Connexions Tutorial and Reference

Collection Editors:

Connexions
Adan Galvan
Brent Hendricks
Mark Husband
Connexions Tutorial and Reference

Collection Editors:
Connexions
Adan Galvan
Brent Hendricks
Mark Husband

Authors:
Connexions
Chuck Bearden
Sarah Coppin
Adan Galvan
Elizabeth Gregory
Brent Hendricks
Mark Husband

Ricardo Radaelli-Sanchez
Charlet Reedstrom
Philip Schatz
Max Starkenburg
Natalie Weber
Daniel Williamson

Online:
< http://cnx.org/content/col10151/1.27/ >

OpenStax-CNX
# Table of Contents

1 Connexions Overview
   1.1 Basic Site Layout .......................................................... 1
   1.2 Create an Account ............................................................ 6
   1.3 Report a Bug ................................................................. 8

2 Viewing and Finding Content
   2.1 Modules and Collections .................................................. 11
   2.2 Metadata ........................................................................ 20
   2.3 Lenses .......................................................................... 24
   2.4 Finding Content .............................................................. 26

3 The Editing Interface
   3.1 MyCNX Tab ............................................................... 33
   3.2 Navigating the Editing Interface ........................................ 35
   3.3 Work Areas ................................................................. 37

4 Authoring Content
   4.1 Create new module ......................................................... 49
   4.2 Import/Export .............................................................. 51
   4.3 Edit Module .................................................................. 74
   4.4 Build Collection .......................................................... 183
   4.5 Creating Adaptations ..................................................... 196
   4.6 Create Lens .............................................................. 201

5 CNXML Tutorial
   5.1 XML Basics ............................................................. 205
   5.2 Combining XML Languages .............................................. 208
   5.3 The Basic CNXML ....................................................... 210
   5.4 The Intermediate CNXML ................................................ 221
   5.5 The Advanced CNXML ................................................... 230
   5.6 CNXML Reference Extensions .......................................... 235
   5.7 Examples .................................................................... 241
   Solutions ........................................................................... 252

Glossary ................................................................................. 253
Bibliography ........................................................................... 254
Index .................................................................................... 256
Attributions ............................................................................ 260
Chapter 1

Connexions Overview

1.1 Basic Site Layout

1.1.1 Overview

Connexions is a web-based document creation and management system for educational and research materials. There are two parts to Connexions: a content repository that contains these materials, and the software tools necessary to create, manage, and access these materials. The software underlying Connexions is called Rhaptos and was initially developed by the Connexions project specifically for Connexions.

From its inception, Connexions was designed to allow the collaborative development and free availability of material. Instructors and authors can modify this material for any educational purpose. Connexions offers free and open source software tools to help students, instructors, and authors manage these information assets for sharing and advancing knowledge to benefit the global educational community. All of this is accomplished through the use of the Creative Commons Attribution license.

There are several general categories of Connexions users:

- Students - who use Connexions to access information on-line, to prepare for their classes.
- Authors - who enter content into Connexions in document files called modules.
- Instructors - who build collections, which are documents created by linking related modules together in a specific order.
- Organizations - who endorse quality content, or highlight content affiliated with their organization, in a post-publication vetting system called lenses.

---

1 This content is available online at <http://cnx.org/content/m10884/2.28/>.
2 http://cnx.org/content
3 http://rhaptos.org/
4 http://creativecommons.org/
5 http://creativecommons.org/licenses/by/3.0/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 1. CONNEXIONS OVERVIEW

An individual or organization may, of course, fall into more than one of these categories.

1.1.2 The Connexions Home Page

The Connexions home page (Figure 1.1) is the starting point for your use of Connexions. It contains the following items:

- Home, Content, Lenses, About Us, Help, and MyCNX tabs (Section 1.1.2.1: Home, Content, Lenses, About Us, Help, and MyCNX Tabs)
- Breadcrumb Navigation Bar (Section 1.1.2.2: Breadcrumb Navigation Bar)
- Search Box and Quick Links (Section 1.1.2.3: Search and Quick Links)
- My Account and Login Sidebar (Section 1.1.2.4: My Account and Login Sidebar)
- Featured Content (Section 1.1.2.5: Featured Content)
- Find Content (Section 1.1.2.6: Find Content)
- Create Content (Section 1.1.2.7: Create Content)
- Spotlight Sidebar (Section 1.1.2.8: Spotlight Sidebar)
- Connexions News Sidebar (Section 1.1.2.9: News Sidebar)

Some of these items also appear on the other Connexions web pages.

Figure 1.1: Connexions home page.

1.1.2.1 Home, Content, Lenses, About Us, Help, and MyCNX Tabs

On the upper left of the Connexions home page are tabs labeled Home, Content, Lenses, About Us, Help, and MyCNX. Clicking on these tabs display the following Connexions pages:

- Home\(^6\) returns to the Connexions home page.
- Content\(^7\) takes you to the Content landing page, where you can search (Section 2.4.1: Searching the Content Repository) for modules and collections of interest or browse (Figure 2.28) the repository by subject, author, popularity and more.
- Lenses\(^8\) is a portal to all user and organization-created lenses\(^9\).

