

# **Common Cartridge Tutorial**

**An introduction to the Common Cartridge  
format**

**Common Cartridge Tutorial**  
**An introduction to the Common Cartridge format**



**Lars Ingesman, UNI-C, 2010**

**C:\Dokumenter\ASPECT\CC tutorial\Common Cartridge Tutorial.doc**

# Content

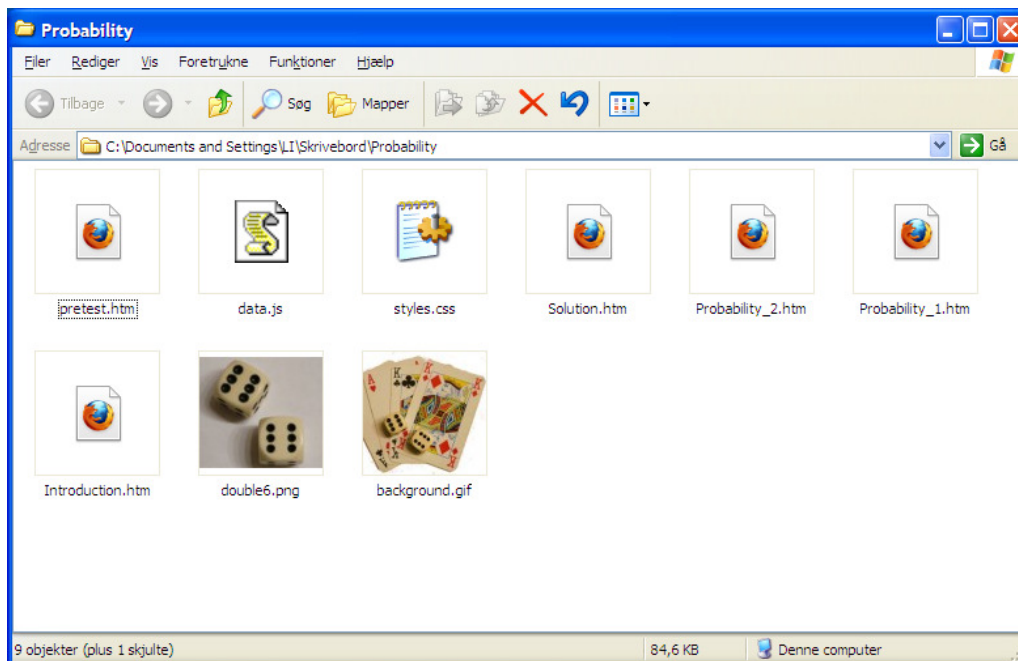
1	Introduction.....	1
2	A simple package .....	2
2.1	Creating the package .....	2
2.2	Uploading and running the package .....	4
3	Adding functionality .....	8
3.1	Types of CC functionality.....	8
3.2	Adding web links .....	8
3.3	Adding a discussion .....	9
3.4	Adding QTI.....	10
4	Conclusion.....	11

# 1 Introduction

Producing learning resources packaged in the Common Cartridge format can be done in a number of ways. In the current tutorial I will be using a simple drag-n-drop tool called the *Common Cartridge Builder* that can be downloaded from *Learning Components* website: <https://www.learningcomponents.com/apis.php>.

The tool requires Adobe AIR, so the first step is downloading and installing this application (<http://get.adobe.com/air/>). Once you have done this, you can install the Common Cartridge Builder and start creating Common Cartridge packages.

As it is a packaging tool only, you need to have your resources prepared beforehand. In the example I will show, I am using a set of HTML-pages that I have prepared beforehand.

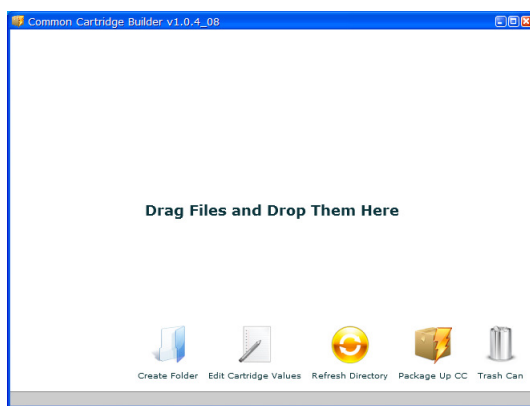


As you can see, I have some HTML-pages, a couple of images (.png and .gif), a stylesheet file (styles.css) and an external JavaScript file (data.js).

