor : Plays Manager



This feature allows the user to:

- Create **New** Plays – In this option the tool will present to the user a form to let the user fill the new Play data. Since this form needs an extensive explanation see the “Play Form” section of this sub-module;
- **Edit** existing Plays – In this option the tool will present to the user a form similar to the previous option but with the chosen Play data filled. The user can change all fields of that Play;

- **Duplicate** existing Plays – In this option the user can select an existing Play and create a copy of that Play. The only different values will be the ID and Title of the Play;
- **Remove** existing Plays – In this option the user can select and delete one existing Play;
- **Move Up** existing Plays – In this option the user can select and move up one position the selected Play in the list order;
- **Move Down** existing Plays – In this option the user can select and move down one position the selected Play in the list order.

#### 4.14.1 Play Form

The Play form available to the user is like the below picture.

This form is divided in six sections, which two of them (“Property Change” and “Notifications”) are related to Level B or C of *IMS-LD* specification. So they are inactive and don’t fall in the scope of this Authoring Tool version.

The sections that belong to Level A and are active are:

- **Properties** – These are the main properties of the Play.

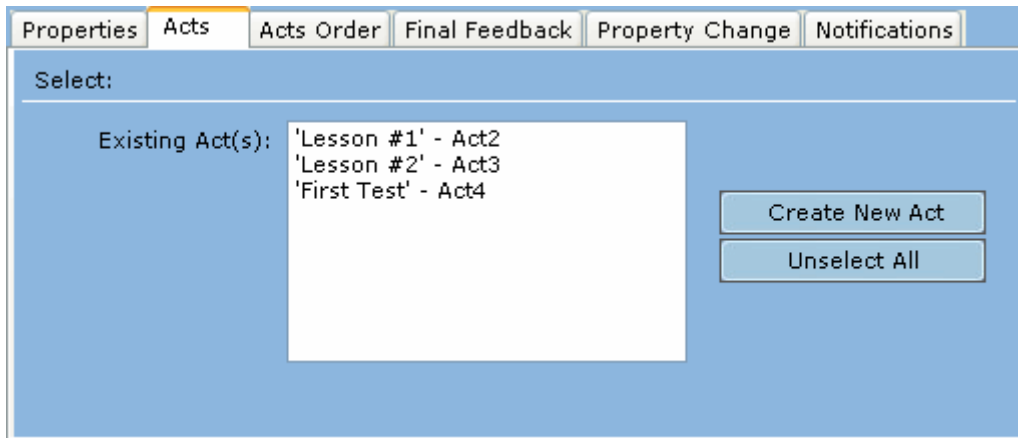
The screenshot shows a software interface with several tabs: 'Properties', 'Acts', 'Acts Order', 'Final Feedback', 'Property Change', and 'Notifications'. The 'Properties' tab is active. It contains a 'Title' text box, a checked 'Is Visible' checkbox, and a section titled 'Completed Play Options'. This section has four radio button options: 'None of Below options' (selected), 'Play has a Time Limit' (with input boxes for H, M, and S), 'Play is completed when last Act is completed', and 'Complete when set a Property Value' (with a 'Define' button). A 'Metadata' button is located to the right of the 'Completed Play Options' section.

The Properties fields are:

- **Title** – Represents the title that the user will give to the Play.  
This is a mandatory field, if this field is not filled the Play can't be created;

**Note:** Despite this value is not the ID of the Play (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing play or other kind of element. **According to this, try to put an explicit value on the title of the play to let you easily remember and identify the different plays;**

