


# **OSLab Module Author Guide**

Attila Weyland and Torsten Braun

IAM-06-004

July 25, 2006



	<p style="text-align: center;"><b>Operating System Laboratory</b></p> <p style="text-align: center;">Module Author Guide</p>	<p style="text-align: right;">v 1.01 July 25, 2006</p>
---	--	--

# Module Author Guide<sup>\*</sup>

Attila Weyland and Torsten Braun

Institute of Computer Science and Applied Mathematics  
University of Bern  
Neubrückstrasse 10, 3012 Bern, Switzerland  
Email: [weyland@iam.unibe.ch](mailto:weyland@iam.unibe.ch)

## Abstract

OSLab is the third generation in a series of distance learning courses established by mostly Swiss and some international universities. The underlying didactics have been developed in the VITELS project and written down in a didactics and design guide. The guide has been further extended in the EuQoS project. This document tries to reduce the guide to a reasonable size and incorporate the experiences made during the operation of the VITELS course.

In the creation process of a learning course, we identified two possible roles: module authors and course authors. A module author creates a learning module. A course author defines the general layout and structure of the course as well as the modules. This document guides you through the process of creating a learning module for the Operating System Laboratory and therefore addresses module authors.

## CR Categories and Subject Descriptors


[K.3.1 Computer Uses in Education]: Distance learning; [K.3.2 Computer and Information Science Education]: Computer science education

## General Terms

Documentation, Design, Management

---

<sup>\*</sup> Thanks to the co-authors of the VITELS Didactics and Design Guide, who laid the foundation for this work. Thanks to Gerald Wagenknecht for comments and suggestions.

	<b>Operating System Laboratory</b> Module Author Guide	v 1.01 25 July 2006
---	---	------------------------

## Table of Contents

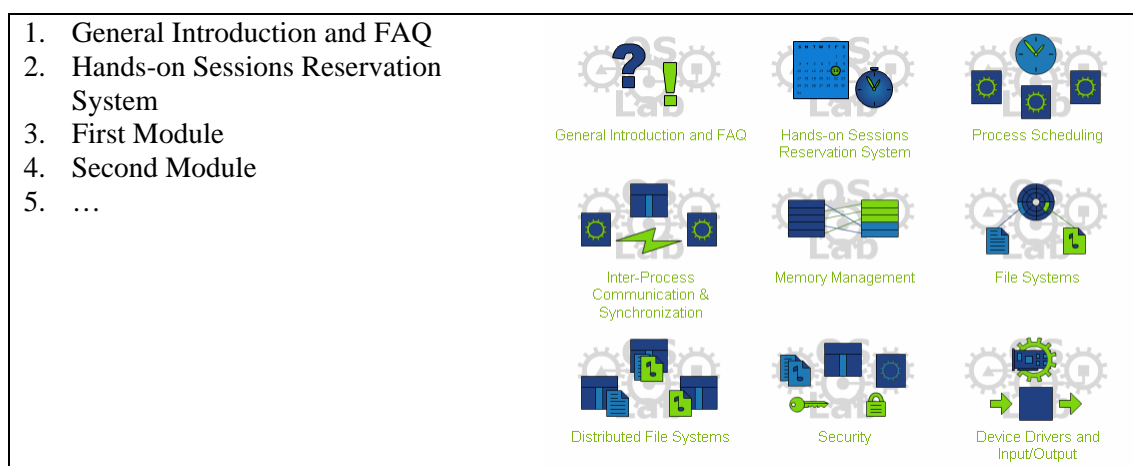
1	Introduction.....	3
1.1	Course Structure.....	3
1.2	Module Structure .....	3
1.3	Learning Tools .....	4
1.4	Implementation Procedure .....	5
2	OSLab Development Bundle .....	6
2.1	Installation.....	6
2.2	Usage.....	7
2.2.1	GCED & FFGF .....	7
2.2.2	FFGF .....	8
2.2.3	Generate PDF .....	8
3	Design .....	9
3.1	Rules .....	9
3.2	HTML Elements .....	9
3.3	Format Options .....	10
4	Content Creation .....	12
4.1	Introduction.....	12
4.1.1	Welcome .....	12
4.1.2	The Goals and How to Reach Them .....	12
4.1.3	Module Vicinity .....	13
4.1.4	My Goals.....	14
4.1.5	Tips .....	14
4.1.6	FAQ.....	14
4.2	Theory .....	14
4.2.1	Theoretical Basics .....	14
4.2.2	Readings.....	15
4.2.3	Self Test .....	16
4.2.4	Quiz.....	16
4.3	Knowledge Application/Exploration .....	16
4.3.1	Introduction.....	17
4.3.2	Hands-on Session.....	17
4.4	Prove Your Knowledge and Skills.....	17
4.4.1	Personal Synthesis .....	18
4.4.2	Final Quiz.....	18
4.4.3	Survey .....	18
5	Module Integration.....	18
5.1	WebCT Access.....	18
5.2	Content Subscription.....	19
5.3	Learning Tools Application .....	21
5.3.1	Discussions .....	22
5.3.2	Common Assessments .....	22
5.3.3	Specific Assessments .....	23
5.3.4	Link Learning Tools and Content .....	24
6	References.....	26

# 1 Introduction

OSLab is based on a constructivist learning approach, which is well suited for hands-on focused courses. This section motivates the didactical concepts and explains the main components, such as the specified course and module structure and the employed learning tools.

## 1.1 Course Structure

The OSLab course structure is based on highly modularized learning content according to subjects and can be consumed according to the learner's need. Figure 1 shows the course structure.



**Figure 1: Course Structure**

The introduction module (General Introduction and FAQ) explain the course objective, didactic concept, and so forth. The link to the laboratory reservation system (Hands-on Sessions Reservation System) allows the learner to book time slots for hands-on sessions using real hardware. Then follow the different learning modules. We describe the structure of the learning modules in the next section.

## 1.2 Module Structure

Each module consists of four chapters: introduction, theory, knowledge application/ exploration as well as evaluation of acquired knowledge and skills. The first chapter "Introduction" motivates the module content and places it into context with other modules of the course. In the second chapter "Theory", the learner acquires the knowledge to solve the practical exercises. In the third chapter "Knowledge Application/ Exploration", the learner solves programming or configuration tasks in a simulated, emulated or real environment. The fourth chapter "Prove Your Knowledge and Skills" evaluates the acquired knowledge of the learner. Figure 2 shows the module structure.

- |   |
|---|
| <ol style="list-style-type: none"> <li>1. Introduction <ol style="list-style-type: none"> <li>1.1. Welcome</li> <li>1.2. The Goals and How to Reach Them</li> <li>1.3. Module Vicinity</li> </ol> </li> </ol> |
|---|

- 1.4. My Goals
- 1.5. Tips
- 1.6. FAQ
- 2. Theory
  - 2.1. Theoretical Basics
  - 2.2. Readings
  - 2.3. Self Test
  - 2.4. Quiz
- 3. Knowledge Application/Exploration
  - 3.1. Introduction
  - 3.2. Hands-on Session
- 4. Prove Your Knowledge and Skills
  - 4.1. Personal synthesis
  - 4.2. Final Quiz
  - 4.3. Survey

**Figure 2: Module Structure**

The module structure is almost fixed. That means no changes are allowed except for three points and only starting from the third-level. Therefore, you can only add sub-sections to “2.1 Theoretical Basics”, “3.1 Introduction” and “3.2 Hands-on Session”. However, you have to create content for almost all of these predefined sections.

### 1.3 Learning Tools


We use different tools to keep track of the learner’s progress. These tools are available at different stages of the learner’s path throughout a module and graded according to their purpose. Table 1 gives an overview on the tools and the achievable points.

**Table 1: Point Distribution**

Chapter				Evaluation Tools	Points	Share
1	2	3	4			
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discussion	0	0%
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Logbook	25	18%
<input type="radio"/>				My Goals	20	14%
	<input type="radio"/>			Self Test	0	0%
	<input type="radio"/>			Quiz	30	21%
		<input type="radio"/>		Personal Synthesis	20	14%
		<input type="radio"/>		Final Quiz	40	29%
		<input type="radio"/>		Surveys	5	4%
				<b>Total</b>	<b>140</b>	<b>100%</b>
				Traditional	70	50%
				Modern	70	50%

We distinguish between traditional and *modern* tools for learning. Traditional tools are known from offline courses. Modern tools take advantage of the new possibilities offered by an online infrastructure.

The traditional tools consist of self test, quiz and final quiz. The self test allows the learner to determine his knowledge state without being graded. In the quiz, the learner has to prove his knowledge acquired in the theory chapter. With the final quiz the

	<p style="text-align: center;"><b>Operating System Laboratory</b></p> <p style="text-align: center;">Module Author Guide</p>	<p style="text-align: right;">v 1.01 25 July 2006</p>
---	--	---

learner summarizes his results obtained by performing the practical tasks from the hands-on chapter.