## 2 A simple package

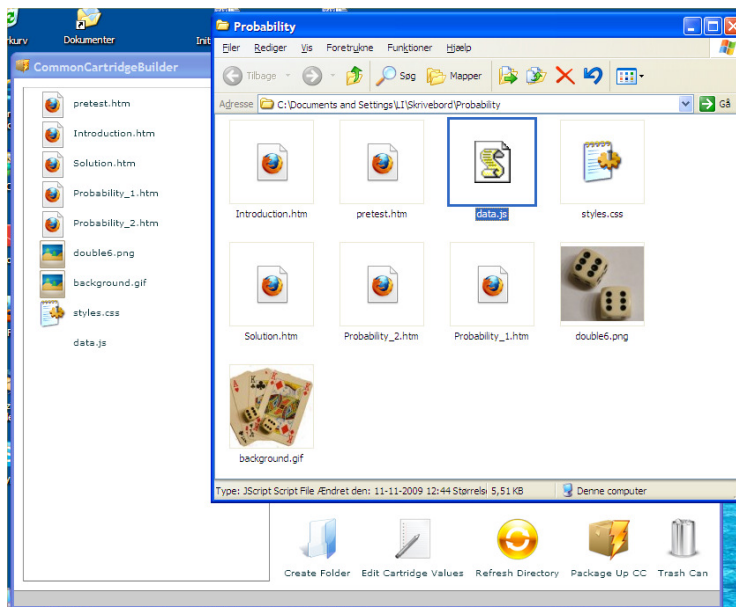
### 2.1 Creating the package

When I start the *Common Cartridge Builder*, all I get is a window with a few buttons in the lower part of the window and the text *Drag Files and Drop Them Here* Here:








Then I start dragging the files that will go into my package from the directory which I have placed into the *Common Cartridge Builder*.

I drag them into the *Common Cartridge Builder* window in the order in which I want them to appear in the package. I can rearrange the order afterwards if necessary, but it is easier doing it right the first time.







Now that I have all my resources in the CCBUILDER, I want to add some metadata about the package. To do this I first click the button *Edit Cartridge Values*, number two from the left. This opens a set of fields into which I can enter various kinds of information:

Filter





Property ▲	Value	
description		
id	LC_CCB_0	
keywords		
name	Untitled	
protect	false	





In order to edit the field and enter new values, I click the pencil in the right side of the field. When I have done that, I can start entering information into the field:

Property ▲	Value	
description	Math and probability	
id	LC_CCB_0	
keywords	math, simple rules, probability, Pascal, Fermat	
name	Math and the rules of probability	
protect	false	

In the example I have entered a description, some keywords and a name. I just leave the id as it is.

When I click one of the resources that I have dragged into the window, I get a set of fields that behave in the same way:

 pretest.htm	Filter <input type="text"/>
 Introduction.htm	
 Solution.htm	
 Probability_1.htm	

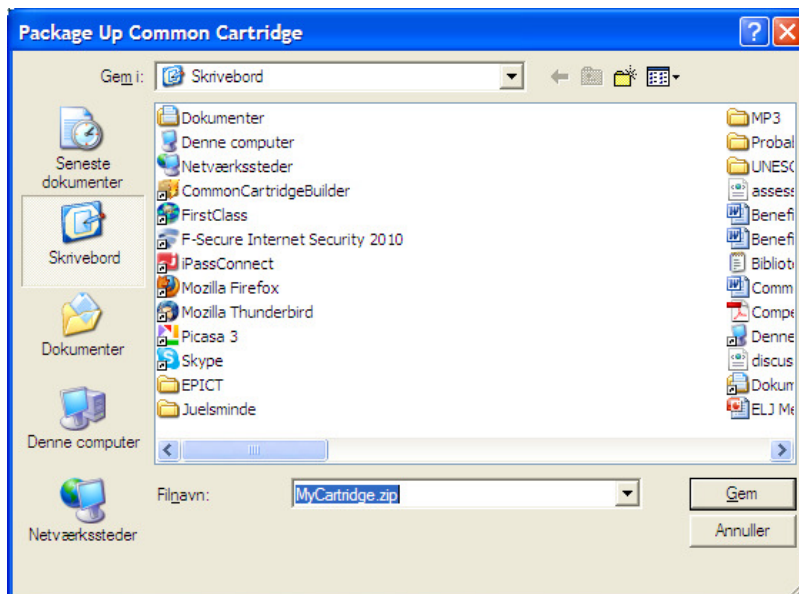
Property ▲	Value	
protect	false	
title	pretest	
type	Web Content	
visible	true	

