LoCuS (Laboratory fOr CompUter Science) System is instructional software for teaching computer science theories through experiments. Experiments within the context of LoCuS are laboratory excursions akin to those found in other sciences like biology, chemistry, and physics. A *locus* (Latin for “place”) is a “collection of points which share a property” or “the path through which a point moves to satisfy a given condition.” The LoCuS logo is a Boy’s surface, named after its founder, Werner Boy. Boy’s surface is a particularly intricate locus which is described by a system of complex equations. Just like the Mobius strip, Boy’s surface has only a single side. Our goal is for LoCuS to define a new place to emphasize the science of computation as well as a new path for computer science, in the form of labs.
Chapter 1

Introduction

LoCuS is a software system containing a set of labs, that investigates theories in computer science. LoCuS is written in Java and runs from a website via a JNLP interface. The basic user interface is the same for all labs and simulates an actual science laboratory environment. Labs are written in XML notation to be executed within this framework. Each lab contains apparatuses that are used to perform specific functions (like display a graph) that can be manipulated by the user. The user records their findings and answers questions in a lab notebook that can be saved and turned in. LoCuS generates data files, completed labs, and answer sheets from the stored answers in the database. Each lab has different versions that are appropriate for different grade levels (middle school through undergraduate college). This document describes how labs are expressed. The hope is that after reading this document, you will be able to understand existing labs and to write your own lab! All of these components will be explained and exemplified within this document.

This chapter describes the aspects of the lab software that are universal across all of the written labs.

1.1 Launching the Application

The user will typically open the application by clicking on a link or a button on a web page. The software loads a program called LAUNCHER via a JNLP interface from the remote server. Then the core application code as well as any plugins (including apparatuses), the lab zip file, and WATCHER are downloaded and cached on the local machine by LAUNCHER. This allows more efficient start up when LoCuS is restarted on this machine.

1.2 Splash Screen

When the application launches, a splash screen is shown that prints several brief messages describing the current step of the loading process (Figure 1.1). After the loading process completes, the LoCuS system is fully initialized and the student is presented with the lab notebook, lab bench, and chalkboard.

Figure 1.1: Splash screen.
1.3 Lab Room

The lab room is the environment the user sees immediately after the splash page. It is designed to emulate an actual science lab, like what might be found in a high school chemistry classroom. Key elements of the lab room are described next.

1.4 Lab Notebook

The lab notebook contains the current lab document. It looks like a real notebook with lined paper (Figure 1.2). The lab notebook contains all the content associated with the lab and is the focal point of most interactions the student has with the LoCuS system.

![Lab Notebook Image]

Figure 1.2: Lab notebook.

The notebook is an interactive entity. The student can click through pages using the Forward and Back buttons, launch apparatuses, answer questions, and read text. The bulk of the work in writing a lab is filling the notebook with instructional content.

1.5 Chalkboard

The chalkboard is a small black window designed to emulate an actual chalkboard. Lab authors can write messages that pertain to the lab on the board for students to read. The mechanism for doing this is described later, in Section 5.6. The chalkboard is most useful as a way to communicate administrative type questions to the student, the kinds of messages that wouldn’t typically be found in a lab notebook. The chalkboard can also be used to remind students of equations, definitions, or concepts that were presented earlier in the lab.
1.6 Lab Bench

The lab bench is a small window in the lower right corner of the screen. It lists all the apparatuses made visible for the particular lab that is loaded. Students can click on an apparatus’s button in the lab bench to launch the apparatus (some labs don’t show a lab bench; see Section 3.5).
Chapter 2

A Brief Introduction to XML

All labs are written in XML (eXtensible Markup Language). XML is a very simple language that allows a lot of expressive power to describe documents like lab writeups in LoCuS. Its design goals emphasize simplicity, generality, and usability over the Internet. If you are familiar with HTML, XML will seem very familiar to you. What follows is a very basic introduction to XML as we use it. For more information visit http://www.w3schools.com/xml/default.asp

2.1 XML Elements

XML documents are composed of XML elements. In an XML document, elements are indicated by start tags and end tags. For an element, the start and end tags are the same, except the end tag begins with a ‘/’. Start tags must always be closed by an end tag. Between these start and end tags, content can be found.