\(^6\)http://cnx.org/
\(^7\)http://cnx.org/content/
\(^8\)http://cnx.org/lenses/
\(^9\)http://cnx.org/help/viewing/lenses

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
About\textsuperscript{10} displays the "About Us" page that contains a general description of Connexions and links to more detailed information about Connexions, how it works, ways to get involved, the people behind it, and a list of contacts.

Help\textsuperscript{11} displays the "Help" page that contains links to installation, reference, and other information to answer your questions about Connexions.

MyCNX\textsuperscript{12} displays your member dashboard if you are signed in. This page includes links to your Work Areas, your lenses, and content you’ve recently visited.

1.1.2.2 Breadcrumb Navigation Bar

The You are here breadcrumb navigation bar (Figure 1.2) is located just below the main site tabs (Section 1.1.2.1: Home, Content, Lenses, About Us, Help, and MyCNX Tabs). This navigation tool is a breadcrumb trail of the Connexions pages you have displayed as you have drilled into Connexions to get to the current page. By clicking on any point within the breadcrumb trail you can return to that page.

![Image not finished](Image not finished)

Figure 1.2: Breadcrumb navigation bar

1.1.2.3 Search and Quick Links

You can search the repository\textsuperscript{13} for a specific module or a collection using the Search button and text box in the upper right of the Connexions home page. Type in a name, keyword, collection title, module title, text string, or object ID in the Search text box (Figure 1.3) and click Search. Connexions will display a list of all the content that matches your entry. To view a module or collection, click on its title.

Above the search box is a toolbar with some quick links for your convenience. These links vary, depending on whether you are logged into the system or not.

\textsuperscript{10}http://cnx.org/aboutus/
\textsuperscript{11}http://cnx.org/help/
\textsuperscript{12}http://cnx.org/mycnx/
\textsuperscript{13}http://cnx.org/content/
If you have a Connexions account and are logged into the system, you can perform the following actions by clicking on the links in this toolbar (Figure 1.4):

- Log in/out (Section 1.2.3: Logging In and Out) of the system
- Contact\textsuperscript{14} Connexions
- Report a bug (Section 1.3)
- Support\textsuperscript{15} Connexions through a tax-deductible donation

\textbf{1.1.2.4 My Account and Login Sidebar}

On the right side of the Connexions homepage is the My Account (Figure 1.5) sidebar. Its display depends on whether you are logged in or not.

The image on the left is displayed if you are not logged into Connexions or do not yet have an account. If you have a Connexions account, you can log into the system using this sidebar. In addition, there is a link to the page for requesting a new Connexions account.

\textsuperscript{14}http://cnx.org/aboutus/contact

\textsuperscript{15}http://cnx.org/aboutus/cnx_donate

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
account (Section 1.2.1: Account Requests), and a link to reset your password (Section 1.2.2: Request a New Password) (if you forgot it).

The image on the right is displayed if you are logged into Connexions. It features links to your MyCNX dashboard (Section 3.1), a link to change your account settings, and a link to modules that you have marked as your favorites. If you have any pending role requests or suggested edits, these will also be displayed in the My Account box.

1.1.2.5 Featured Content

The Featured Content (Figure 1.6) portal is located on the left side of the home page. It is used to showcase Connexions’ most popular and outstanding content, ranging from math and science collections to academic lenses.

Figure 1.6: Featured Content.

1.1.2.6 Find Content

You can start browsing Connexions’ modules and collections from the home page by using the Find Content (Figure 1.7) portal. The Search Content box is used to search for keywords related to the content you would like to find.

Figure 1.7: Find content portal.

There are also several links in the Find Content portal that will point you toward interesting Connexions content. First, place your mouse over the way you wish to browse the repository – either by Subject, Language, Popularity, or Title, etc. This will change the list of links on the right. Click on a link to begin browsing that particular selection of Connexions content.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
1.1.2.7 Create Content

It’s easy to get started creating content on Connexions. The Create Content portal, located just below the Find Content portal, contains links to help you begin authoring your own modules and collections.

1.1.2.8 Spotlight Sidebar

The Spotlight (Figure 1.8) sidebar features authors and their content, feedback from Connexions users, and events related to Connexions.

![Image not finished](a) ![Image not finished](b)

Figure 1.8: The Spotlight sidebar

1.1.2.9 News Sidebar

This sidebar contains links to news items about the latest developments with Connexions. This includes updates to the site and outside publications that feature Connexions.

1.2 Create an Account

1.2.1 Account Requests

To apply for a Connexions account, use the following steps:

1. Click on the Get An Account link in the My Account (Figure 1.9) sidebar.

![Image not finished](a)

Figure 1.9: The "My Account" portlet.

---

16 http://cnx.org/news/
17 This content is available online at <http://cnx.org/content/m37412/1.2/>.
18 http://cnx.org/join_form

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2. The "Request a Connexions Account" (Figure 1.10) page displays.

![Image not finished]

**Figure 1.10:** "Request a Connexions Account" page.

3. Type in your first and last name, e-mail address, and home page URL. The fields with red squares next to the field names are required.

4. Type in a user name.

5. Review the site license agreement and check the box next to **I have read the Connexions Site License and I agree to be bound by its terms.** You must agree to the license in order to receive an account.