In the example above I have clicked the resource *pretest.htm*, and now I can edit title, type and visibility of this element. For the time being we will just leave the values as they are, except with a few of the resources. I don't want the background

images, the stylesheet and the JavaScript file to crop up as a menu item in my package, so I set their visible property to *false*.

Property ▲	Value
protect	false
title	double6
type	Web Content
visible	false
	true
	false

This is basically all there is to it: now I am ready to click the button *Package up CC* and create the actual package.



After clicking the button I get a *Save file* dialog with the suggested name *MyCartridge.zip*. This I will change to something better such as *Probability.zip*.

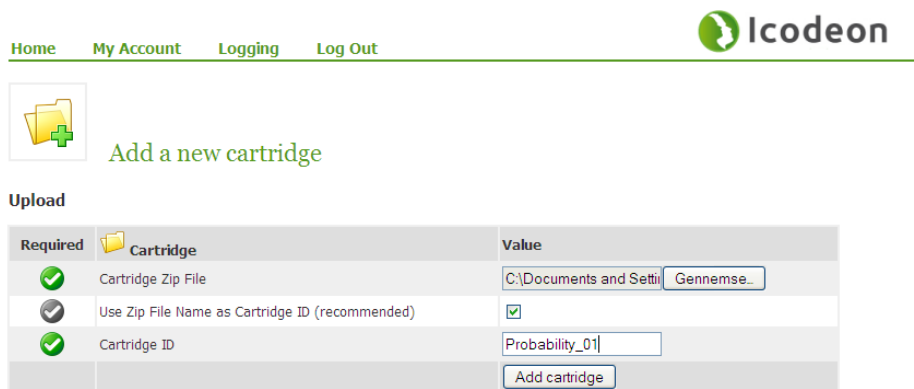
## 2.2 Uploading and running the package

If you don't have access to the Icodeon Common Cartridge platform that I am using, there are other options. You may use the *MIR Common Cartridge System* (<http://dev.learngauge.com/MCCR/>) or an up-to-date version of the Moodle LMS.

Now I'll upload the package to the Icodeon Common Cartridge platform and see what it looks like. I do not, however, close the *Common Cartridge Builder* yet, because I may need to edit something in the package, and at the moment the

*Common Cartridge Builder* cannot load an existing Common Cartridge zip-file. So once you close the window, you have to start all over again!

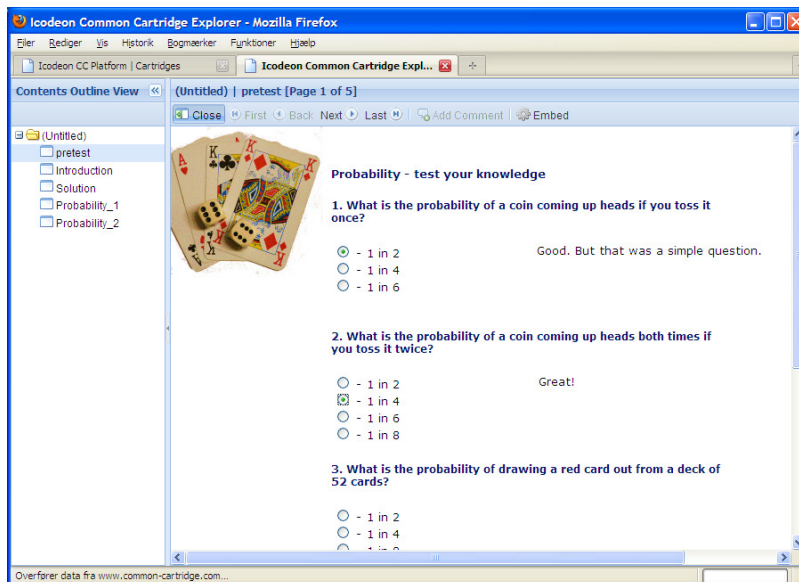
To add a new cartridge to my list of cartridges, I go to the *Add a new cartridge* page, locate my cartridge, and click the *Add cartridge* button. Because I am going to add more versions of this cartridge, I have added a *\_01* to the suggested Cartridge ID, so that it now is *Probability\_01*:



Required	Cartridge	Value
<input checked="" type="checkbox"/>	Cartridge Zip File	C:\Documents and Settings\Gennemse...
<input checked="" type="checkbox"/>	Use Zip File Name as Cartridge ID (recommended)	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Cartridge ID	Probability_01

Add cartridge

The cartridge now shows up on my list of cartridges. To see what it looks like when run, I simply click *Launch*.