There is one exception to this rule. Some tags exist by themselves with no closing tag. For example, the <br/> element inserts a line break into the document. Since there is no closing tag, the ‘/’ is put at the end of tag, before the closing ‘>’. Another element that does this is the <img/> element for images.

Let’s say we want to include a greeting: “Hello, how are you?”. The XML for this would be:

1 <p>Hello, how are you?</p>

In this example, the element is greeting. <p> is the start tag, </p> is the end tag. Note that tags are invisible on web pages or other parsed documents like LoCuS labs, so for the above example, one would simply see this on a web page as:

Hello, how are you?

Elements are often nested within other elements creating an organized hierarchy. When writing XML, it is crucial that elements are properly nested. Consider the following as an example.

1 <b><i>This text is bold and italic.</i></b>

Note that the <i> tags exist only within the <b> tags. Swapping the closing </i> and </b> tags would be incorrect and would generate an error from the parser.
In XML, white space on the page is ignored. In order to start a new line, or to leave spaces in the webpage, line breaks must be explicitly inserted using a `<br/>` element and the end of the text. Otherwise, material will be shown on the page continuously, with no line breaks or paragraphs.

In XML, the user specifies their own elements: they are not predetermined, but the elements are given names that are as descriptive as possible without being too long. XML tags are case sensitive. For our labs, we use a convention referred to as “CamelCase”. All letters of the element name are lowercase except the first letter of any words beyond the first. There are no spaces in the names of elements. Consider the following as an example.

```
<chalkboardSetter>
```

Note that the first letter is lower case, but all subsequent words have a capitalized first letter.

XML elements that will be needed by a lab author have already been specified as part of the LoCuS project. These elements are discussed later in this document and are broken down into different type (e.g. questions, apparatus, etc.). If an additional XML element is needed the Chief Programmer should be contacted to alter the XML schema.

### 2.2 Attributes

XML elements can have attributes that contain additional information about that element. Attributes often contain information that is valuable to the software manipulating the element. Attributes are linked to an element by an ‘=’ sign and the attribute itself is in quotes.

```
<introductionSection name="A Brief Introduction to XML"></introductionSection>
```

In the above example, the element `<introductionSection>` has an attribute, called `name`, with a value of “A Brief Introduction to XML”. This attribute sets it apart from other potential sections.

### 2.3 XML Schemas

Since element names are user-defined and simply contain text, an XML schema must be created that describes the structure of an XML document.

There is a schema already in place for this software. When writing a lab, it is important to only use elements that are part of the defined schema. These elements and attributes are described in detail in subsequent chapters. If the current schema is insufficient for any reason, then contact the CP about needed changes.

### 2.4 Comments

Sometimes, a writer might want to include comments within an XML document explaining how a particular element appears or what it does. Comments in XML are enclosed in special tags: `<!-- comment -->`. 
Do you want to participate in this study?

Note, when content is removed from a lab due to time constraints it is best to comment out the necessary lines. In the future, this content will not have to be rewritten, instead, they can be moved out of the comments.
Chapter 3

Lab Preamble

This chapter details the tags that are required at the beginning of every lab document. These tags typically define data and behavior that applies to the lab as a whole, rather than to a specific section.

3.1 Document Tag

The first line of any XML document should declare what type of document it is and what version of XML it uses. For LoCuS, this line should look exactly like this:

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

This tells the XML parser to use version 1.0 of the XML language and be ready for UTF-8 encoding of characters.

3.2 <lab>

The <lab> element is the root element within which all other elements must exist. XML requires all code to be contained within a single root element. So each lab begins with <lab> and ends with </lab>.

3.3 <configuration>

The configuration section allows the lab author to control various aspects of the LoCuS environment during the lab. One use of this section is to activate or deactivate the <evaluation> section at the end of the lab. While LoCuS is under development, labs will be periodically used for evaluation and feedback. While the <evaluation> element is true, LoCuS will collect profiling data on how the user is moving through the lab and will also display a survey at the end of the lab.