6. Click **Request Account.** The "Account Request Complete" page displays. Connexions will send an e-mail to the address you entered in the step above. The e-mail contains directions for activating your new account.

### 1.2.2 Request a New Password

If you never receive a confirmation email, please check your "spam" or "junk" folder, and be sure to add cnx@cnx.org\(^{19}\) to your list of allowed contacts. To have a new password sent (or if you received a confirmation email but have since forgotten your password) you will need to request that a new password be sent.

1. Click on the **Forgot your password?**\(^{20}\) link in the My Account (Figure 1.9) sidebar. The Lost Password (Figure 1.11) page displays.

![Image not finished]

**Figure 1.11:** "Lost Password" page

2. Type in your user name and click the **Start password reset** button.

3. If you have also forgotten your user name, email cnx@cnx.org\(^{21}\) for help.

---

\(^{19}\)cnx@cnx.org  
\(^{20}\)http://cnx.org/mail_password_form  
\(^{21}\)cnx@cnx.org

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
1.2.3 Logging In and Out

To log into Connexions, enter your user name and password in the My Account (Figure 1.9) sidebar and click Log in.

To log out, click the Log out link located in the My Account (Figure 1.9) sidebar, or above the Search Box (Figure 1.12), in the upper right side of the Connexions home page.

![Image not finished]

Figure 1.12: Log out button above search bar.

1.3 Report a Bug

1.3.1 Submitting a Bug Report

In any software package you may encounter the occasional flaw with a feature or the documentation. Please bring these flaws, or bugs, to the attention of the Connexions team with the Report a Bug link, which is located on the Connexions Home page and on other Connexions web pages in the Personal Toolbar.

To report a bug, use the following steps:

1. Click on the Report a Bug (Figure 1.13) link at the top of the page.

![Image not finished]

Figure 1.13: Report a Bug link above the search bar.

The Bug Report Submission Form displays.

2. Enter a short summary of the problem.

3. Enter your name and e-mail address. These aren’t required, but it helps us to know who’s having trouble, and if you don’t enter an e-mail address we can’t get back to you with an answer.

---

22 This content is available online at <http://cnx.org/content/m11836/1.8/>.
23 <http://cnx.org/>
24 <http://cnx.org/bug_submit_form>

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4. The page you were on immediately before the bug form is entered automatically into the "Problem URL" field. If you didn’t come directly from the error page, please go back to the page where you had the problem and enter its address here. If the problem was on a module or collection, the object ID number is sufficient, e.g. m99999. NOTE: A content object’s ID appears in its URL or on the content’s metadata (Section 2.2) page.

5. Specify the nature of the error. Click the down arrow in the "Category" field and select the appropriate entry from the drop-down menu.

6. Specify the impact of the error. Click the down arrow in the "Severity" field and select the appropriate entry from the drop-down menu.

7. Enter a one-line summary of the bug in the "Task Title" field.

8. Enter an explanation of the circumstances of the error in the "Description" field. Include your actions and the system’s actions that led up to the error condition. Please include your browser and operating system. If possible, include step-by-step instructions for reproducing the bug.

9. Click Submit to enter the bug into the Connexions tracking system.

### 1.3.2 Helpful Tips for Submitting Bug Reports

Here are some tips for submitting error reports that will help us fix the bugs quickly.

- First, do not close the error message display until you have captured some information from it to include in your bug report. The error message contains important information. If you have left it to go to the Bug Report Submission Form or to a Help page, use the Back function on your browser to redisplay the error message.
- Include a copy of the error page. If the error displays an actual error message, please copy the text of that message and paste it into the bug report. It is even better to include the HTML source of the error message – some of our systems include important information in HTML comments. While the error message is displaying, right click your mouse and select the View Source or View Page Source option, depending on your browser, to display the HTML for the page. Then you can copy the HTML information and paste it into the bug report.
- Full descriptions of errors are very useful. If your error involved a specific module, include the name of the module. If you were in the editing interface (Section 3.1), include the workgroup and/or module with which you were working. If you were using the Collection Composer (Section 4.4.3), mention which collection you were working with, and so forth. Include the exact URL of the page on which the error occurred. Always include a description of what you were doing, if you can.

---


Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Chapter 2

Viewing and Finding Content

2.1 Modules and Collections

NOTE: Each piece of content has two major components available for viewing: the actual content itself, and the metadata page, containing information about the content, author, and more. When you first browse to a module or collection, you will be taken to the content page. To see the metadata, select the metadata link at the bottom of the page. Metadata are non-content information about a module, such as the module ID, license type, version number, creation date, revision date, authors, maintainers, copyright holders, module name, keywords, and abstract.

All Connexions content is located in the content repository, which can always be navigated to by clicking the Content tab at the top of the page. To allow you to easily access other parts of Connexions while viewing content (e.g. your MyCNX page or our Help documentation) the blue Connexions logo will always be located above the content you are viewing, including many of the links also on the home page (Section 1.1).

What’s in this module?
This module first describes the basic layout of the "Viewing Content" pages. There are additional features added to the page when you are viewing content in the context of a collection or a lens. Those extra features are described in the second and third sections of this module.