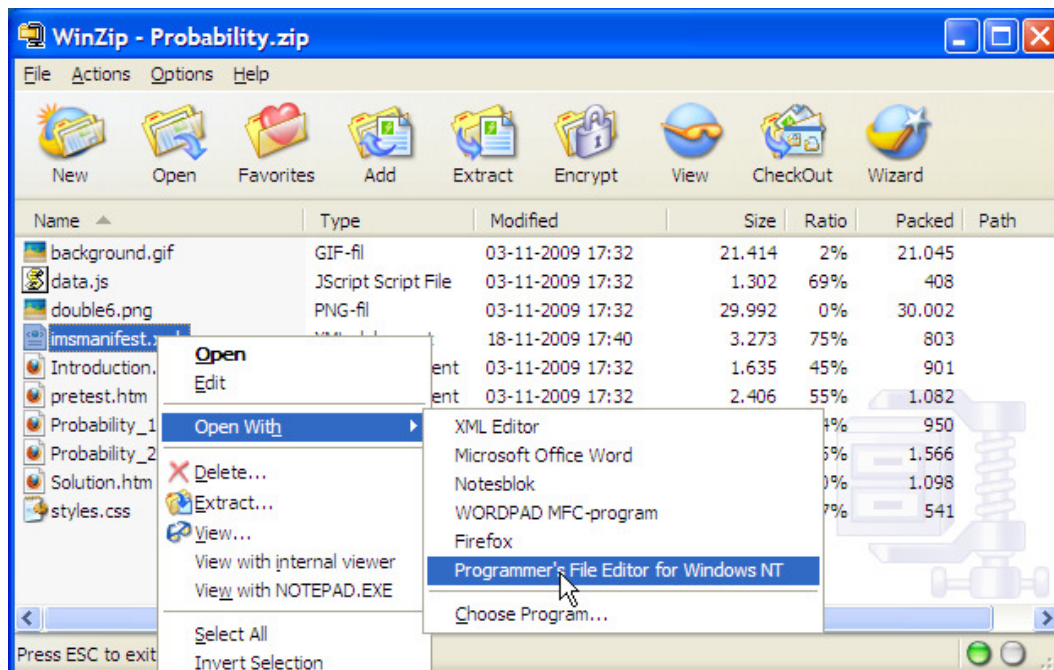
As you can see, we have the list of resources in the left-hand side of the window; in the right we have the JavaScript test that is listed as the first element in the package.