In addition, the <configuration> section can activate and deactivate the button controls along the right side of the lab notebook. These buttons include the help button, the export button, and the save button.

Finally, the front cover prompt can be customized. This is the string displayed next to the blank line on the cover. By default, this string is “Name”. During evaluations, it may not be ethical to collect the names of participants, so the prompt can be customized to collect, for example, the
name of the class the student is in. Note, that when the lab is run from the website all of the student’s actions and answers to questions within the lab will be identified with that student’s ID. Therefore, there is no need to collect the student’s name through the front cover prompt.

Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;evaluation&gt;</td>
<td>true or false</td>
<td>This lab is used for evaluation. Yes activates the &lt;evaluation&gt; section in the lab.</td>
</tr>
<tr>
<td>&lt;saveButtonOn&gt;</td>
<td>true or false</td>
<td>Activate the save button</td>
</tr>
<tr>
<td>&lt;exportButtonOn&gt;</td>
<td>true or false</td>
<td>Activate the export button</td>
</tr>
<tr>
<td>&lt;helpButtonOn&gt;</td>
<td>true or false</td>
<td>Activate the help button</td>
</tr>
<tr>
<td>&lt;frontCoverPrompt&gt;</td>
<td>String</td>
<td>String to display on the front cover. Default is “Name”. Strings have a colon (:) appended</td>
</tr>
</tbody>
</table>

Here is an example of the configuration section.

```
<configuration>
  <evaluation>true</evaluation>
  <saveButtonOn>false</saveButtonOn>
  <exportButtonOn>false</exportButtonOn>
  <helpButtonOn>true</helpButtonOn>
  <evalActivateTime>35</evalActivateTime>
  <frontCoverPrompt>Class</frontCoverPrompt>
</configuration>
```

Figure 3.1: All buttons in the configuration section relative to the forward arrow.

The front cover of the notebook is shown in Figure 1.2.
3.4 <metadata>

The <metadata> element allows specification of information about the lab including author(s), creation date, modification dates, and a description for the lab. One example <metadata> section is shown below.

```
<metadata>
  <title>Exploring the Theory of Locality</title>
  <author>Nathaniel Gephart</author>
  <author>Jaimie Sauls</author>
  <createdDate>2011-10-06</createdDate>
  <lastModifiedDate>March 3, 2012</lastModifiedDate>
  <description>Example lab that demonstrates the lab features.</description>
  <version>5.0</version>
  <chalkboardText>
    Welcome to the HelloWorld lab!
    Please enter in the name of your class on the given line. Once you have done that, you can use the arrow to continue to the next page.
  </chalkboardText>
</metadata>
```

As you can see, you may specify multiple authors if the lab was a collaborative effort. Dates are free form – you can use any format that is comfortable for you. Dates are not parsed by the system; they are simply for your reference.

![Figure 3.2: The title page outputted by the metadata above.](image)

Another element that can exist in the <metadata> section is the <chalkboardText> tag, which will be described in Section 5.6.
### Metadata Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;author&gt;</td>
<td>Name of an author, there can be multiple of this element</td>
</tr>
<tr>
<td>&lt;chalkboardText&gt;</td>
<td>Text appearing in the chalkboard panel at the start of the lab</td>
</tr>
<tr>
<td>&lt;createdDate&gt;</td>
<td>Date the lab was created</td>
</tr>
<tr>
<td>&lt;description&gt;</td>
<td>A short description of the lab</td>
</tr>
<tr>
<td>&lt;lastModifiedDate&gt;</td>
<td>Date the lab was last modified</td>
</tr>
<tr>
<td>&lt;level&gt;</td>
<td>A string representing the target audience level (e.g. high school)</td>
</tr>
<tr>
<td>&lt;longDescription&gt;</td>
<td>A longer description of the lab</td>
</tr>
<tr>
<td>&lt;version&gt;</td>
<td>A version number as a decimal value</td>
</tr>
<tr>
<td>&lt;versionDate&gt;</td>
<td>Date this version was released</td>
</tr>
<tr>
<td>&lt;timeEstimate&gt;</td>
<td>Number of minutes this lab is estimated to take</td>
</tr>
</tbody>
</table>