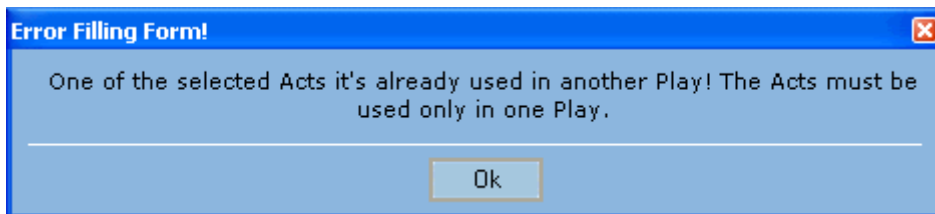
- **Is Visible** – Option value to define if this Learning Activity will be visible to the end user of the course;
- **Completed Play Options** – Option value to the user defines how can be considered the Play completed.  
The options are “Play has a Time Limit” and “Play is completed when last Act is completed”.  
In the “Play has a Time Limit” option the Play will be considered completed after the time defined on this value (sum of defined hours, minutes and seconds), **relative to the start of the course**. If this option is selected the user must insert a time value on the numeric fields (hours, minutes and seconds) in front of the option.  
In the “Play is completed when last Act is completed” option the play will be considered completed when the last act is completed.
- **Metadata** – Button that allows the user to define the Metadata associated with this Play. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Acts** – This is the list of Acts defined in the Course and where the user will define which of them belongs to this Play.



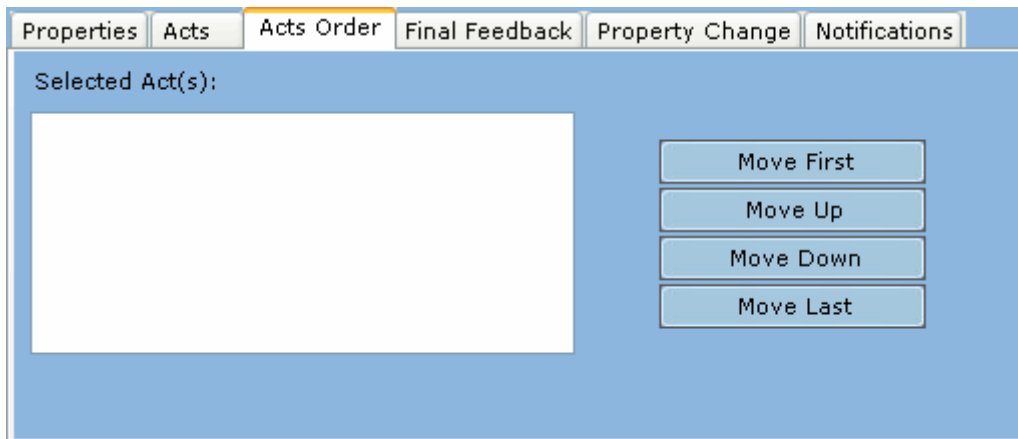
The Acts fields are:

- **Existing Act(s)** – List of all Acts defined inside of this course.  
The user must choose the Acts that will belong to this Play;
- **Create New Act** – Button that allows to the user creates a new Act that will belong to this Play and, due to that fact, will be automatically selected.  
For more information about the creation of an Act please read the “Acts” sub-module section of this manual;
- **Unselect All** – Button that allows to the user erases the selection made.  
The “Existing Act(s)” list will be with none selected element and the “Selected Act(s)” list of the “Acts Order” section will be empty;

**Note:** The acts can be only be used in one and only one Play! Due to this fact, when the user is going to save the Play and if he has selected an Act that is already selected in another Play, the Authoring Tool will present an error message like the below picture and the Play can't be created.

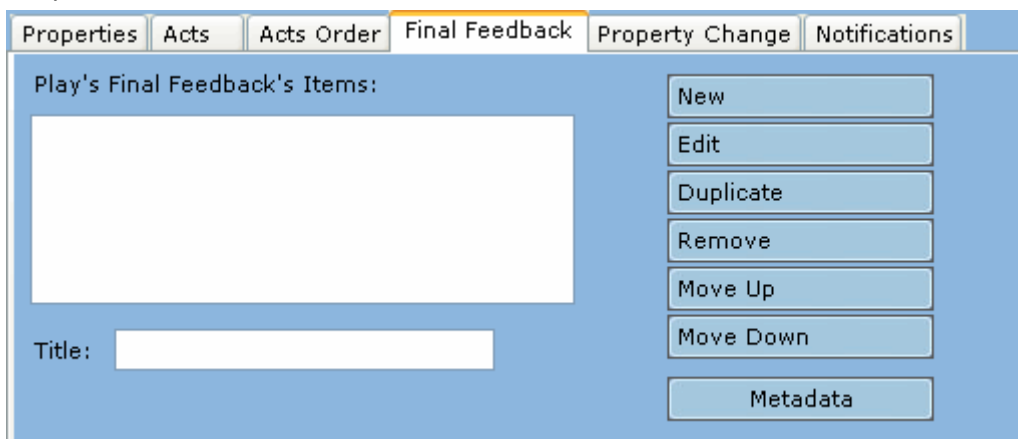


- **Acts Order** – This is the section where the user defines the order of the Acts selected in the “Acts” section.