Jump to:

1. Basic Layout (Section 2.1.1: Basic Layout)
2. Collection Context (Section 2.1.2: Collection Context)
3. Lens Context (Section 2.1.3: Lens Context)

1This content is available online at <http://cnx.org/content/m11837/1.15/>.
2http://cnx.org/mycnx
3http://cnx.org/help

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 2. VIEWING AND FINDING CONTENT

2.1.1 Basic Layout

The components of the content viewing page include:

- Content Actions Bar (Section 2.1.1.1: Content Action Bar)
- Content Title Bar (Section 2.1.1.2: Content title bar)
- Featured Links (Section 2.1.1.3: Featured Links)
- Sidebar Portlets (Section 2.1.1.4: Sidebar Portlets)
- Footer (Section 2.1.1.5: Footer)

2.1.1.1 Content Action Bar

The Content Action (Figure 2.1) bar is a white horizontal bar above the module or collection title that includes a series of dropdown menus. All actions in the Content Action bar are also available as plain hyperlinks in the footer (Section 2.1.1.5: Footer) of the page.

Add to My Favorites

The Add to My Favorites dropdown allows you to quickly add content to your My Favorites4 lens. In addition to bookmarking content of personal interest, the My Favorites lens also allows readers to start reading from where they last left off5 within a bookmarked collection.

Add to a Lens

The Add to a Lens dropdown allows you to quickly add content to a lens (Section 2.3) you own.

Reuse/Edit

The Reuse/Edit dropdown menu allows you to quickly checkout the content to a Work Area (if you have permissions to edit the content) or derive a copy (Section 4.5.2: Creating Adaptations) of the content to create an adaptation (if you don’t have permissions).

---

4 http://cnx.org/help/accounts/myfavorites
5 http://cnx.org/help/accounts/trackprogress

 Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Download
The Downloads dropdown menu allows you to quickly access PDF and EPUB versions of the module and/or collection. The link to More downloads... will take you to all available downloads (Section 2.2.2: Downloads) for the content on its metadata page.

Print this Page
Click the Print this page button to print a printer-friendly version of the HTML page.

2.1.1.2 Content title bar

Title, Author, and Summary
The content title bar (Figure 2.2) displays the title, author(s), and summary provided by the author(s) for the content. More detailed metadata is on the Metadata (Section 2.2) page, which you can access through a link in the footer (Section 2.1.1.5: Footer) of the page.

Figure 2.2: Content title bar.

---

Facebook and Twitter
Connexions has rolled out Facebook "Like" and Twitter "Tweet" buttons on all of our content. If you are reading a module or collection that you like, you can like or tweet it directly from that page.

NOTE: Facebook gives much less feedback than Twitter. You won’t see a confirmation message after clicking "like", but it will show up on Facebook wall nonetheless.

Figure 2.3: Twitter and Facebook buttons in the content title bar.
Buttons exist in the content title bar (Figure 2.3) at the upper right corner the page, and also down at the bottom (Figure 2.4) in the footer. To like an entire collection of modules, go back to the START page of the collection by clicking the collection title. The description at the bottom of the page will say "Share collection" instead of "Share module", as it does on the module pages.

![Image not finished](a)

![Image not finished](b)

**Figure 2.4:** Twitter and Facebook buttons in the footer.

The Facebook and Twitter buttons give direct feedback to authors by letting them see instantly how many people have liked or tweeted their content.

### 2.1.1.3 Featured Links

In the Featured Links (Figure 2.5) portlet that appears to the right of the module content you can view links to examples, supplemental material, or prerequisite material provided by the module author. The importance of the links are shown by the number of bars in the box to the left of the link name. These links can be to material within Connexions, to included files, or to a website outside of Connexions. Featured Links are only included in the online views of content; the PDF and EPUB downloads (Section 2.2.2: Downloads) will not include these.

![Image not finished](a)

**Figure 2.5:** Featured Links portlet.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2.1.1.4 Sidebar Portlets

Related Material
You can view content related to the content you are displaying using the links in the Related Material (Figure 2.6) portlet that appears on the left side of the page. The links inside this box are specific to the content you are viewing at the time.

Figure 2.6: Links to related material

Included in the portlet are:

"Similar content": a list of similar content generated by the Rhaptos\textsuperscript{6} platform underlying Connexions. Click a title to display that content. Browsing the content listed here is a great way to find similar learning materials on a related subject.

"Collections using this material": lists any other collections that contain the current module. (This header only appears if you are viewing a module which is included in at least one collection.) You can click on the names of the listed collections to be taken to the start page of that collection.

Recently Viewed
Unless this is your first time viewing Connexions content on this computer or under this account, the Recently Viewed (Figure 2.7) portlet will be populated with modules, collections, and lenses you have recently browsed, so that you can easily jump back to them. This portlet can be collapsed by clicking on the triangle next to the title of the portlet.

Figure 2.7: "Recently Viewed" portlet.

\textsuperscript{6}http://rhaptos.org/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Lenses
If the content you are viewing is included in any public lenses, then the "Lenses" (Figure 2.8) portlet will appear on the left side of the page. This portlet lists all the various lenses which include the content being viewed.

Figure 2.8: Display lens information in a popup.

Hover over the name of a lens to see more information about it, including who created it, and the comment (if any) they wrote in their lens about the content. Clicking on the name of a lens will take you to the lens page (Figure 2.23) with the full list of content included in the lens.