#### 3.5 <visibility>

Following the <metadata> section, the <visibility> tag controls which aspects of a lab are visible to the user. The most common use for this section is to declare which apparatuses to show on the work bench. The <visibility> tag may appear at any point in the lab, allowing the author to make specified controls visible or invisible. Consider the following example.

```xml
<visibility>
  <labBench>
    <apparatus name="plotter"/>
    <apparatus name="sequencer"/>
    <apparatus name="coloring"/>
  </labBench>
</visibility>
```

By default, apparatuses are not shown; however, by declaring them in the <visibility> section, you can show them on the lab bench when the lab is opened. The listing above informs LoCuS that it should show the Cartesian Grapher, Sequence Analyzer, and Color-by-Number apparatuses.

The <visibility> section also controls visibility of a progress bar that indicates how far into the lab the student is at any moment in time. By default, the progress bar is hidden. You can show the progress bar by declaring it in the <visibility> element as follows:

```xml
<visibility>
  <progressBar>true</progressBar>
</visibility>
```
In this case the lab bench is not present, and thus will not be displayed. This is useful when the student should not be distracted.

The final element that can be made visible or invisible is the chalkboard. If the chalkboard is visible then `<chalkboard>` will appear within the `<visibility>` section and shown below.

```
<visibility>
    <chalkboard>
</visibility>
```

By default the chalkboard is invisible.

Figure 3.4: Progress bar.

Visibility Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;labBench&gt;</code></td>
<td>complex type</td>
<td>If this element is present, the lab bench is visible. Otherwise, the lab bench is hidden.</td>
</tr>
<tr>
<td><code>&lt;chalkboard&gt;</code></td>
<td>empty</td>
<td>If this element is present, the chalkboard is visible. Otherwise, the chalkboard is hidden.</td>
</tr>
<tr>
<td><code>&lt;progressBar&gt;</code></td>
<td>true or false</td>
<td>Controls visibility of the progress bar.</td>
</tr>
</tbody>
</table>
Chapter 4

Lab Sections

Labs have a lot of flexibility in their format and presentation. However, all labs share some elements in common. In particular, labs typically all have certain sections like an Introduction, Pre-Lab, Lab, and Post-Lab. These common sections are given predefined tags for convenience.

4.1 Introduction

The introduction typically gives a short paragraph summarizing what the lab is investigating, and a rough outline of the objectives and procedures the lab will explore.

The introduction is enclosed in `<introductionSection>` tags.

4.2 User Assent

If the responses from this lab are intended for data collection and assent is required from the user, the `<surveyAgreement2>` tags are available. If assent is required, then the following code should be included as part of the introduction section described above.

```
1 <surveyAgreement2>
2 <question>
3 <text>
4 <!--actually use true / false to determine the agreement status -->
5 Do you want to participate in this study?
6 </text>
7 </question>
8 <choice correct="true">
9 <text>Yes, I want to be in this research study.</text>
10 </choice>
11 <choice correct="false">
12 <text>No, I do not want to be in this research study.</text>
13 </choice>
14 </surveyAgreement2>
```

The true/false response of the student is saved in the exported version of the lab as a yes/no answer.
4.3 Pre-Lab

Each lab has a pre-lab section. This section must be completed before students can begin the actual lab portion. For easy material, the pre-lab could be short, designed to be completed in a few minutes. For more difficult material, the pre-lab could be more in depth, designed to be completed in one 50 minute class session. The pre-lab contains:

- background information
- practical applications of the theories presented
- practice questions and exercises where appropriate

The pre-lab section is enclosed in `<preLabSection>` tags.

At the end of the pre-lab section is a quiz, enclosed in `<labQuiz>` tags. The quiz sections are designed to test the concepts presented in the section. The student is required to answer every question on the quiz before they he or she can continue to the Lab section.

4.4 Lab

The lab section is where the student uses the scientific method in order to investigate a theory. Each lab must require the student to:

- form or identify a hypothesis
- test the hypothesis by experimental manipulation of variables
- record the results of their experiment
- analyze the results of their experiment
- form a conclusion based on the results of their experiment (did the data support the hypothesis?)