The modern tools consist of discussion, logbook, my goals, personal synthesis and survey. The discussion forum helps learners to exchange knowledge and obtain support from other learners or the tutor(s). The logbook helps the learner to keep track of his learning progress and note things for future reference. The discussion and logbook are available throughout the whole module. The section “My Goals” requires the learner to formulate his learning goals based on the introduction and thereby activates prior knowledge. The personal synthesis helps the learner in structuring the acquired knowledge from the theory and the hands-on chapter. It also allows the comparison of the initially specified and the finally achieved goals. The survey helps the tutor to adapt and improve the module.

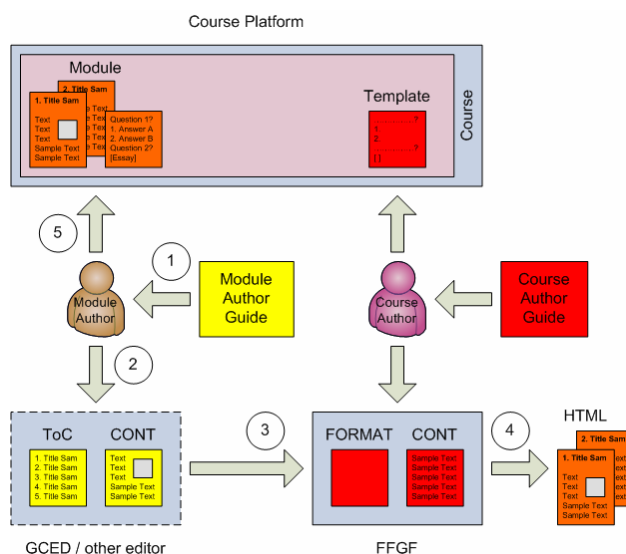
### ***1.4 Implementation Procedure***

In this section, we describe the common procedure of implementing the presented course and module structure. As mentioned earlier, we identified two roles in the creation process of a learning course: course and module author.

The module author is solely responsible for the creation of content for his learning module(s). The course author is responsible for the maintenance of the overall course design, the predefined text of the learning modules and the initialization module. In addition he has to prepare the course in the course platform, i.e. creating it, establishing the common design as well as creating the template for the learning tools. Last, the course author has to integrate the initialization module.

To help the module author concentrate on the task of writing and illustrating the topic, we separated the creation of content and the layout, while a tool merges this content into a predefined framework. The tool is called file framework generator & formatter, ffgf [4]. It takes a table of contents and the according content files (both written by a module author) as well as the formatting information and predefined content (specified by the course author) as input. From this, the tool generates formatted HTML [5] pages as output. The layout is based on Cascading Style Sheets, CSS [6] and invisible tables.

To support the module author in the creation of content, a graphical editor has been developed, which is called Guided Content Editor, GCED [7]. The GCED is a Java-based WYSIWYG editor, which provides basic HTML formatting functionality. It also helps in the specification of a valid table of contents for the ffgf. Both, ffgf and GCED are available in a development bundle [8]. Figure 3 illustrates the interaction of the different components in the content creation process.



**Figure 3: Course Implementation Process**

Initially the course author specifies the format and the predefined content for the ffgf as well as prepares the course inside the course platform in accordance with the course author guide [9]. The module author specifies the table of contents and creates the module content (2) in accordance with the module author guide (this document) [10] (1) and by using the GCED. Next, the GCED invokes the ffgf (3), which finally creates the resulting HTML pages (4). Last the module author integrates the module into the course platform (5). It is possible to use a different editor and directly invoke the ffgf. Section 2 explains both possibilities in detail.

## 2 OSLab Development Bundle

The OSLab Development Bundle, ODB [8] is an archive which includes the Guided Content Editor, GCED, the file framework generator & formatter, ffgf as well as some graphics. FFGF requires Perl [11]. The script to generate a PDF file from the HTML pages requires HTMLDOC [12].


### 2.1 Installation

Unzip the ODB archive into a directory of your choice. After unpacking the archive, you find the following main directory structure:

```
doc/
gced_ffgf/
gced_ffgf/etc/
gced_ffgf/input/
gced_ffgf/output/content
gced_ffgf/output/htmlCE
gced_ffgf/output/htmlVista
graphics/
pdf/
```

```
Module and Course Author Guide, GCED
GCED JAR, FFGF start script, README
configuration and template files
input files (Table of Contents)
generated content files
generated HTML files for WebCT CE
generated HTML files for WebCT Vista
OSLab (module) logos, position map
PDF generation script based on htmldoc
```



	<p style="text-align: center;"><b>Operating System Laboratory</b></p> <p style="text-align: center;">Module Author Guide</p>	<p style="text-align: right;">v 1.01 25 July 2006</p>
---	--	---


## 2.2 Usage

There exist two possibilities for creating content. The first is to use the GCED, which provides a nice interface to modify the table of contents file and to write to the respective content files. It also invokes the ffgf automatically upon changes. The second is to directly edit these files using another editor of your choice. The GCED is based on the Java HTML EditorKit [13], which exposes some weird behavior. If you have difficulties in adjusting, the second option might be your choice. In the following two sections, we describe both possibilities. In the third section, we describe how to generate PDF with the provided script.

### 2.2.1 GCED & FFGF

The next steps give an overview on how to create a learning module using the GCED. The given references lead to the detailed descriptions and must be followed. The documentation for the GCED can be found in `doc/Introduction_to_GCED.pdf`.

1. Change into the directory `gced_ffgf`.
2. Create your module's table of contents file in the directory `gced_ffgf/input` from the file `Template.txt`, e.g. copy `Template.txt` to "`Module Name.txt`". You can adjust the table of contents within GCED.
3. In case this is your first time using this table of contents,
  - a. Start the ffgf: `perl ffgf.pl`.
  - b. Select the table of contents file of your module from the list.
  - c. If requested enter the module author(s) and/or the module name (these values are saved in a `gced_ffgf/input/*.stat` file).
  - d. Quit the ffgf by entering "q" in the "Options" menu.
4. Start the GCED: `java -jar GCED-*version*.jar`.
5. In case this is the first time starting GCED, you will be asked to specify the location of the FFGF.
6. Open your module via the menu "Project->Open", choose from the drop down list the respective module, GCED will display a table of contents on the left and the concatenation of all content file in the middle.
7. Adjust the table of contents from the menu "Project->Edit structure". The red entries indicate the places where you can add files below the current level. Adapt the duration of the chapter entries. The rest is not editable. Click "Update Project" to commit the changes.
8. Write your module content to the files based on Section 4 using the elements and format options from Sections 3.2 and 3.3 respectively and stay in compliance with the rules defined in Section 3.1.
9. Save your project and zip the output for your module located under directory `htmlVista/module_name` to `module_name.zip`. Don't include the base directory, i.e. your root in the zip must contain the directories 1 2 3 4 doc pictures. WebCT Vista will create a folder based on your zip archive name upon unzipping.
10. Integrate your module into WebCT Vista as described in Section 5.

	<p style="text-align: center;"><b>Operating System Laboratory</b></p> <p style="text-align: center;">Module Author Guide</p>	<p style="text-align: right;">v 1.01 25 July 2006</p>
---	--	---

### 2.2.2 FFGF

The following steps describe how to create your learning module relying solely on the ffgf. The given references lead to the detailed descriptions and must be followed. The ffgf offers text-based menu navigation. It calls external programs to for the editing or display functionality. The external programs to be used can be configured in `gced_ffgf/etc/ffgf.conf`.

1. Change into the directory `gced_ffgf`.
2. Create your module's table of contents file in the directory `gced_ffgf/input` from the file `Template.txt`, e.g. copy `Template.txt` to "`Module Name.txt`" and modify the content accordingly.
3. Start ffgf: `perl ffgf.pl`.
4. Select the table of contents file of your module from the list (You can skip this step by passing the table of contents file name directly when calling the script, e.g. `perl ffgf.pl input/Module Name.txt`).
5. In case this is your first time using ffgf
  - a. enter the module author(s) and/or the module name (these values are saved in a `gced_ffgf/input/*.stat` file)
6. Change to the "Editing/Viewing" menu by entering "e" in the "Options" menu.
7. Adjust the table of contents file by entering "T" in the "Editing/Viewing" menu. Note that you are only allowed to add subsections below Section 2.1, 3.1 and 3.2. As soon as you leave the editor, ffgf checks for changes and updates the files accordingly.
8. You can access every content file from the "Editing/Viewing" menu. Write your module content to the files based on Section 4 using the elements and format options from Sections 3.2 and 3.3 respectively and stay in compliance with the rules defined in Section 3.1.
9. You can view the result by entering "V" in the "Editing/Viewing" menu.
10. Save your project and zip the output for your module located under directory `htmlVista/module_name` to `module_name.zip`. Don't include the base directory, i.e. your root in the zip must contain the directories 1 2 3 4 doc pictures. WebCT Vista will create a folder based on your zip archive name upon unzipping.
11. Integrate your module into WebCT Vista as described in Section 5.