Clicking on the tag item next to any of the lens names in the Lenses (Figure 2.9) portlet will expand the portlet to show all the tags the lens creator has associated with this piece of content.

Figure 2.9: Expand tags in the "Lenses" portlet.

Click one of these tags to view all other content within this lens associated with this tag. The link will take you to a view of the lens page with results filtered to show only the content associated with this tag.

Tags
If the content you are viewing is included in any public lenses, then you will also see a Tags (Figure 2.10) portlet on the left side of the page. This portlet lists a combination of all of the tags associated with this content from all the lenses it is included in.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Click on a tag here to see a listing of all the content in the Connexions repository that is included in a lens and has this tag associated with it. NOTE: Because only content contained in a lens can have a tag associated with it, this only finds content that an individual or organization has highlighted with this tag in a lens.

### 2.1.1.5 Footer

If you scroll to the bottom of the page, you will reach the content footer (Figure 2.11), which contains links to all the various actions in the Content Action bar (Section 2.1.1.1: Content Action Bar), plus more.

There is one further footer section (Figure 2.12), with links to various components of the metadata (Section 2.2) page (circled here in red).
2.1.2 Collection Context

How is a collection context set?
When you first open a Collection, the collection title page (Figure 2.13) displays. When you land on this page, your browser saves a cookie that lets it know you are viewing that collection. After that, whenever you view a module that is included in the collection, you will view that module in the context of the collection, explained below. This is true until you land on a different collection’s start page, at which point your browser will only display collection context information for modules in that second collection. Even if a module is actually included in 50 different collections by different authors, your browser will only show it in the context of the collection you are viewing. NOTE: To view collections correctly, make sure that you have enabled cookies in your browser.

Collection start page
Many portions of the collection start page are analogous to those when viewing a module, but some parts are new.

In addition to containing information about the collection itself, such as the title, authors, and summary, the center of the page displays a large Start ≫ link. Click Start ≫ to begin reading the collection from start to finish, module by module. Or, use the Table of Contents (Figure 2.14) portlet on the left side of the page to jump to any module within the collection.

Table of Contents
The Table of Contents (Figure 2.14) is shown on the collection start page itself, and on any module viewed within the context of that collection. Click a module name to jump to that module. You can use the triangles next to each subgrouping to open and close sections of the table of contents, or you can collapse the entire Table of Contents to save space by clicking the small triangle at the top of the portlet.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Collection Context bar
When you view a module in the context of a collection, there will be a collection context bar (Figure 2.15) above the module title bar to remind you that you are inside a collection. Analogous to the module title bar, this contains the collection title, authors, and summary.

Previous or Next Modules
Once you display a module from within a collection, you can move to the previous module or to the next module in the collection by clicking the ≪Previous or Next≫ links that appear in the upper right corner just below the collection context bar (Figure 2.15).

2.1.3 Lens Context
How is a lens context set?
When you first land on a lens page (Figure 2.23), your browser saves a cookie that lets it know you are viewing that lens. After that, whenever you view a module or collection that is included in the lens, you will view that content in the context of the lens, explained below. This is true until you land on a different lens’s start page, at which point your browser will only display lens context information for content that is in that second lens. Even if a module or collection is actually included in 50 different lens by different creators, your browser will only show it in the context of the lens you are viewing. NOTE: To view lenses correctly, make sure that you have enabled cookies in your browser.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 2. VIEWING AND FINDING CONTENT

2.2 Metadata

NOTE: Each piece of content has two major components available for viewing: the actual content itself, and the metadata page, containing information about the content, author, and more. When you first browse to a module or collection, you will be taken to the content page. To see the metadata, select the metadata link at the bottom of the page.

To display the metadata for the module or collection, scroll down to the footer, where it says More about this module/collection.

You can click on Metadata (Section 2.2.1: Metadata), Downloads (Section 2.2.2: Downloads), or Version History (Section 2.2.3: Version History), highlighted in the figure (Figure 2.17) in red. You can also click on How to reuse/cite and attribute this content (Section 2.2.4: Reuse and Attribution). These links all take you to different parts of the metadata page.

This content is available online at <http://cnx.org/content/m37413/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2.2.1 Metadata

The top of the metadata page includes a link (Figure 2.18) to return to the Viewing Content (Section 2.1) page.

The Metadata (Figure 2.19) section summarizes all the metadata the author included with his/her content. This includes the module title, language, summary, subject, keywords, as well as the authors, copyright holder, and maintainer Roles. Other sections are auto-generated by the software underlying Connexions, like the unique content ID, the latest version, the publication date, and the last revision date.

2.2.2 Downloads

The Downloads (Figure 2.20) section of the metadata page gives you easy access to all the available downloadable formats for the content you are viewing.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2.2.2.1 PDF Files

Each module and collection has a PDF document version that can be downloaded for viewing without an internet connection. The PDF version of a collection is formatted for printing with page numbers, headers, footers, and numbered headings and is identical to the hardcopy printed version of the collection. PDF files can be downloaded from the Download dropdown in the Content Actions bar (Section 2.1.1.1: Content Action Bar) at the top of content pages, at the bottom of content pages, and from a content’s metadata page.