Just like the pre-lab, the lab can contain practice questions for the students to answer. The lab section is contained by the `<labSection>` tags.

4.5 Post-Lab (Optional)

If desired, a lab can have a post-lab section containing additional questions for fun or for homework. These can be take-home questions that do not require the lab software in order to be answered. This is a good place for free-response questions that require the student to take some time and thought in writing.

The post-lab is contained by the `<postLabSection>` tags.
4.6 Open Lab (Optional)

If desired, a lab can include an additional section at the end titled Open Lab. Here, students are encouraged to explore the program in a more free-from way, with very little guidance. This section could be used in order to:

- encourage students to explore, tinker, manipulate the various apparatuses
- give the students a hypothesis to test that is different from the guided lab, and have them write up a lab in a less guided fashion

The open lab section is contained by the <openLabSection> tags.

4.7 Evaluation (Optional)

This section is activated with the <evaluation> element in the <configuration> section (see above). Evaluation sections are defined by the <evaluation>..</evaluation> tags. Within these tags, the author can include any of the questions defined in Section 6 and an additional question type: the Likert Question. Evaluation sections have the ability to be placed immediately after the introduction section or they may occur at the end of the lab.

If an evaluation is present the survey agreement button must be included at the end of the introduction section. The student will be required to answer this question in order to go further in the lab. If the student does not consent to the survey, then the student will be prompted with a button that will skip over the survey questions. Note, if the evaluation section is at the end of the lab the button presented to the student will skip over the survey and close the lab. Otherwise the student will be shown a button that will allow them to proceed to the survey when they reach the evaluation section.

Likert questions ask a person to rank how much they agree with a statement on a scale from one to four or five. They are defined with the <likertQuestion> element which has one attributes called choices that can be either 4, 5 or 7. Four choices gives the student the following options:

- Strongly disagree
- Disagree
- Agree
- Strongly agree

If five choices is indicated, an additional middle choice is given: “neither disagree nor agree”. If all seven choices are given “somewhat agree” and “somewhat disagree” are also added. Note, if the number of choices is not indicated the default options is seven choices. Here’s an example of a Likert question.

<likertQuestion choices="5">I like pizza.</likertQuestion>

4.8 Custom Sections

LoCuS also gives the author the ability to create custom sections using the <section> tag. These sections function identically to the other sections described in this chapter, but are defined by the author. The have a name attribute that allows the author give the section a customized name.
Figure 4.1: Likert question outputted from the XML above.
Chapter 5

Creating Content

Within each lab section, text is processed as HTML. All of the typical conventions and tags apply. There are some extensions that implement functionality that is unique to LoCuS as well. This chapter describes the elements that are used most frequently when writing labs.

5.1 Paragraphs, Text, and Lists

Most text in lab documents is enclosed in `<p>` tags, for paragraphs. This is usually generic prose. There are a few simple tags that allow changing the style of text in `<p>` tags.

- `<b>` Emboldens the enclosed text.
- `<i>` Italicized the enclosed text.
- `<u>` Underlines the enclosed text.
- `<sub>` Creates subscripted text.
- `<sup>` Creates superscripted text.

Below is an example of some of the above tags, including an image:

```
1 <p> Normal, <b>Bold</b>, <i>italic</i>, <u>underline</u>. </p>
```

Lists can be created with either bullets or numbers. Bulleted lists, referred to as unordered lists, are enclosed in `<ul>` tags, while those with numbers, ordered lists, use `<ol>` tags. For both types of list, each item must be enclosed in `<li>` (list item) tags. Below is an example of a bulleted list. Both the XML and the image as it appears in the lab are shown.

```
1 <ul>
2 <li>Item one</li>
3 <li>Item two</li>
4 <li>Item three</li>
5 <li>Item four</li>
6 </ul>
```
Finally, there are a few miscellaneous tags that help to direct the flow of text in the document.

\(<br/>\) Forces a line break at the location of the tag.

\(<pageBreak/>\) Forces a page break at the location of the tag. Content written after a page break element will display on the next page.