### 2.2.3 Generate PDF

The next steps describe how to create a PDF file from the HTML pages generated by the ffgf. The script requires HTMLDOC [12] to be installed. Only the source code version is available free of charge. The script calls HTMLDOC with `gced_ffgf/output/htmlVista/Your Module/all/Your Module.html` as input and creates `gced_ffgf/output/htmlVista/Your Module/doc/Your Module.pdf` as output. Therefore, it is important that you first run ffgf and generate the output structure for your module.

1. Change into the directory `pdf`.

2. Create your module's title file in the same directory from the file `Template.title`, e.g. copy `Template.title` to "`Module Name.title`" and modify the content accordingly.
3. Start the script: `./generate_pdf "Module Name"`

## 3 Design

The highly modularized content provided by different partners requires the specification of a uniform look and feel for each learning module. This section presents some design rules, which you must follow, when creating module content. In addition, we describe the available elements and format options.

### 3.1 Rules

When you prepare your module's content you must follow certain rules of good design.

1. Keep your content short and simple
2. Limit the numeration hierarchy of the Table of Contents entries (recommended depth of 3, maximum 4 numbers)
3. Limit the number of different font families and formats per page (maximum 3)
4. Limit the amount of different colors per page (maximum 5)
5. Reuse font formats and colors in the same context

Especially when creating graphics or animations try to apply these rules:

1. Respect the maximum width of 540 pixel in your graphic/animation
2. Use color selective (emphasize key elements, keep the surroundings black/white/grey)
3. Use light, appealing colors (not too dark or somber)
4. Limit the amount of gradients
5. Use the JPEG format for real-life images (photographs, paintings, etc.) and the PNG format for computer generated graphics (charts, flows, diagrams etc.). Do NOT use the GIF format!

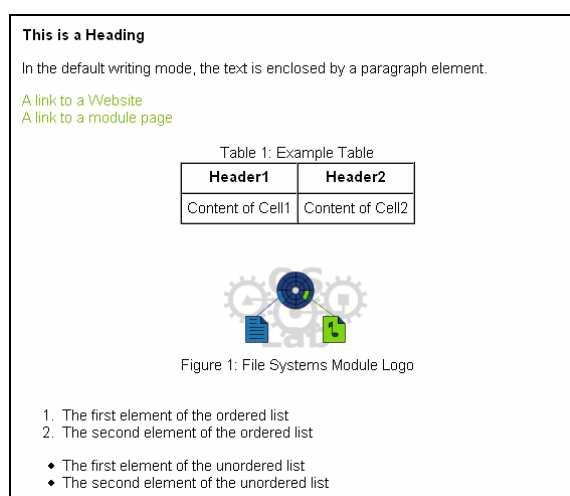
### 3.2 HTML Elements

We reduced the number of available HTML elements to a reasonable size. Table 2 describes the available GCED elements and their HTML counterpart. The GCED provides access to these different elements via the "Inset" menu. Normal text is placed within paragraphs. Figure 4 shows a rendered example of the elements and Listing 1 contains the corresponding HTML code.

**Table 2: Elements**

GCED Element	HTML element	Description
normal text	<p>	In the default writing mode, the text is placed within a paragraph element.
Header	<h3>	Formats the selected text as header. <u>Important:</u> A heading must be formatted after some text has been written below.
Hyperlink	<a>	Transforms the selected text into a hyperlink pointing to a

		given address.
Table	<code>&lt;table&gt;</code>	Inserts a table with the specified number of rows and cells.
Picture	<code>&lt;img&gt;</code>	Inserts a picture from the module's picture repository. The GCED offers a menu entry to manage the pictures of a module. <u>Important:</u> A picture can only be added inside a paragraph. It is currently not possible to add Flash animations via the GCED. Instead, you have to add them manually.
Ordered List	<code>&lt;ol&gt;</code>	Creates an ordered list and inserts the first element.
Ordered List Item	<code>&lt;li&gt;</code>	Adds an additional element to the ordered list.
Unordered List	<code>&lt;ul&gt;</code>	Creates an unordered list and inserts the first element.
Unordered List Item	<code>&lt;li&gt;</code>	Adds an additional element to the unordered list.



**Figure 4: HTML Elements Example**

**Listing 1: HTML Elements Example Source Code**

```
<h3>This is a Heading</h3>
<p>In the default writing mode, the text is enclosed by a paragraph element.</p>
<a href="http://www.oslab.ch/" target="_blank">A link to a Website</a><br/>
<a href="http://2/2_4_0_quiz.html">A link to a module page</a><br/><br/>
<div class="center">Table 1: Example Table</div>
<table class="center" border="1" cellpadding="5" cellspacing="0">
  <tr><th>Header1</th><th>Header2</th></tr>
  <tr><td>Content of Cell1</td><td>Content of Cell2</td></tr>
</table>
</div>
<br/><br/>
<div class="center">
<br/>
Figure 1: File Systems Module Logo
</div>
<br/>
<ol>
  <li>The first element of the ordered list</li>
  <li>The second element of the ordered list</li>
</ol>
<ul>
  <li>The first element of the unordered list</li>
  <li>The second element of the unordered list</li>
</ul>
```

### 3.3 Format Options

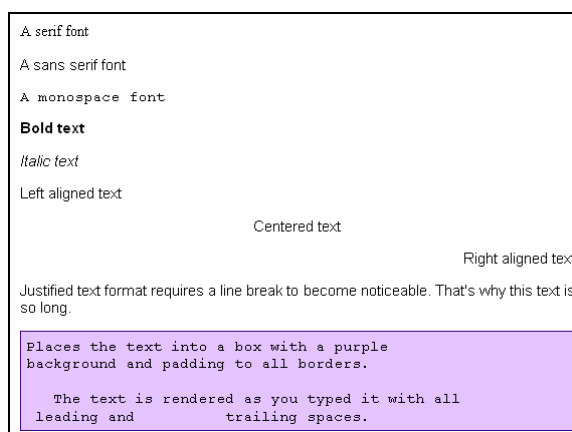
In order ensure a common look, we defined a set of basic format options, to which you must stick when creating content. The GCED provides access to these different styles via the Format menu. From there you can change the font family and style, the text alignment as well as emphasize distinguished text parts. These formats can also

be directly used as Cascading Style Sheet classes in HTML code. Table 3 presents the available GCED format options, the corresponding CSS class names and descriptions.

**Table 3: Format Options**

Format (Default)	Options	CSS Class Name	Description
Font Family (Sans Serif)	Serif	serif	Uses a serif font
	Sans Serif	sansSerif	Uses a sans serif font
	Monospace	monospace	Uses a monospace font
Font Style (Normal)	Normal		Normal text
	Bold	bold	Uses a bold typeface
	Italic	italic	Uses an italic typeface
Text Alignment (Left)	Left	left	Aligns the text to the left
	Center	center	Centers the text
	Right	right	Aligns the text to the right
	Justify	justify	Justifies the text
Code		code	Places the text into a box with a purple background and padding to all borders. The text is rendered as you typed it with all leading and trailing spaces.

The font family and style as well as the text alignment options are valid for text-based HTML elements (p, div, span, table, ol, ul, and so on). When you need a paragraph header, use the h3 element. If you want to display code snippets, use the code option. Figure 5 illustrates a rendered example and Listing 2 presents the corresponding source code.



**Figure 5: Format Option Examples**

**Listing 2: Format Option Examples Source Code**

```
<p class="serif">A serif font</p>
<p class="sansSerif">A sans serif font</p>
<p class="monospace">A monospace font</p>
<p class="bold">Bold text</p>
<p class="italic">Italic text</p>
<p class="left">Left aligned text</p>
<p class="center">Centered text</p>
<p class="right">Right aligned text</p>
<p class="justify">
Justified text format requires a line break to become
noticeable. That's why this text is so long.
</p>
<p class="code">Places the text into a box with a purple
background and padding to all borders.
  The text is rendered as you typed it with all
leading and      trailing spaces.
</p>
```

<p>The text is rendered as you typed it with all leading and trailing spaces.</p> <p>&lt;/p&gt;</p>
---

## 4 Content Creation

In the following, we describe the necessary content of the four chapters with their sections, which has to be created by the module author. Some sections or paragraphs are generated automatically during every run of the ffgf (initiated by the GCED or manually). These sections or paragraphs are emphasized accordingly.

### 4.1 Introduction

The introduction chapter consists of six sections. The goal is to introduce the current module to the learner.

#### 4.1.1 Welcome

In the welcome section you summarize the learning content, define the module requirements and mention the responsible person(s) of the module content. This information is contained in three paragraphs:

**Abstract:** Write a welcome message and a 10 lines introduction (abstract) to the module. It is recommended to add a short video welcome message.