2.2.2.2 EPUB Files

EPUB is a standard electronic book (ebook) format that can be read on a computer and on most small devices (smartphones, iPad, e-book readers, etc.). EPUB files can be downloaded from the Download dropdown in the Content Actions bar (Section 2.1.1.1: Content Action Bar) at the top of content pages, at the bottom of content pages, and from a content’s metadata page. For more information on using EPUB files, see our EPUB help8 page.

2.2.2.3 XML

XML provides a machine-readable representation of a module or collection.

Collection XML, called CollXML⁹, contains a collection’s metadata (authors, maintainers, keywords, subject list, collection abstract, print parameters, featured links, etc.) and its structure (links to modules contained in the collection, featured links for each module, subcollections and the modules in the subcollection). The CollXML cannot be reimported into collection editor.

Module XML, called CNXML¹⁰, contains the metadata for a module and the content of the module. The CNXML can be imported into the module editor.

XML files for a collection or module can be downloaded from their respective metadata pages.

2.2.2.4 Source Export ZIP

The module source export ZIP file contains the module XML (CNXML) file along with any media files that are part of the module. This ZIP can be imported into the module editor. It can be downloaded from a module’s metadata page or be exported from the module editing interface.

---
⁸http://cnx.org/help/viewing/epub-help
⁹http://cnx.org/help/authoring/xml#collxml
¹⁰http://cnx.org/help/authoring/xml#cnxml

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
The collection source export ZIP contains the collection XML (CollXML) as well as the module XML and module media files for each module it contains. It is intended for developers who want to create their own version of a collection by transforming the XML into a different look. It can be downloaded from a collection’s metadata page or from the collection editing interface.

2.2.2.5 Offline ZIP

Collections and modules have a downloadable ZIP file that contains all of the information and contents of the collection or module. The ZIP includes:

- start.html which displays the collection or module and a table of contents in your browser
- HTML files of the content that can be viewed offline.
- All of the images and other media (videos, audio files, documents, applets, etc.) included in the module. The collection Offline ZIP contains the media files of all the modules in that collection.
- The content’s XML file. The collection Offline ZIP also contains the XML files for each of the modules in the collection.
- The Offline ZIP files each have a README file that guides users through the contents of the ZIP. Developers can view the exact contents of the README from the links below.
  - Collection README File\(^{11}\)
  - Module README File\(^{12}\)

Offline ZIP files can be downloaded by choosing More downloads... from the Download (Download, p. 12) dropdown at the top of content pages, at the bottom of content pages, and from a content’s metadata page.

2.2.3 Version History

The Version History (Figure 2.21) section of the metadata page contains an archive with links to all of the past versions of the content.

\(^{11}\)https://trac.rhaptos.org/trac/rhaptos/browser/rhaptos/packages/Products.RhaptosPrint/trunk/Products/RhaptosPrint/epub/static/README-COLLECTION-OFFLINE.txt

\(^{12}\)https://trac.rhaptos.org/trac/rhaptos/browser/rhaptos/packages/Products.RhaptosPrint/trunk/Products/RhaptosPrint/epub/static/README-MODULE-OFFLINE.txt

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
When a new module is published Connexions assigns a revision number to that version of the module. If it becomes necessary to update the module, the author checks it out from the repository, edits it, and publishes it again. An incremented revision number is assigned to the updated version. Connexions stores all this revision information and you can view it on the History page for the module.

### 2.2.4 Reuse and Attribution

The Reuse and Attribution (Figure 2.22) section of the metadata page reminds you of the attribution requirements of the license the content is published under. It also gives a list of bibliographic citations in various popular styles, like the APA, MLA, or Chicago style.

### 2.3 Lenses

#### 2.3.1 What is a lens?

Lenses enable both organizations and individuals to give their stamps of approval to content in the repository, allowing for user-driven quality control of modules and collections. Through these lenses, users can provide their own **tags** and **comments** for items in the repository. Lenses can also be used as "bookmarks" within the repository to keep track of related or otherwise interesting content.

---

13This content is available online at [http://cnx.org/content/m37415/1.2/](http://cnx.org/content/m37415/1.2/).

Available for free at Connexions [http://cnx.org/content/col10151/1.27]
A list of all publicly-viewable lenses can be found at under the Lenses\(^\text{14}\) tab. Lenses are available in four different forms:

- **Endorsement Lenses\(^\text{15}\)** - Organizations, such as professional societies, can create endorsement lenses containing content they have carefully reviewed and deem to be of high quality. Organizations use their own criteria for endorsement and are encouraged to describe their selection process on the lens home page.
- **Affiliation Lenses\(^\text{16}\)** - Organizations can also create affiliation lenses identifying content created by members of that organization (but not necessarily reviewed). Though not as rigorous as an endorsement lens, content in an affiliation lens is assumed to be of a quality on par with other members of that organization.
- **My Favorites Lens** - Each user account has a private My Favorites\(^\text{17}\) lens which can be used to keep track of a user’s favorite modules and collections. In addition to bookmarking content of personal interest, the My Favorites lens also allows readers to start reading from where they last left off\(^\text{18}\) within a bookmarked collection. You can access your personal My Favorites lens by selecting the MyCNX tab\(^\text{19}\) and clicking on the My Favorites link (login required).
- **Member Lists\(^\text{20}\)** - An account holder can create member list lenses to organize related content, such as to keep track of referenced source materials or to use as a course reading list for students.