---

**Figure 5.1:** The \(<br/>\) element kicks down one line.

These tags are very useful for breaking up content into manageable chunks. It is typically better to break paragraphs with \(<p>\) tags instead of \(<br/>\), however there are specific instances when an author may not want to logically break a paragraph but needs a physical line break.

---

**Figure 5.2:** The use of p tags kicks down two lines.

### 5.2 Tables

Tables follow a similar structure as lists, though there is some additional work involved. In LoCuS, tables are organized into rows and then columns. They are defined using the \(<table>\) tag. Each row is enclosed in \(<tr>\) tags, and each column of that row is in \(<td>\) (for table data) tags. Here’s an example.

```html
1  <table border="1">
2    <tr>
3      <td><b>Month</b></td>
4      <td><b>Savings</b></td>
5    </tr>
6    <tr>
7      <td>January</td>
8      <td>$100</td>
9    </tr>
10   </table>
```
This would produce a table that looked something like this:

<table>
<thead>
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>$100</td>
</tr>
</tbody>
</table>

Note the `border` attribute of the `<table>` tag. This attribute specifies the thickness of the border in pixels. In this example, you simply get a solid black line around each cell of the table.

In LoCuS, tables can also be used as “fill in the blank” type questions. You can create blank cells in a table using the `<fillUp>` element like this:

```html
<fillUp size="12"/>
```

This example creates a blank space twelve characters wide into which a student can type an answer. You can insert these tags between any `<td>..</td>` tags to create an empty cell of that size.

### 5.3 Images

The `<img/>` element inserts images into the lab document. Images are centered on the page at the point at which the element exists in the document. Here’s an example:

```html
<img src="resources/picture.png"/>
```

The `src` attribute specifies the path to the image file. All paths are relative to the lab zip file root. In this example, LoCuS looks for `picture.png` in the resources directory in the lab zip file. Note, any images specific to a lab should be contained in a resources folder for that lab.

### 5.4 Launching Apparatuses

The `<launchApparatus>` element creates a button in the lab document that the student can click on to launch an apparatus or a set of apparatuses. The element has one optional attribute: `caption`. The `caption` attribute is the text that is displayed on the button that the user can click to launch the apparatus.

The `name` is how LoCuS knows the apparatus. This name is internal to the LoCuS system and should be specified in the writeup describing the apparatus.

Within the `<launchApparatus>` element there can be any number of `<apparatus>` and `<wire>` tags. One instance of an apparatus will be launched for each `<apparatus>` element when the button is pressed. The `<apparatus>` element can have three attributes:

1. The `name` is how LoCuS knows the apparatus. This name is internal to the LoCuS system and should be specified in the writeup describing the apparatus.

2. The optional `primary` attribute is a boolean attribute that tells LoCuS that the apparatus must be shown in the foreground.

3. The optional `caption` attribute is displayed as the button text when the `primary` attribute is set to true.
The `<apparatus>` element can contain a single sub-element called `<options>`. The options specified in this element are specific to each apparatus and will also be described in the apparatus specification. Refer to those documents for help in passing options to apparatuses.

The `<wire>` element specifies how the apparatuses launched by the button should be connected. This tags accepts no attributes and two sub-elements: `<input>` and `<output>`. Both `<input>` and `<output>` tags accept the same set of attributes.

1. The **apparatus** attribute is the name of the apparatus that owns the socket.
2. The **name** attribute is the name of the socket.
3. The **type** attribute is the type of the data that the socket accepts.

For more information on the socket framework within LoCuS, please refer to the Apparatus Developer’s Guide.

Below is the example that launches the Locality Analyzer and Plot Tool apparatuses with their sockets connected.

```xml
<launchApparatus caption="Sequence Analyzer: Matrix Multiplication">
  <apparatus name="localityanalyzer" caption="Sequence Analyzer: Matrix Multiplication" primary="true">
    <options>
      <dataFile>2matrices.seq</dataFile>
      <mode>timestep</mode>
      <showBackButtons>false</showBackButtons>
    </options>
  </apparatus>
  <apparatus name="plottool" caption="Plotting Apparatus">
    <options>
    </options>
  </apparatus>
  <wire>
    <input apparatus="plottool" name="command" type="String" />
    <output apparatus="localityanalyzer" name="command" type="String" />
  </wire>
  <wire>
    <input apparatus="plottool" name="point" type="double[]" />
    <output apparatus="localityanalyzer" name="point" type="double[]" />
  </wire>
</launchApparatus>
```

Most of these lines are specific to the apparatus, but as you can see, all the major elements described above are present.
5.5 Questions

In addition to the ability to insert text and various other media into a lab document, LoCuS also has the ability to gather feedback from students in the form of three types of questions: multiple choice, fill in the blank, and free response.