**Special Requirements:** List all the requirements of the module, such as knowledge in programming languages.

**Authors:** Give the responsible organization and person of the module.

<p><b>1.1 Welcome</b></p> <p><b>Abstract</b></p> <p>Welcome to the distance learning module "Security and Privacy in the Internet". In this module you will learn about theoretical and practical aspects of security and privacy. The theoretical part covers a wide range of topics - from vulnerabilities over attacks and countermeasures to general security mechanisms and concepts. In the practical part you'll have the possibility to set up and validate a Kerberos installation.</p> <p><b>Special Requirements</b></p> <ul style="list-style-type: none"> <li>• none</li> </ul> <p><b>Authors</b></p> <p>The module "Security and Privacy in the Internet" has been created by the <a href="#">Computer Networks and Distributed Systems</a>, CNDs research group at the <a href="#">University of Bern</a> and has been financially supported by a Microsoft Research Curriculum Request for Proposal Award.</p> <p>For further conceptual information regarding the module "Security and Privacy in the Internet" please contact <a href="#">Attila Weyland</a>.</p>
---

**Figure 6: Introduction Welcome Example**

#### 4.1.2 The Goals and How to Reach Them

In this section you specify in detail what the student is going to learn and how long it should take him. This information is split into three paragraphs:

**Goals:** Formulate precise and brief learning goals, which cover these three aspects:

1. The task: What kind of activity must be done by the learner? You must use a verb in the formulation. Try to use a verb or an action that can be measured / assessed. Possible verbs are for example: writing, managing, reading, organizing, researching...
2. The condition: How must the learner perform the task? You must add objects to the verb: by measuring the traffic, by writing a socket, by summarizing the theory. You can add more conditions: in a team, alone, with the two routers, without help. You can even include variables: no matter how high the CPU load is, no matter how much time it takes.

3. The standards: How well the learner must perform the task. Many times you find standards defined like this:
  - How often? (every day, ten times, before starting the laboratory)
  - How well? (until passing the test with 0 errors, within one hour)
  - How many? (set up two different networks, collect two logs of the measurement)
  - How much? (during two hours)
  - How will I know it is done? (until being able to answer the asked questions)

**Storyboard:** Write a storyboard which covers these three aspects:

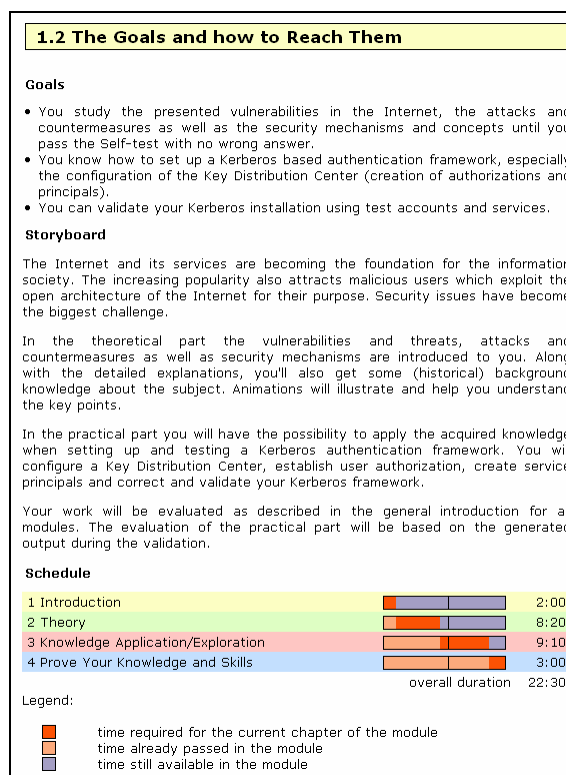
1. Write a short description of the learning and studying activities, which contains some lines of text as a contrary to the very short formulated goals.
2. Explain why doing the proposed activities will permit to reach the target goals with high standards.
3. Specify how they will be evaluated or how the level of goals reached will be assessed.

**Schedule:** Assess the duration of each main chapter of your module to help learners to plan their online work and study time. The total time for one module is 22.5 hours. Enter the duration in the table of contents file as described in Section 2.2.1 and 2.2.2.

Important: The Schedule section including title is generated automatically!

#### 4.1.3 Module Vicinity

Here you illustrate the module's relationship to other modules in the course in a graphical position map as well as create a mind map of your module. For example, Microsoft Visio [14] supports the creation of mind maps. You can find all OSLab related graphics, such as module logos, in the development bundle [8].



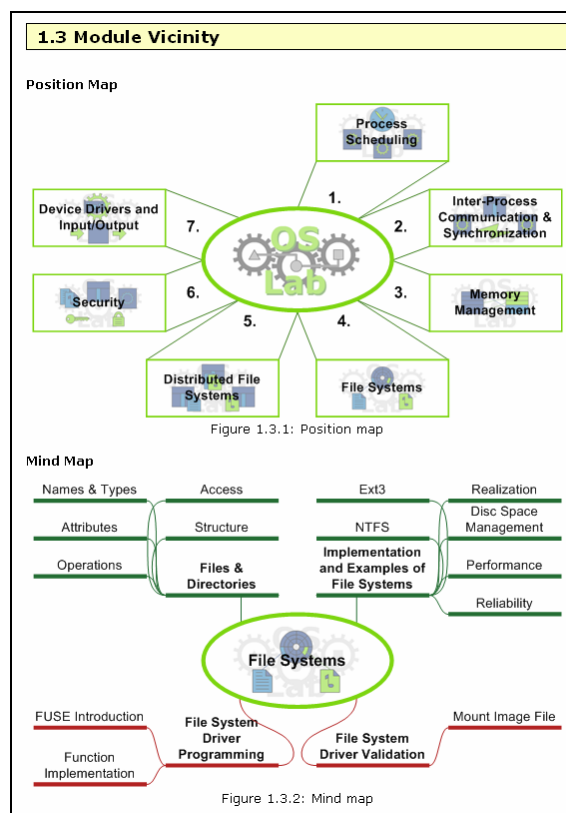
**Figure 7: Introduction The Goals Example**



**Position Map:** Propose a position of your module among all the other modules for those learners that work through the whole course.

Important: The position map is typically the same for all modules and created once. It is included in the module author bundle.

**Mind Map:** Locate the main topic of your module in the centre. From there, create branches outwards in all directions. Separate theoretical (north) and practical (south) topics. Use color codes for the theoretical (dark green - rgb(35,109,51)) and practical (dark red - rgb(178,34,34)) topics as well as for the module circle (light green - rgb(124,213,15)). The course and module logos can be obtained from the module author bundle.



**Figure 8: Introduction Module Vicinity Example**

#### 4.1.4 My Goals

Important: The content of the “My Goals” page is generated automatically. The essay field is made available with the help of the course platform as described in Section 5.

#### 4.1.5 Tips

If your module has any specific procedures, which are not covered in the global introduction, you have to explain them here. You can also mention well known pitfalls.

#### 4.1.6 FAQ

This section lists questions and answers, which arise during the usage of your module.

### 4.2 Theory

The theory chapter consists of four sections. The goal is to provide the necessary knowledge to the learner for the following hands-on session.

#### 4.2.1 Theoretical Basics

Create the theory of your module using text, graphics, animations, and interactive animations according to the guidelines in Section 3.1. You must also provide a summary of the complete theory in PDF format.



Important: The content of the first “Theoretical Basics” page is generated automatically and lists the sub-sections on the third level, which have been specified by the module author, e.g. “2.1.1 An Introduction to Security”.

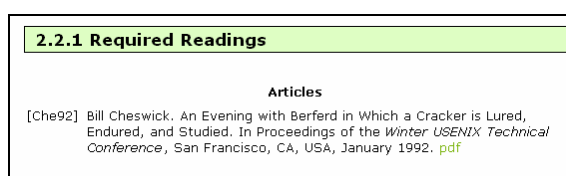
## 4.2.2 Readings

Introduce the learner to (scientific) literature related to your module and provide a categorized overview. The learner is required to read at least one scientific publication and should find a variety of recommended readings.

Important: The content of the first “Readings” page is generated automatically.

### 4.2.2.1 Required Readings

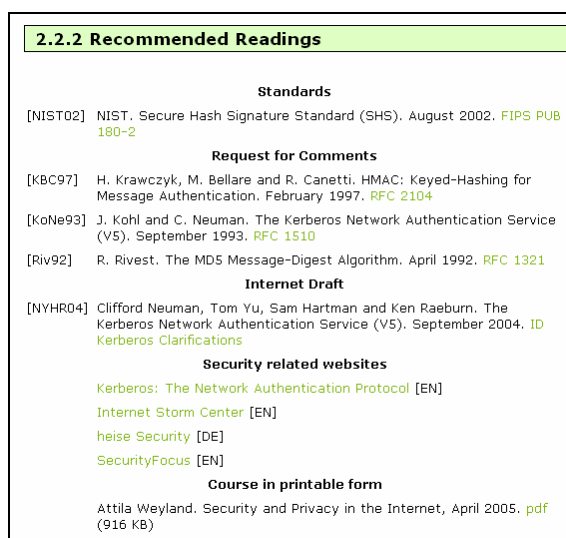
List one scientific publication, which is related to the module topic. The publication should allow a learner to get acquainted with scientific writing and provide some interesting background knowledge.