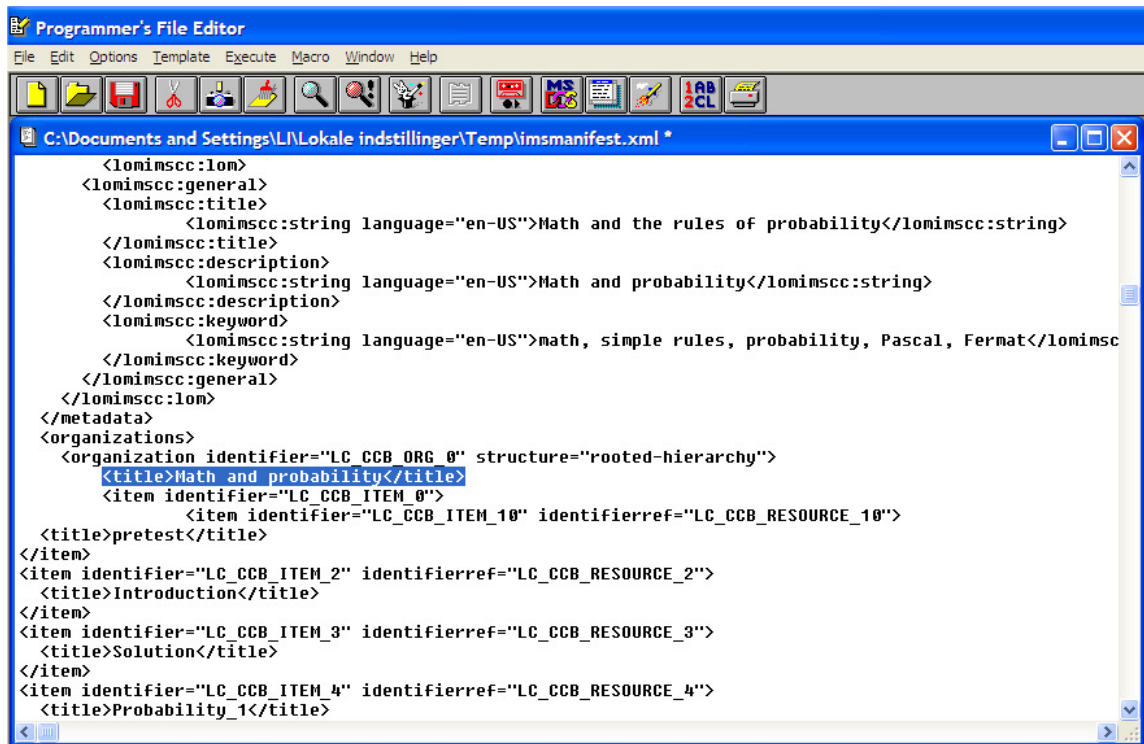
If I want to change an element in the package, all I need to do is make the necessary changes to the resource file, drag the copy already in *Common Cartridge Builder* window onto the *Trash can* and then drag the edited file into the *Common Cartridge Builder* window once more. You may have to click on one of the other elements to move focus away from the *Trash Can* before you are allowed to drag a new resource into the window (it is, after all, a beta version of tool...)

One thing about my package, though, is still not quite ok. The main package name is listed as *untitled*, although I did write a name in the *Common Cartridge Builder*. Apparently this gets thrown away in the building process. To fix this I'll just edit the *imsmanifest.xml*.

In order to edit the *imsmanifest* file, I double-click the zip-file to start Winzip, right-click *imsmanifest.xml* and select the editor of my choice to open the file with:



This opens the manifest file in *Programmer's file editor*, where I add the following line: `<title>Math and probability</title>` in the organizations part



Then I save the file, close my editor and return to Winzip. Here I confirm that I want to save the changes to the zipped package.

I could of course also have unzipped the entire package, done the editing and rezipped the package. The result is the same. Now my package displays the right name.

## 3 Adding functionality

### 3.1 Types of CC functionality

The Common Cartridge format support different types of functionality such as web links and discussions. Now we will add first a couple of web links and then a discussion.

### 3.2 Adding web links

There are two steps to adding a web link. The first involves creating a simple XML-file that describes the web link:

```
<?xml version="1.0" encoding="UTF-8"?>

<wl:webLink xmlns:wl="http://www.imsglobal.org/xsd/imswl_v1p0">



    <title>Different aspects of basic probability</title>

    <url href="http://www.probabilitytheory.info/" target="_self" window-
Features="width=640, height=800"/>

</wl:webLink>
```

As you can see, it is a very simple file. All you need to do is copy the example, change the title and url href parts and save it in the resource directory for your package.

Next step is dragging the XML-files into the *Common Cartridge Builder* to include them in the package – and because we are working with a different type of resource, we also need to change the resource type of our XML-file to *Web Link*:

Property ▲	Value
protect	false 
title	WL-exercises 
type	Web Content ▼
visible	Web Content ▲
	Associated Content
	Discussion Topic
	Web Link
	QTI Assessment ▼

So for all our web link resources we need to change the type. If we do not change the type from Web Content to Web Link, the Common Cartridge platform will simply display the XML inside the file instead of setting up a web link.

Once we have this for all web link resources, we click the package button again, upload the new version to the Common Cartridge platform and run it. And now we do have web links in our menu structure.