All questions can be inserted directly into a section tag, or nested in a quiz section. If the question is in a section element it is considered a practice question.

All questions have an optional vname attribute that can be used to uniquely identify the question in the database. If this is not specified the correct answer cannot be extracted from the database for grading purposes or extract study information.

Further, all questions have the ability to show hints if they are specified by the lab author.
When a hint is added to a question, a small question mark icon appears near the question in the lab. The student can click on the question mark to read the hint. Since this feature is somewhat unique for each question type, it will be explained in more detail in each section.

Responses to all questions are saved as part of the lab when a student clicks the “Save” button in LoCuS. In addition, clicking on the “Export to PDF” button in LoCuS will export the lab including their answers to any questions as a PDF document.

5.5.1 Multiple Choice

The `<multipleChoiceQuestion>` element gives a question in multiple choice format. The following example shows the general format for a multiple choice question in XML. Each question has a `<question>` element and multiple `<choice>` elements. Here is an excerpt from our lab on the theory of locality.

```xml
<multipleChoiceQuestion vname="theory_developement">
  <question>
    <text>The theory of locality developed out of which field?</text>
  </question>
  <choice correct="false">
    <text>Biology</text>
    <feedback>Wrong!</feedback>
  </choice>
  <choice correct="false">
    <text>Physics</text>
    <feedback>Nope!</feedback>
  </choice>
  <choice correct="false">
    <text>Geology</text>
    <feedback>Try again!</feedback>
  </choice>
  <choice correct="true">
    <text>Computer science</text>
    <feedback>Yes!</feedback>
  </choice>
</multipleChoiceQuestion>
```

Note, that the above code contains an attribute to the multiple choice questions called `vname`. This attribute is used to uniquely identify individual multiple choice questions as described in Section 5.5.

As you can see, each choice has an attribute called `correct` that tells LoCuS whether it is the right answer or not. Students indicate their answer by clicking on a radio button next to the appropriate choice in the lab document.

For multiple choice questions embedded in section tags (i.e. those considered practice questions), a green check mark or red ‘X’ will appear next to the student’s selected answer to indicated if that answer is correct or not. Questions in evaluation sections do not give this feedback.

Hints can be inserted into each `<choice>` element via the `<feedback>` tag. If a hint exists for a particular choice and the question is a practice question, the hint will be shown along with the icon indicating whether the answer was correct or not. Hints appear in blue text to the right of the selected answer. Note that it is best if hint is no longer than one line. If the feedback is longer than one line, it can push material after the question off the page.
If you are intending to compare the answers to the practice questions in the pre-lab and lab sections to those in the final evaluation section to analyze the knowledge progression from the beginning to end of the lab, it is suggested that you do not include the correct="true/false" portion of the <choice> tag. Since the green check and red 'X' appear right away, students generally continue to search for the correct answer. When their answers are later exported to the csv or pdf formats, only the most recent selection is recorded. So, their original guess is lost, and only the correct answers are saved from the lab. In this case, it might be more helpful to instead include a follow-up page after each section of practice questions that lists the correct answers and their explanations.

Figure 5.4: The multiple choice question outputted from the XML above.