**Figure 9: Theory Required Reading Example**

### 4.2.2.2 Recommended Readings

List other readings that help to deepen the module topic for interested learners. If appropriate categorize the readings. List the generated PDF of your module theory.



**Figure 10: Theory Recommended Reading Example**

### 4.2.3 Self Test

Formulate questions and answers, which help the learner to discover missing parts of the theory. Give feedback for each correct or wrong answer. Include a reference to the corresponding theory section. Specify at least ten questions.

<p><b>2. Protocols</b> Select the answer where all correlations of ISO/OSI layer and the respective protocols at that layer are correct!</p> <p><input type="radio"/> Application Layer UDP Transport Layer TCP Network Layer IP Data Link Layer IEEE 802.3 Physical Layer 100 Base-T</p> <p><input checked="" type="radio"/> Application Layer FTP Transport Layer UDP Network Layer TCP Data Link Layer IEEE 802.3 Physical Layer 100 Base-T</p> <p><input type="radio"/> Application Layer SMTP Transport Layer TCP Network Layer IP Data Link Layer ARP Physical Layer 100 Base-T</p> <p><input type="radio"/> Application Layer HTTP Transport Layer UDP Network Layer IP Data Link Layer ARP Physical Layer IEEE 802.3</p>	<p><b>Feedback</b></p> <p><b>Incorrect</b> TCP is not a network layer protocol, but a transport layer protocol (see Theory section 1.1.1 Internet Basics).</p>
--	--

**Figure 11: Theory Self Test Example**

Important: The content of the “Self Test” page is generated automatically. The questions are made available with the help of the course platform as described in Section 5.

### 4.2.4 Quiz

Formulate questions and answers, which assess the knowledge of the learner and support the decision, whether he is granted access to the hands-on session or not. Write down the correct solution for each question to support the tutor in the grading process. Specify at least ten questions.

<p><i>[Quiz Security and Privacy in the Internet] 01. Vulnerabilities and Threats</i> Select all correct pairs of vulnerability and corresponding threat.</p> <p><input type="checkbox"/> 1. Resource Access - Intrusion</p> <p><input type="checkbox"/> 2. Resource Identity - Imitation</p> <p><input type="checkbox"/> 3. Person Privacy - Espionage</p> <p><input type="checkbox"/> 4. Person Identity - Disclosure</p> <p><input type="checkbox"/> 5. Resource Availability - Misuse</p> <p><input type="checkbox"/> 6. Information Confidentiality - Impersonation</p> <p><input type="checkbox"/> 7. Resource Consistency - Malfunction</p> <p><input type="checkbox"/> 8. Resource Usage - Disruption</p> <p><input type="checkbox"/> 9. Information Integrity - Manipulation</p>
---

**Figure 12: Theory Quiz Example**

Important: The content of the “Quiz” page is generated automatically. The questions are made available with the help of the course platform as described in Section 5.

## 4.3 Knowledge Application/Exploration

The knowledge application/exploration chapter consists of two sections. The goal is to let the learner apply the acquired knowledge in a realistic laboratory environment.

### 4.3.1 Introduction

Describe the particularities of your module's hands-on session, e.g. give a short manual for the hardware or the available commands of the used applications. You can also create sub-sections to structure the information, e.g. "3.1.2 Laboratory Environment".

**3.1.2 Laboratory Environment**

**User Accounts**

From the main page of this module you can log into the (not yet) Kerberos server *cell02* as *aeneas* or as *orpheus*. For the configuration of Kerberos you only need the *aeneas* account. The *orpheus* user will be used later for testing the Kerberos server and to obtain the test certificates.

**Editing Configuration Files**

The setup of the server is done using a set of configuration files, which have to be written/adjusted by the user (*aeneas*). Several editors are available for that purpose. For the less experienced user I recommend *nano*, others might prefer *emacs* or *vi*.

**Where to put data**

All configuration take place in the home directory of the user *aeneas*. A skeleton directory structure provides the recommended locations for the configuration files. However, the structure is not mandatory, so feel free to install Kerberos wherever you want (and abjure any kind of support from our side).

**Kerberos Programs**

*krb5kdc*  
is the kerberos key distribution center (aka Kerberos daemon). You can start the daemon directly, but we recommend to start it using the *krb.\**

**Figure 13: Knowledge Introduction Example**

### 4.3.2 Hands-on Session

Formulate the exercises for your laboratory. You must use simulation, emulation or real hardware to support the learner in the application of the acquired knowledge. Ensure that simulation and emulation are as close to reality as possible. Support trial-and-error learning. The following aspects have proven to be efficient in the formulation of an exercise:

1. Task: Describe the task in one or two sentences.
2. Preparation: Describe the necessary preparations to be done by the learner.
3. Directions: Give some guidance, e.g. for complex tasks, list the sub tasks.
4. Evaluation: Explain the steps necessary to evaluate this exercise.
5. Hints: List possible hints, which can help progress.

**3.2.2 Kerberos - Exercise 2**

The second exercise builds upon the first. On host *cell03* is a server running which can be contacted by a client *kerbzerb\_client*. The server only allows clients to log in, which are authorized for the "kerbzerb" service. You have to add the according principals and to create and distribute the proper keytabs allowing the client on *cell02* to connect to the server *cell03*.

**Task**            Add the according principal to allow the client on *cell02* to connect to the server on *cell03*.

**Preparation**   Finish exercise 1.

**Directions**    1. Add a principal for the *kerbzerb* service on *cell03* [format: *primary/instance@realm*]  
2. Add the new key to the *krb5.keytab* file for *cell03*  
3. Copy *krb5.keytab* to *cell03* (see below)


**Evaluation**    • Login in to *cell02* as user *orpheus* using the laboratory page.  
• From *cell02* use the *kerbzerb\_client* to connect to *cell03*.  
• If the client can login successfully, a test certificate will be automatically generated and stored. The test certificate can be viewed from the laboratory page.

**Figure 14: Knowledge Exercise Example**

Important: The content of the first "Hands-on Session" page is generated automatically and lists the sub-sections on the third level, which have been specified by the module author, e.g. "3.2.1 Kerberos - Exercise 2".

## 4.4 Prove Your Knowledge and Skills

The chapter "Prove Your Knowledge and Skills" consists of three sections. The goal is to assess the acquired knowledge of the learner and to obtain feedback for further improvements of the module.

	<p style="text-align: center;"><b>Operating System Laboratory</b></p> <p style="text-align: center;">Module Author Guide</p>	<p style="text-align: right;">v 1.01 25 July 2006</p>
---	--	---

#### 4.4.1 Personal Synthesis

Important: The content of the “Personal Synthesis” page is generated automatically. The essay field is made available with the help of the course platform as described in Section 5.

#### 4.4.2 Final Quiz

Formulate questions, which assess the knowledge of the learner and support the decision whether he has completed the hands-on session (and thus the course) successfully or not. Write down the correct solution for each question to support the tutor in the grading process.

Important: The content of the “Final Quiz” page is generated automatically. The questions are made available with the help of the course platform as described in Section 5.



**Figure 15: Prove Final Quiz Example**

#### 4.4.3 Survey

Important: The content of the “Survey” page is generated automatically. The questions and essay fields are made available with the help of the course platform as described in Section 5.

## 5 Module Integration

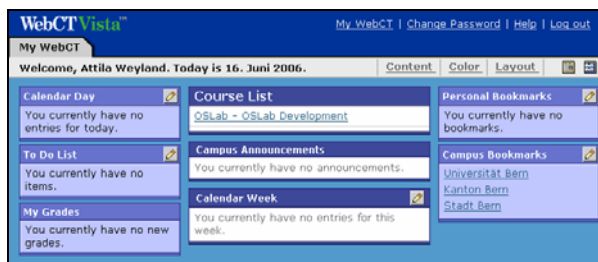
This section describes how to integrate your module content into a course platform and take advantage of the provided learning tools, such as essays and quizzes. The descriptions are based on the course platform WebCT Vista [15] version 3.0.6.17. The following sections describe how to access WebCT Vista, index content and apply the learning tools.

### 5.1 WebCT Access

WebCT Vista is operated by EduTech [16] and SWITCH [17]. You can access it via the portal page [18], where you should also check your browser for compatibility with WebCT Vista. The next steps describe how to access WebCT Vista.

1. Access the WebCT Vista portal. Select your institution from the “Institution Listing”. Log in with your provided designer account information. After you log in, you’ll see the page “My WebCT”.