### 3.3 Adding a discussion

Adding a discussion is done in the same way. First we create a small XML-file with the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>

<dt:topic xmlns:dt="http://www.imsglobal.org/xsd/imsdt_v1p0">

  <title>Probability discussions</title>

  <text texttype="text/html">Use this discussion for posing questions
or puzzles related to probability. You may e.g. ask your classmates to
solve probability questions.</text>

</dt:topic>
```

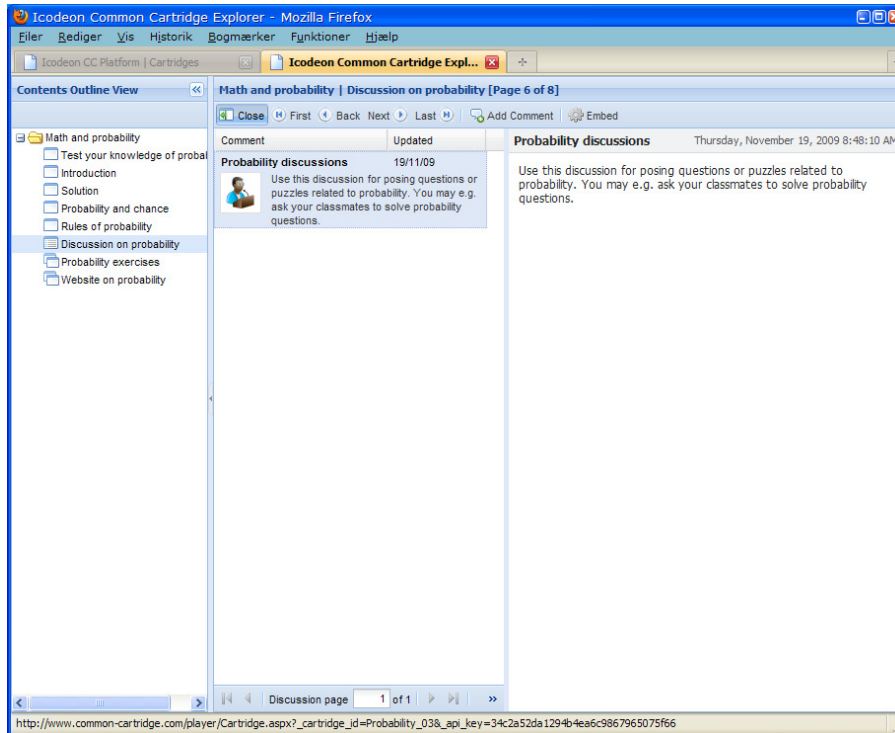
Again we have a few fields that we need to change to make the discussion reflect the position and role in the package and learning process. We provide a title that will be shown in the menu along with the other resources in the package. And then we provide a short introduction to the discussion so that the learners will know what to use it for.

Next we drag the discussion into the *Common Cartridge Builder* and change the type of the element to *Discussion Topic*:

Property ▲	Value
protect	false
title	discussion
type	Web Content ▼
visible	Web Content ▲
	Associated Content
	Discussion Topic
	Web Link
	QTI Assessment ▼

We can also change the title of the discussion to something better than simply the filename, e.g. *Discussion on probability*.

Finally we create a new version of the package and upload it to the Common Cartridge platform. When we run the new version of the package, we have three new menu items: one discussion and two web links. Below you can see what a discussion looks like on the Icodeon CC platform:



This is the basic procedure when adding some of the new Common Cartridge specific functionality to our package.

### 3.4 Adding QTI

Finally, I'll show how to add a simple QTI question set to my package. The questions are defined in an XML-file; to add the questions to our package we simply drag this XML-file into our package and change the type to *QTI Assessment*. Then we create a new version of the package, upload it and run it. That is all there is to it on the packaging side. The big task is creating the QTI file with good questions.

## 4 Conclusion

I have demonstrated how to build simple packages using the *Common Cartridge Builder*, which is a nice and easy-to-use tool. It is a great tool for exploring Common Cartridge packaging.

Once you have understood the basics of the *imsmanifest.xml*, it is fairly easy to set up a more automated workflow, depending on how you have your resources stored.

In the examples I have used standard HTML-pages, a couple of images (.png and .gif) inserted both as background images through the stylesheet and as normal images in the HTML, an external stylesheet file (styles.css) and an external JavaScript file (data.js). The idea was to see if the package format and the Icodeon Common Cartridge platform would allow me to do what I want to do. So far the answer is yes: I can create complex web-based learning that I package and distribute as Common Cartridge packages.