### 5.5.2 Free Response

Free response questions are larger blanks in which a student can compose more detailed responses to questions. The text entry boxes that these tags create will extend via scroll bars to allow the student as much space as is necessary to answer the question. They may be included within the lab for all students to do, or added on to the end of the lab as a homework assignment or option in-class assignment (this is especially useful in case some students get finished with the lab early). Free response questions use the <freeResponse> element like this:

```xml
<freeResponse lines="10" vname="compare_networks" isEssay="true">
  <question>Compare and contrast human social networks with food web networks.</question>
</freeResponse>
```

The lines attribute indicates how large to draw the entry box in lines in the lab document. The vname attribute identifies this question in the database as described in Section 5.5. Lastly, the isEssay attribute requires a boolean variable. If this attribute is true then the question is counted as an essay question when the answers are evaluated, however the appearance of the question in the lab is not changed. It is left to the discretion of the lab author to determine which questions should be considered essay type questions. As a general rule an essay question should take at least a paragraph to completely answer. Note, the default for isEssay is false.

Each <freeResponse> must have a <question> element, which can take HTML tags. You can optionally include a hint with the <hint> tag. Additionally, free response questions may have an isEssay = option, which take a Boolean. If this is set to true, then the question is considered an
essay question when calculating overall lab statistics, such as, average essay length. Note, false is the default.

| Compare and contrast human social networks with food web networks. |
|---|---|
| Students can write inside of this box. |
| It will become scrollable when necessary. |

Figure 5.5: Free response question outputted from the XML above.

5.5.3 Likert Question

Likert questions are commonly used in psychological evaluations. They are generally statements for which the subject must choose whether they disagree, slightly disagree, slightly agree, or agree. LoCuS has the capability to generate these kinds of questions with four, five, or seven options. If you want five options, there is a "neither disagree or agree" option in the middle. In addition, “somewhat agree” and “somewhat disagree” are added if seven options are specified. Here’s what a Likert question looks like:

```xml
<likertQuestion choices="5" vname="apples">I like apples and bananas.</likertQuestion>
```

Figure 5.6: Likert question outputted from the XML above.

These questions are typically used in the <evaluation> section for doing lab evaluations.
5.6 Chalkboard

The chalkboard is a small window in the 2D LoCuS window onto which the lab author can write arbitrary text. Text is written to the chalkboard whenever a `<chalkboardSetter>` or `<chalkboardAppender>` element is encountered on the current page. The former sets the text of the chalkboard to exactly the text within the tag, and the latter appends the text to the bottom of the chalkboard. In addition, the `<chalkboardClear/>` element clears all text from the chalkboard. All text within these tags can be composed with standard HTML tags as described in previous sections.

Here is a simple ChalkboardSetter tag:

```
<chalkboardSetter>
Welcome! I can make colors using HTML
<b><font color="green">like this!</font></b>
</chalkboardSetter>
```

![Figure 5.7: Result of the Chalkboard Setter above.](image)
Chapter 6

Generated Files

A number of files are generated when a lab is created and used by students. This chapter describes each one.

6.1 Saved Lab Files

Saved lab files are obsolete. This data can now be extracted from the database.

As students go through the lab, their work is saved. They can also save manually. The saved lab file can be opened by the student at a later time so that he or she can continue to work on it. The saved file is located on a local machine in the `/locus/labs/` directory.

The student’s answers can also be exported as a csv file. The csv file contains only the answers the students gave to each question type in the order they appear in the lab.
Appendices
A Lab zip File Format

All labs are packaged as zip archives. When LoCuS loads a lab, the archive is unzipped and the contents are stored and read on the local machine. This chapter outlines the important aspects of the zip file. The zip file is configured by the Chief Programmer. This appendix provides an overview of the structure of the lab archive.

All lab archives must contain a lab.xml file in the root of the archive. This is an XML file that is parsed by LoCuS to display the contents of the lab notebook, determine which apparatuses to show on the lab bench, and do basic configuration for each lab.

Apart from the necessary lab.xml file, the rest of the contents of the zip file can be determined by the author. It is up to him or her to decide how to organize content for the lab. All files referenced in the lab.xml are relative to the root of the zip archive. For example, a file called “resources/test.png” is located in the resources folder of the zip file.

As an example consider the archive layout for the Introductory lab.

```
lab.xml
resources/
  back.png
  export.png
  forward.png
  help.png
  save.png
datafiles/
  rex.txt
```

As you can see, lab.xml exists in the root of the archive. There are also two other directories: resources and datafiles, each with some files beneath them. These files are referenced in lab.xml as noted above.