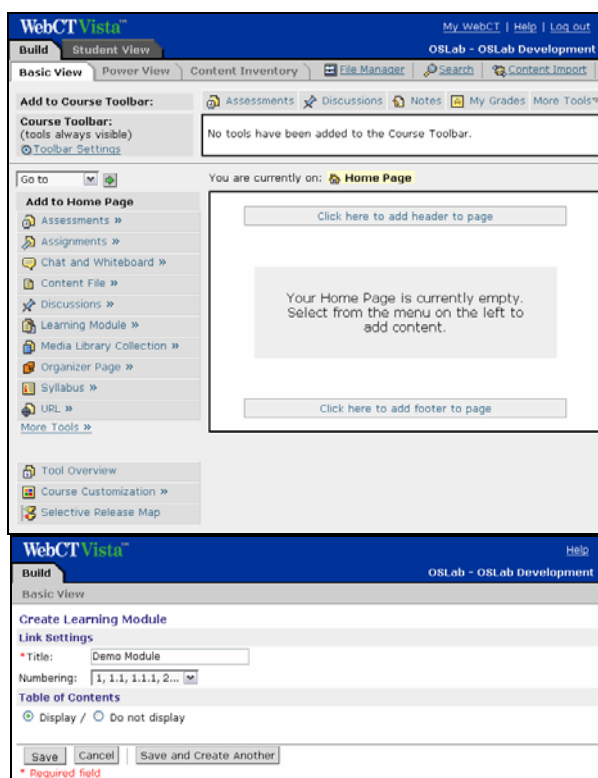
- Select the course, where you want to add your learning module, e.g. “OSLab – OSLab Development”. If you keep the mouse over a course list entry, your role pops up. After you selected the course, its homepage appears.



## 5.2 Content Subscription

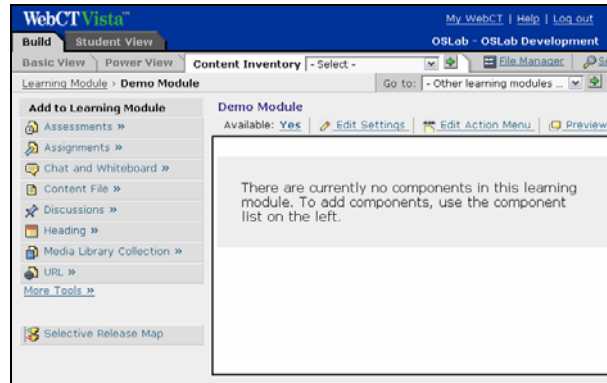
This section explains how to index your content inside a learning module. Notice the different tabs on the top of the page, which allow you to switch between different views. The following operations are all for the “Build” tab. The “Student View” tab allows you to preview your module.

- On the course homepage, select “Learning Module” from the “Add to Home Page” menu on the left. In the unfolding dialogue choose “Create Learning Module”. A “Create Learning Module” page appears.
- Type the title of your module, e.g. “Demo Module”, keep the other settings unchanged and click “Save”. The course homepage appears again. Click on your newly added module and select “Edit” from the popup menu. The content inventory of your learning module appears.

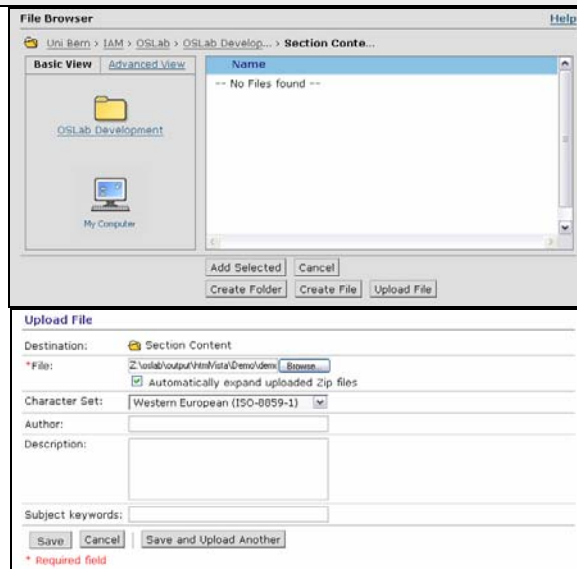




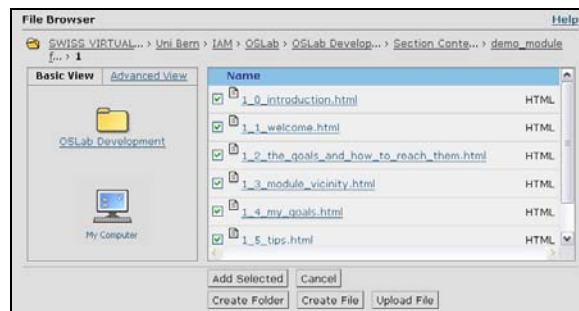
3. Select “Content File” from the “Add to Learning Module” menu on the left. In the unfolding dialogue choose “Select File”. The “File Browser” loads in a separate page.



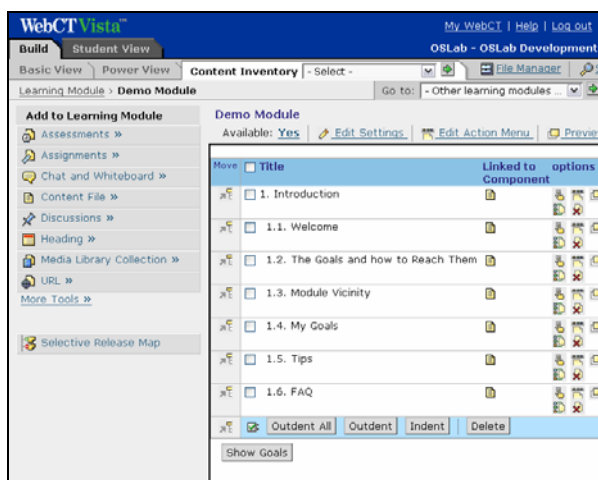
4. Select “Upload File”. An “Upload File” page appears.



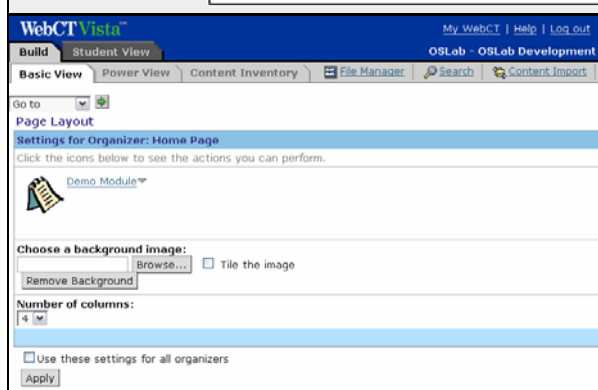
5. Browse for your local zipped learning module. Mark the check box “Automatically expand uploaded Zip files” and click “Save”. The “File Browser” appears. It lists your zip and a newly created directory, which contains all your files.
6. Select all files inside a chapter and click “Add Selected”. The “File Browser” disappears and the content inventory lists the newly added files.



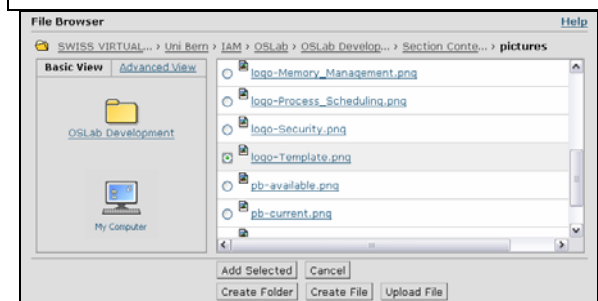
7. Indent the files according to their position in the table of contents. Repeat the processes described in 6 and 7 for each chapter.



8. On the module homepage click “Course Customization”. In the unfolding menu choose “Page Layouts”. The layout page appears. Click on the icon of your module and select “Replace Icon” from the popup menu. The “File Browser” appears.



9. Select the logo corresponding to your module from the pictures directory and click “Add Selected”. The “File Browser” disappears and the layout page is now updated. Click “Basic View” to return to the module homepage.




### 5.3 Learning Tools Application

This section describes how to realize the learning tools presented in Section 1.3 on page 4 with the possibilities of WebCT Vista. Table 4 gives an overview of the didactic learning tools and how we mapped them to the WebCT Vista learning tools.