All lenses except for the My Favorites lens can either be made privately viewable for personal use only or publicly viewable for anyone to see. The My Favorites lens is always private.

### 2.3.2 Using lenses

Clicking on the name of a lens will take you to its lens page (Figure 2.23), which includes information about the lens (and the organization, if the lens was created by an organization), a listing of all the content within that lens, and a tags (Figure 2.24) portlet on the left side of the page.

\(^{14}\)http://cnx.org/lenses  
\(^{15}\)http://cnx.org/endorsements  
\(^{16}\)http://cnx.org/affiliations  
\(^{17}\)http://cnx.org/help/accounts/myfavorites  
\(^{18}\)http://cnx.org/help/accounts/trackprogress  
\(^{19}\)http://cnx.org/mycnx  
\(^{20}\)http://cnx.org/memberlists

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 2. VIEWING AND FINDING CONTENT

2.4 Finding Content

Learn to

- Search (Section 2.4.1: Searching the Content Repository) the repository.

---

This content is available online at <http://cnx.org/content/m37430/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
• Browse (Section 2.4.2: Browsing the Content Repository) the repository.
• Sort (Section 2.4.3: Sorting Criteria) by various criteria.
• Discover (Section 2.4.4: Discovering Content) quality content in other ways.

2.4.1 Searching the Content Repository

You can search the content repository for a specific module or a collection using the Search button and text box in the upper right of the Connexions web pages. Type in a name, keyword, collection title, module title, text string, or object ID in the Search text box and click Search. Connexions will display a list of all modules and collections that match your entry. You can sort (Section 2.4.3: Sorting Criteria) and filter these results by various criteria. To view a module or collection, click its title.

2.4.2 Browsing the Content Repository

Select the Content tab (above the You are here Breadcrumb bar) to display the browsing interface screen. This page also allows you to search (Figure 2.26) for modules or collections by title, author name, or subject area.

Or you can use the Connexions Browse (Figure 2.27) feature to display a listing of all content by subject, title, author name, keyword, popularity, language or revision date.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
1. **Browse** by selecting the appropriate item under the **Browse All Content** heading on the tab. (The last filter criterion, "All Collections", will display a listing of only collections instead of searching individual modules.)

2. **Refine** your results. The refinement differs depending on what you chose to browse by in Step 1.

3. **View** a module or collection by clicking on its title.

On the right side of the browsing interface screen is the Repository (Figure 2.29) portal. Use this to jump to a random module or random collection in the repository.

### 2.4.3 Sorting Criteria

There are several pages on Connexions where you might see a list of content like those shown below. Some of the pages are the results of searching (Section 2.4.1: Searching...
the Content Repository) or browsing (Section 2.4.2: Browsing the Content Repository) Connexions, but you will also see these lists through lens pages (Figure 2.23), author profiles (Through member profiles, p. 31), and elsewhere. This interface allows you to choose multiple views of the content listings, sort them by various criteria, and even quickly checkout or derive a copy (Section 4.5.2: Creating Adaptations) of content to one of your Work Areas.

A typical listing is shown below (Figure 2.30). Notice the left side includes basic metadata (Section 2.2) like the title, author, summary, etc. and the right side includes further metadata which you can sort by (Sort by, p. 29).

---

**Image not finished**

*Figure 2.30: Detailed search returns.*

---

**Sort by**

You can choose a new sorting criteria from the *Sort by* dropdown at the top of the listing. You can sort by:

- Relevance - to what you searched by
- Popularity - based on the number of times the content has been visited in ratio to the total number of page hits of all Connexions content.
- Language - as specified by the author.
- Revision date - to sort the most recently updated content to the top of the list.
- Title - to list the content alphabetically
- Type - to group collections together at the top, and modules together at the bottom.

---

**Image not finished**

*Figure 2.31: Search returns with "Sort by" list shown.*

---

**Views**

On the left side above the content listing, you should see View: **Detail | Compact | Statistics.** Clicking on one of these changes the view. Often, the default view is "Detail".

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
When you click on "Compact", you can view a compact (Figure 2.32) listing of the content, which only shows the title and author on the left, and your current sorting criterion on the right.

Finally, the statistics (Figure 2.33) view gives you a bit more information behind Connexions "Popularity" ranking. You can see the total number of times the content has been viewed, the average number of views/day, how high the content is ranked (a rank of "1" means that content is the most-viewed content in the Connexions repository), and the rank percentile (a percentile of 98.72% means that 98.72% of the content in the repository is ranked lower than the this content).

At the top of the statistics view are links that allow you to download a spreadsheet of the statistics, either for the past week or for all time.

2.4.4 Discovering Content

If you are looking for quality content on a certain subject, or tailored for a certain audience, there are other ways to find what you are looking for.

Through content
You can use the Related Material (Related Material, p. 15) portlet to find content similar to the content you are viewing. The "Related Material" portlet also lists other collections which include this content.
Through lenses
When viewing content, you can use the Lenses (Lenses, p. 15) portlet to find lenses which include this content. Those lenses are likely to have similar content of high quality. You can also use the Tags (Tags, p. 16) portlet to search the repository for other content included in lenses that is associated with this tag.