The Acts Order fields are:

- **Selected Act(s)** – List of all Acts selected by the user in the previous section. This list has four related buttons which correspond, each one, to a specific operation about this list of selected Acts. Those operations are:
  - **Move First** – Move the selected Act to the first position of the list;
  - **Move Up** – Move the selected Act one position upper on the list;
  - **Move Down** – Move the selected Act one position lower on the list;
  - **Move Last** – Move the selected Act to the last position of the list.
- **Final Feedback** – This is a list of Items that will be shown to the end user when Play is completed.



The Final Feedback fields are:

- **Play's Final Feedback's Items** – List of Items that represents the Final Feedback of this Play. This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:
  - Create **New** Items;
  - **Edit** existing Items;
  - **Duplicate** existing Items;
  - **Remove** existing Items;
  - **Move Up** existing Items;
  - **Move Down** existing Items;



For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;

- **Title** – Text value to define a specific title for the Final Feedback of this Play.
- **Metadata** – Button that allows the user to define the Metadata associated with the Final Feedback of this Play. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

## 4.15 Acts

The Acts sub-module is the third option of the Method menu.



In the *IMS-LD* specification we can find the following description about Acts:

*“Play elements unfold in a series of one or more acts, **which are always run in sequence**. No act is made visible to the role/s that play a part in it until the previous act has completed. This factor can be used to synchronize the activities of those playing roles within the play.*

*The activities associated with the roles within a play may have complex sequencing, so it is possible to have a ‘one-act’ play which has internally complex sequencing, as long as the activities of the players do not need synchronizing (e.g., when the use of a learning resource is completed, all participants move to a discussion forum).*

(...)

*Although within an act there may be complex sequencing, and there may be only one active act, if there is more than one act within a play, the acts are run in sequence.*

*Thus acts can be used as synchronization points, either waiting for all players to finish before starting the next, or forcing an end when a certain number of players have finished, when terminated by a support staff member or under some other condition. All role participants in the next act may then start together at the same time, subject to their being simultaneously logged into the runtime service. When the last act is completed, the unit of learning is also complete.*

*An act brings together one or more role-parts. This is the mechanism that allows more than one role to perform at the same time. Therefore role-parts within an act are always run in parallel.”*

Role-parts and Parts are the same concept.

(...)

*Role-parts enable several users, playing the same or different roles, to participate in the same act. Each role-part associates exactly one role with exactly one type of activity (including the performance of another unit-of-learning and activity-structures), or with one environment (equivalent to an organization in Content Packaging). Multiple role-parts within one act are performed concurrently.*

*The same role can be associated with different activities or environments in different role-parts, and the same activity or environment can be associated with different roles in different role-parts. However, the same role may only be referenced once in the same act. If multiple activities or environments need to be associated for the same role an activity-structure or wrapper environment should be used.*

*When an act within a play is activated, all the role-parts in the act go ‘on-stage’ or become live. Depending on the implementation, players of the roles referenced by the role-parts may then have their associated activities (or environments) made visible in their ‘activity-tree’ and any content associated with the activity made accessible.”*

The below picture shows how is the form available to the user:

## Learning Editor: Acts Manager

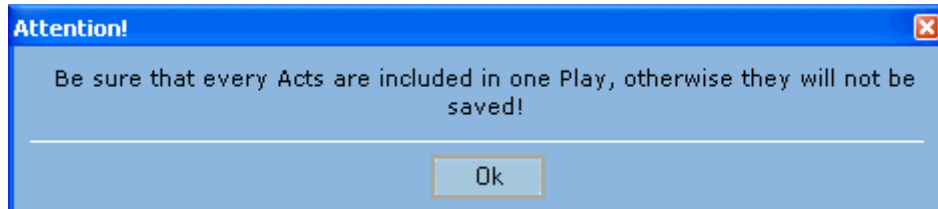


This feature allows the user to:

- Create **New** Acts – In this option the tool will present to the user a form to let the user fill the new Act data. Since this form needs an extensive explanation see the “Act Form” section of this sub-module;
- **Edit** existing Acts – In this option the tool will present to the user a form similar to the previous option but with the chosen Act data filled. The user can change all fields of that Act;
- **Duplicate** existing Acts – In this option the user can select an existing Act and create a copy of that Act. The only different values will be the ID and Title of the Act;
- **Remove** existing Acts – In this option the user can select and delete one existing Act;

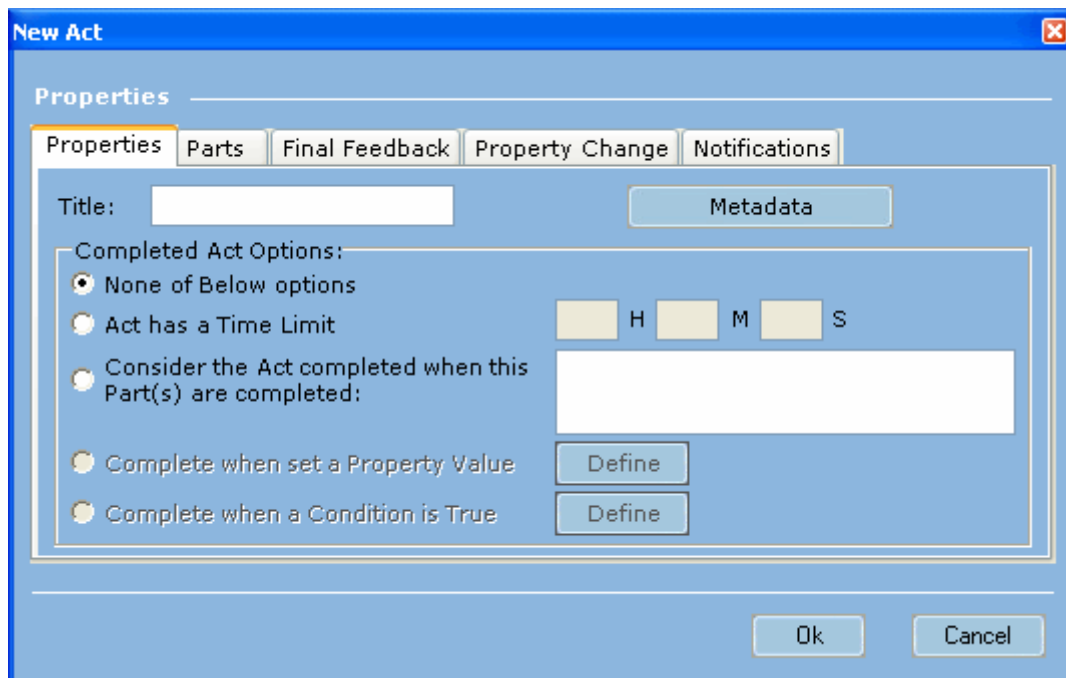
**Note: When you define an Act you need to associate it to a Play otherwise that Act will not be saved! This is due to the fact that Acts are an inner concept of the Plays, but the Authoring Tool organizes the Acts separately from the Plays to make the interface of the Plays simpler.**

**According to this fact, when the user is leaving the Method menu, the Authoring Tool launches an alert message to the user. The process is described in the below pictures.**



### 4.15.1 Act Form

The Act form available to the user is like the below picture.



This form is divided in five sections, which two of them (“Property Change” and “Notifications”) are related to Level B or C of *IMS-LD* specification. So they are inactive and don’t fall in the scope of this Authoring Tool version.

The sections that belong to Level A and are active are:

- **Properties** – These are the main properties of the Act.

The Properties fields are:

- **Title** – Represents the title that the user will give to the Act. This is a mandatory field, if this field is not filled the Act can't be created;

**Note:** Despite this value is not the ID of the Act (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing act or other kind of element. **According to this, try to put an explicit value on the title of the act to let you easily remember and identify the different acts;**

- **Metadata** – Button that allows the user to define the Metadata associated with this Act. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

- **Completed Act Options** – Option value to the user defines how can be considered the Act completed.