**Table 4: Learning Tools Representation**

Availability	Learning Tool	Points	WebCT Vista Representation
Always	Discussion	0	Discussions – Discussion Topic
Always	Logbook	25	Assessment – Quiz
1.3	My Goals	20	Assessment – Quiz
2.3	<i>Self Test</i>	0	<i>Assessment - Self Test</i>
2.4	<i>Quiz</i>	30	<i>Assessment – Quiz</i>
4.1	Personal Synthesis	20	Assessment – Quiz



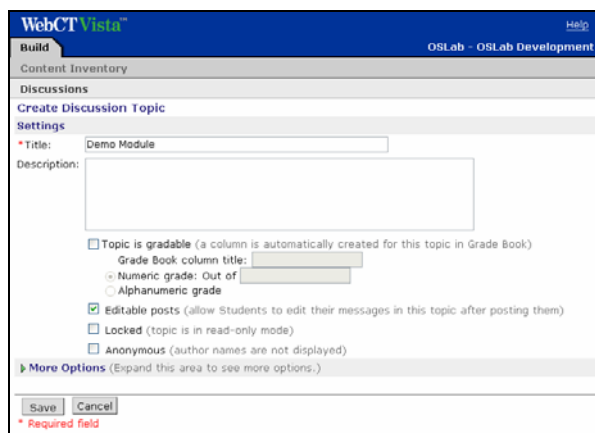
	<p align="center"><b>Operating System Laboratory</b></p> <p align="center">Module Author Guide</p>	<p align="right">v 1.01 25 July 2006</p>
---	--	--

4.2	Final Quiz	40	Assessment – Quiz
4.3	Survey		Assessment – Quiz

We only use discussions and assessments (of type quiz or self test). An assessment can contain multiple questions (of type Paragraph or Multiple Choice). We distinguish between common and *specific* assessments. The first category (Logbook, My Goals, Personal Synthesis, and Survey) are identical for each module, because they contain exactly the same questions. The second category (Self Test, Quiz and Final Quiz) contains specific questions formulated by the module author.

### 5.3.1 Discussions

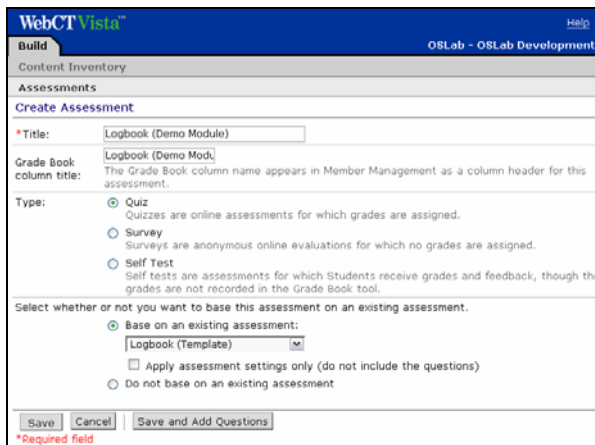
1. Click “Content Inventory” on the menu bar. The content inventory page appears. Select “Discussions”. The discussions page appears. Choose “Create Topic”. A new page appears. Enter the name of your module and click “Save”.



### 5.3.2 Common Assessments

This section describes how to add the common assessments (Logbook, My Goals, Personal Synthesis, and Survey) to your module. As common assessments rely on the same questions for each module, you can base your own assessments on the existing one from the Demo Module (including the question).

1. Click “Content Inventory” on the menu bar. The content inventory page appears. Select “Assessments”. The assessment page appears. Choose “Create Assessment”. A new page appears. Enter the name of the common assessment and the corresponding module name, select the type according to Table 4 and base your assessment on the prototype, e.g. “Logbook (Prototype)”. Click “Save”.
2. Repeat the procedure described in 1 for each common assessment listed in Table 4.

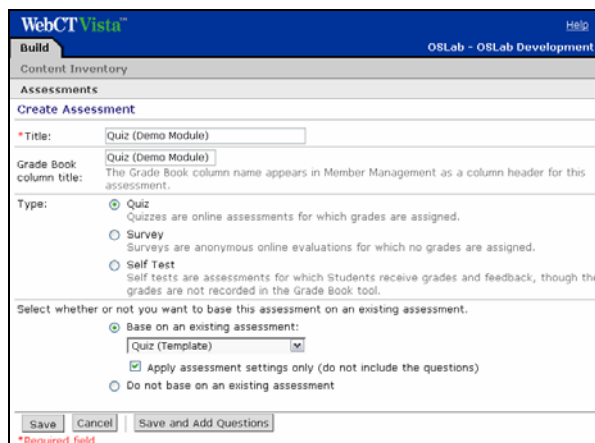




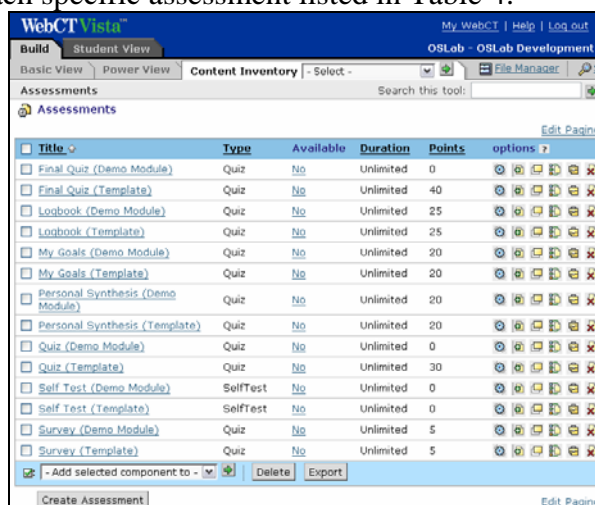
### 5.3.3 Specific Assessments

This section explains how to add specific assessments (Self Test, Quiz and Final Quiz) to your module. As specific assessments contain questions specific to your module, you have to exclude the questions, when basing your assessment on an existing one from the Demo Module. You'll have to add your own questions afterwards.

1. Analogue to the common assessment described above, enter the name of the specific assessment and the corresponding module name, select the type according to Table 4 and base your assessment on the prototype, e.g. "Quiz (Prototype)". Make sure that you do not include the questions. Also note the specific type of the self test. Click "Save".

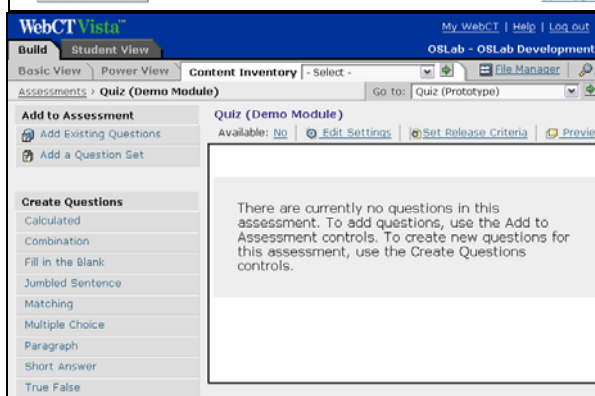


2. Repeat the procedure from 1 for each specific assessment listed in Table 4.
3. Select "Assessments" on the content inventory page. The assessment page appears. Choose a specific assessment from the list, e.g. "Quiz (Demo Module)". The assessment inventory page appears.



Title	Type	Available	Duration	Points	options
Final Quiz (Demo Module)	Quiz	No	Unlimited	0	
Final Quiz (Template)	Quiz	No	Unlimited	40	
Logbook (Demo Module)	Quiz	No	Unlimited	25	
Logbook (Template)	Quiz	No	Unlimited	25	
My Goals (Demo Module)	Quiz	No	Unlimited	20	
My Goals (Template)	Quiz	No	Unlimited	20	
Personal Synthesis (Demo Module)	Quiz	No	Unlimited	20	
Personal Synthesis (Template)	Quiz	No	Unlimited	20	
Quiz (Demo Module)	Quiz	No	Unlimited	0	
Quiz (Template)	Quiz	No	Unlimited	30	
Self Test (Demo Module)	SelfTest	No	Unlimited	0	
Self Test (Template)	SelfTest	No	Unlimited	0	
Survey (Demo Module)	Quiz	No	Unlimited	5	
Survey (Template)	Quiz	No	Unlimited	5	

4. Select the question type from the "Create Questions" menu on the left, e.g. "Multiple Choice". A new page appears.





## Operating System Laboratory

### Module Author Guide

v 1.01  
25 July 2006

5. Enter the title of your question, including the assessment type and your module name, e.g. “[Quiz: Demo Module] Question 1”. Enter the question text and the possible answers. Unfold the section “More Options” and specify the category of this question, e.g. “Quiz”. Click “Save”. The assessment inventory page appears again.

WebCT Vista™  
Build  
Content Inventory  
Assessments: Quiz (Demo Module)  
Create Multiple Choice Question  
Question  
\*Title: [Quiz: Demo Module] Question 1  
HTML Creator: On / Off  
Question text: Is this question useful?  
More Options (Expand this area to see more options.)  
General feedback:  
Use HTML: ☒  
Section Designer notes:  
Category: Questions may be placed in categories in the Question Database, for organizational purposes.  
Specify an existing category: Quiz  
Create a category:  
Save Preview Cancel  
\* Required field

6. Repeat the procedures described in 4 and 5 for every question you have in this assessment.
7. On the assessment inventory page, adjust the points of the single questions to give a total which corresponds to the values listed in Table 4.