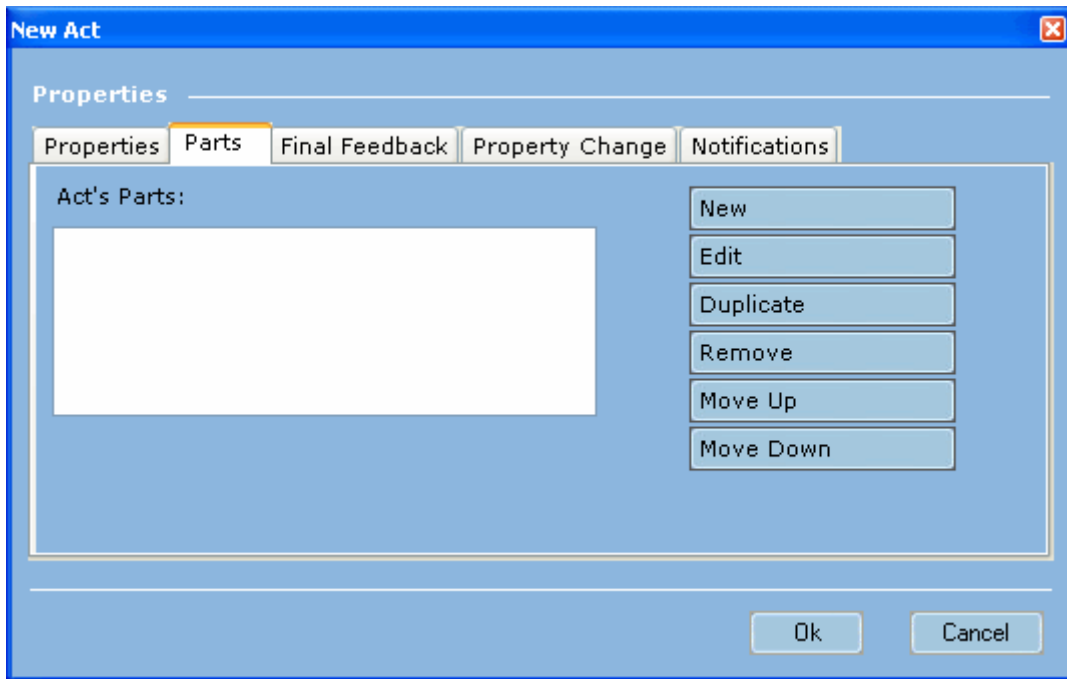
The options are “Act has a Time Limit” and “Consider the Act completed when this Part(s) are completed”.

In the “Act has a Time Limit” option the Act will be considered completed after the time defined on this value (sum of defined hours, minutes and seconds), **relative to the start of the course**. If this option is selected the user must insert a time value on the numeric fields (hours, minutes and seconds) in front of the option.

In the “Consider the Act completed when this Part(s) are completed” option the user defines that the course will be considered when a certain Part(s) is completed.

This is defined in the list in front of the option. The user will select which Part(s) correspond to this option. The list is of every Parts already defined in the Act. If the user selects more than one Part this can be seen that all those selected Parts need to be completed to consider the Act completed.

- **Parts** – This is the list of Parts defined in the Course.



In the Parts section the user can:

- Create **New** Parts – In this option the tool will present to the user a form to let the user fill the new Part data. Since this form needs an extensive explanation see the “Part’s Form” section of this sub-module;
- **Edit** existing Parts – In this option the tool will present to the user a form similar to the previous option but with the chosen Part data filled. The user can change all fields of that Part;
- **Duplicate** existing Parts – In this option the user can select an existing Part and create a copy of that Part. The only different values will be the ID and Title of the Part;
- **Remove** existing Parts – In this option the user can select and delete one existing Part;
- **Move Up** existing Parts – In this option the user can select and move up one position the selected Part in the list order;
- **Move Down** existing Parts – In this option the user can select and move down one position the selected Part in the list order.
- **Final Feedback** – This is a list of Items that will be shown to the end user when Act is completed.

The screenshot shows a software window with several tabs: 'Properties', 'Parts', 'Final Feedback' (selected), 'Property Change', and 'Notifications'. The main content area is titled 'Act's Final Feedback's Items:' and contains a large empty rectangular box. Below this box is a 'Title:' label followed by an empty input field. To the right of the main area is a vertical column of buttons: 'New', 'Edit', 'Duplicate', 'Remove', 'Move Up', 'Move Down', and 'Metadata'.

The Final Feedback fields are:

- **Act's Final Feedback's Items** – List of Items that represents the Final Feedback of this Act.
  - This list has six related buttons which correspond, each one to a specific operation about this list of Items. Those operations are:
    - Create **New** Items;
    - **Edit** existing Items;
    - **Duplicate** existing Items;
    - **Remove** existing Items;
    - **Move Up** existing Items;
    - **Move Down** existing Items;
  - For more information about the Items concept please read the “Items” section of the “Learning Editor sub-modules” section;
- **Title** – Text value to define a specific title for the Final Feedback of this Act.
- **Metadata** – Button that allows the user to define the Metadata associated with the Final Feedback of this Act. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;

#### 4.15.2 Part's Form

The Part (or Role-Part) concept can be seen as a sub-set of an Act.

In the *IMS-LD* specification we can find the following description about Parts:

*“An act brings together one or more role-parts. This is the mechanism that allows more than one role to perform at the same time. Therefore **role-parts within an act are always run in parallel.**”*

*Role-parts enable several users, playing the same or different roles, to participate in the same act. Each role-part associates exactly one role with exactly one type of activity (...) or with one environment (equivalent to an organization in Content Packaging). **Multiple role-parts within one act are performed concurrently.***

*The same role can be associated with different activities or environments in different role-parts, and the same activity or environment can be associated with different roles in different role-parts. However, **the same role may only be referenced once in the same act. If multiple activities or environments need to be associated for the same role an activity-structure or wrapper environment should be used.***

*When an act within a play is activated, all the role-parts in the act go ‘on-stage’ or become live. Depending on the implementation, players of the roles referenced by the role-parts may then have their associated*

activities (or environments) made visible in their 'activity-tree' and any content associated with the activity made accessible.”

The Parts are defined inside the Act's form in the “Parts” section but has a specific form to define the Part's data. This form is explained in this section.

The Part's form available to the user is like the below picture.

The Part's Form fields are:

- **Title** – Represents the title that the user will give to the Part. This is a mandatory field, if this field is not filled the Part can't be created;

**Note:** Despite this value is not the ID of the Part (the ID is created automatically by the tool), this is very important because the ID will be based on this value, except when that ID will be duplicated with some existing part or other kind of element. **According to this, try to put an explicit value on the title of the part to let you easily remember and identify the different parts;**

- **Related Role** – List of every defined Roles in the course in which the user must choose the Role related to this Part;
- **Metadata** – Button that allows the user to define the Metadata associated with this Part. Please see “Metadata Information Categories” section of the “Manage Global Metadata” module to see more information how to do this step;
- **Part references a** – Option value to the user define which kind of object is related to this Part. When the user chooses the type he must select on element of the correspondent list. In front to each option the user have a list of every defined objects of the related type. This list will be active only when the related option is selected. The available types are: “Learning Activity”, “Support Activity”, “Activity Structure” and “Environment”.



## Appendix 1 References

IMS Global Consortium. 2003. *IMS Learning Design Information Model. Version 1.0 Final Specification*. [http://www.imsglobal.org/learningdesign/ldv1p0/imslid\\_infov1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imslid_infov1p0.html)

IMS Global Consortium. 2003. *IMS Content Packaging Information Model. Version 1.1.3 Final Specification*. [http://www.imsglobal.org/content/packaging/cpv1p1p3/imscp\\_infov1p1p3.html](http://www.imsglobal.org/content/packaging/cpv1p1p3/imscp_infov1p1p3.html)

IMS Global Consortium. 2001. *IMS Learning Resource Meta-Data Information Model. Version 1.2.1 Final Specification*  
[http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd\\_infov1p2p1.html](http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_infov1p2p1.html)

IMS Global Consortium. 2001. *IMS Learning Resource Meta-Data Best Practice and Implementation Guide. Version 1.2.1 Final Specification*  
[http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd\\_bestv1p2p1.html](http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_bestv1p2p1.html)