WebCT Vista™  
My WebCT | Help | Log out  
Build Student View  
Basic View Power View Content Inventory - Select -  
Assessments: Quiz (Demo Module)  
Go to: Quiz (Prototype)  
Add to Assessment  
Add Existing Questions  
Add a Question Set  
Add a Part  
Create Questions  
Calculated  
Combination  
Fill in the Blank  
Jumbled Sentence  
Matching  
Multiple Choice  
Paragraph  
Short Answer  
True False  
Quiz (Demo Module)  
Available: No Edit Settings Set Release Criteria Preview  
Move Title Points Type options  
1. [Quiz: Demo Module] Question 1 30 Multiple Choice  
Total Points: 30 Update Total  
Add Question Alternates Remove

8. Repeat the procedure from 3 to 7 for each specific assessment.
9. Select “Assessments” on the content inventory page. The assessment page appears. Make all assessments of your model available by clicking the respective “No” in the “Available” column.

WebCT Vista™  
My WebCT | Help | Log out  
Build Student View  
Basic View Power View Content Inventory - Select -  
Assessments  
Assessments  
Search this tool:  
Title Type Available Duration Points options  
Final Quiz (Demo Module) Quiz Yes Unlimited 40  
Final Quiz (Template) Quiz No Unlimited 40  
Logbook (Demo Module) Quiz Yes Unlimited 25  
Logbook (Template) Quiz No Unlimited 25  
My Goals (Demo Module) Quiz Yes Unlimited 20  
My Goals (Template) Quiz No Unlimited 20  
Personal Synthesis (Demo Module) Quiz Yes Unlimited 20  
Personal Synthesis (Template) Quiz No Unlimited 20  
Quiz (Demo Module) Quiz Yes Unlimited 30  
Quiz (Template) Quiz No Unlimited 30  
Self Test (Demo Module) SelfTest Yes Unlimited 0  
Self Test (Template) SelfTest No Unlimited 0  
Survey (Demo Module) Quiz Yes Unlimited 5  
Survey (Template) Quiz No Unlimited 5  
- Add selected component to - Delete Export  
Create Assessment

### 5.3.4 Link Learning Tools and Content

This section explains how to link the learning tools (discussion and assessments) with the content. The integration is done via the action menu provided by WebCT Vista.



According to Table 4 the discussions and logbook must be available throughout the whole module. The other learning tools have specific locations.

1. Click “Basic View” on the menu bar. The module homepage appears. Click on your module and select “Edit” from the popup menu. The module inventory page appears. Select “Edit Action Menu” from the top of the content list. The action menu settings page appears.

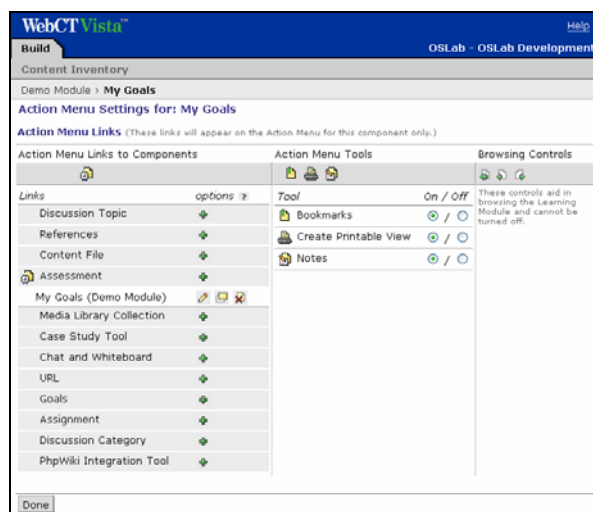
Links	options	Tool	On / Off
Discussion Topic	+	Bookmarks	On / Off
References	+	Create Printable View	On / Off
Content File	+	Notes	On / Off
Assessment	+		
Media Library Collection	+		
Case Study Tool	+		
Chat and Whiteboard	+		
URL	+		
Goals	+		
Assignment	+		
Discussion Category	+		
PhpWiki Integration Tool	+		

2. Select “Create Link to Component” by clicking the plus behind the entry “Discussion Topic”. A new page appears. Select the topic for your module, e.g. “Demo Module” and click “Add Selected”. The action menu settings page appears again.
3. Select “Create Link to Component” by clicking the plus behind the entry “Assessment”. A new page appears. Select the logbook for your module, e.g. “Logbook (Demo Module)” and click “Add Selected”. The action menu settings page appears again. Click “Done”. The module inventory page appears again.

Title	options
<input type="checkbox"/> Default Topic	
<input checked="" type="checkbox"/> Demo Module	

Title	options
<input type="checkbox"/> Final Quiz (Demo Module)	
<input type="checkbox"/> Final Quiz (Template)	
<input checked="" type="checkbox"/> Logbook (Demo Module)	
<input type="checkbox"/> Logbook (Template)	
<input type="checkbox"/> My Goals (Demo Module)	
<input type="checkbox"/> My Goals (Template)	
<input type="checkbox"/> Personal Synthesis (Demo Module)	
<input type="checkbox"/> Personal Synthesis (Template)	
<input type="checkbox"/> Quiz (Demo Module)	
<input type="checkbox"/> Quiz (Template)	
<input type="checkbox"/> Self Test (Demo Module)	
<input type="checkbox"/> Self Test (Template)	
<input type="checkbox"/> Survey (Demo Module)	
<input type="checkbox"/> Survey (Template)	


4. On the module inventory page select the “Edit Component Action Menu” for specific entries, e.g. “1.4 My Goals” by clicking the icon from the option column. The action menu settings page appears. Select “Create Link to Component” by clicking the plus behind the entry “Assessment”. A new page appears. Select the My Goals for your module, e.g. “My Goals (Demo Module)” and click “Add Selected”. The action menu settings page appears again.



5. Repeat the procedure in 4 for each of the remaining learning tools (2.3 Self Test, 2.4 Quiz, 4.1 Personal Synthesis, 4.2 Final Quiz, and 4.3 Survey).

## 6 References

- [1] Virtual Internet and Telecommunications Laboratory of Switzerland, VITELS Website. <http://www.vitels.ch/>
- [2] Marc-Alain Steinemann, Attila Weyland, Jacques Viens and Torsten Braun. VITELS Didactic and Design Guide v1.91. July 2005. Available online: [http://www.vitels.ch/doc/vitels\\_didactics\\_graphics\\_v1.91.pdf](http://www.vitels.ch/doc/vitels_didactics_graphics_v1.91.pdf)
- [3] End-to-end Quality of Service support over heterogeneous networks, EuQoS Website. <http://www.euqos.org/>
- [4] Pedro Vapi et al. file framework generator & formatter, FFGF Perl Version. Available online: <https://subversion.cnds.unibe.ch/svn/e-learning/ffgf/>
- [5] W3C. HyperText Markup Language, HTML. Available online: <http://www.w3.org/TR/html401/>
- [6] W3C. Cascading Style Sheets, CSS. Available online: <http://www.w3.org/TR/CSS21/>
- [7] Matthias Scheidegger. Guided Content Editor, GCED. Available online: <https://subversion.cnds.unibe.ch/svn/e-learning/gced/>
- [8] OSLab Development Bundle. Available online: <http://www.oslab.ch/>
- [9] Attila Weyland et al. OSLab Course Author Guide, Available online: [http://www.oslab.ch/doc/oslab-course\\_author\\_guide.pdf](http://www.oslab.ch/doc/oslab-course_author_guide.pdf)
- [10] Attila Weyland et al. OSLab Module Author Guide, Available online: [http://www.oslab.ch/doc/oslab-module\\_author\\_guide.pdf](http://www.oslab.ch/doc/oslab-module_author_guide.pdf)
- [11] Perl Website. <http://www.perl.com/>
- [12] HTMLDOC Website. <http://www.easysw.com/htmldoc/>
- [13] Java HTMLEditorKit Website. <http://java.sun.com/j2se/1.5.0/docs/api/javax/swing/text/html/HTMLEditorKit.html>
- [14] Microsoft Visio Website. <http://office.microsoft.com/visio/>

	<p align="center"><b>Operating System Laboratory</b></p> <p align="center">Module Author Guide</p>	<p align="right">v 1.01 25 July 2006</p>
---	--	--

- [15] WebCT Vista Website. [http://www.webct.com/software/viewpage?name=software\\_vista](http://www.webct.com/software/viewpage?name=software_vista)
- [16] EduTech. WebCT Vista 3 Support Corner Website. <http://www.edutech.ch/vista/>
- [17] SWITCH Website. <http://www.switch.ch/>
- [18] WebCT Vista Portal. <http://athena.virtualcampus.ch/>