You can always browse our list of public Lenses\(^\text{24}\) to find lenses related to the subject you are searching.

Through member profiles
If you enjoyed one module or collection, you might enjoy more content by that author. Click the author’s name to go to the author’s profile (Figure 2.34), where you will find links to all of their modules, collections, and lenses, as well as links to the profiles of any coauthors whom they authored content with.

---

**Figure 2.34:** Author profile, with link to author’s statistics.

---

In this figure (Figure 2.34), a link (view statistics) is circled in red. Click this to go directly to the popularity statistics (p. 30) of this author’s content. Rank them by popularity to find the content by this author most visited.

---

\(^{24}\)http://cnx.org/lenses

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Chapter 3

The Editing Interface

3.1 MyCNX Tab

You can begin managing and creating new content from the editing interface component of Connexions, known as MyCNX. It is also called your dashboard because it offers quick and easy access to various parts of the editing interface. You can access it by clicking on the MyCNX tab located at the top of every Connexions page.

The MyCNX Home page is a gateway to all of Connexions’ authoring areas, such as your Personal Workspace (Section 3.3.1.1.1: My Workspace) and your Workgroups (Section 3.3.1.1.2: Workgroups). It is also a quick way to manage and edit your Lenses (Section 3.1.2: Access lenses), including your My Favorites list. You can access the MyCNX home page by clicking on the MyCNX tab at the top of any Connexions page. You must obtain a Connexions account (Section 1.2.1: Account Requests) and log in to the site (Section 1.2.3: Logging In and Out) in order to access MyCNX.

3.1.1 Create and Edit Content

The Create and edit content section of the MyCNX Home page is the quickest way to start authoring in Connexions. Clicking the Create a new module link allows you immediately begin creating a new module and choose later which Work Area to keep it in (likewise for the Create a new collection link). Underneath these links is the “Last modified” area, which provides an easy way to resume working on content that you have already created.

---

1This content is available online at <http://cnx.org/content/m10885/2.23/>.
2http://cnx.org/mycnx

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 3. THE EDITING INTERFACE

3.1.2 Access lenses

To start creating and editing lenses (Section 2.3) from the MyCNX home page you can follow the links under the Access lenses (Figure 3.2) section. This area provides users with a list of their lenses. Click the lens name to view the contents of the lens as if you were a normal viewer; click Edit to change the lens properties.

NOTE: If you wish to add new content to a lens, navigate to the content in the repository and choose Add to a lens... in the Content Actions bar (Section 2.1.1.1: Content Action Bar) at the top of the content.

If you have several lenses, you can see them all by clicking the More link at the bottom of the section. You can also create a new lens by clicking on the Create a new lens link.

3.1.3 Guides and Tutorials

The MyCNX Home page provides several links to guides and tutorials (Figure 3.3) that can help you begin authoring content in Connexions.
3.1.4 Recently Viewed

Finally, the Recently Viewed (Figure 3.4) portlet located on the MyCNX Home page lists the collections, modules, and lenses that you’ve most recently visited.

3.1.5 Navigating to Work Areas

The left side of the McCNX homepage is filled by the blue navigation sidebar (Section 3.2.1: Navigation Bar). This sidebar remains on the left side of the page as long as you are within the editing interface portion of Connexions, and allows you to switch among your Work Areas and content easily and quickly.

3.2 Navigating the Editing Interface³

3.2.1 Navigation Bar

The navigation sidebar (Figure 3.5) is located on the left side of all editing interface pages. You can always use the link at the top of the sidebar to return to the MyCNX Home page (Section 3.1). You can collapse and expand the different sections by clicking the small triangle located next to the heading.

³This content is available online at <http://cnx.org/content/m37463/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 3. THE EDITING INTERFACE

The "By Type" section allow you to find all of the Modules, Collections, and Lenses you’ve created by clicking on the respective link. This is a great tool if you can’t remember in which Work Area you saved your content. The "By Location" section displays your Personal Workspace and all of the workgroups you belong to; this section is initially collapsed on all pages except for the MyCNX Home page.

3.2.2 Show/Hide Toolbars

When editing a module or collection, a useful Show/Hide Sidebars (Figure 3.6: Using the Show/Hide Sidebars icon) icon is featured in the upper right hand corner. When you click on the icon, the right and left authoring sidebars are hidden and the content pane is expanded. This is especially useful when editing modules using Edit-In-Place, or when using a netbook or other small-screened device.

Once the Show/Hide Sidebars icon has been clicked, you can reduce the editing area and display the sidebars by clicking it again.

Figure 3.6: Demonstration of the Show/Hide Sidebars icon (a) Click on the "Show/Hide Sidebars" icon... (b) ...and the editing area fills the page. Click on the icon again to return to your normal view.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
3.3 Work Areas

3.3.1 Overview

3.3.1.1 Connexions Work Areas

Connexions provides two types of areas in which to work on content: My Workspace (Section 3.3.1.1.1: My Workspace) and Workgroups (Section 3.3.1.1.2: Workgroups). You can create, edit, and manage content in both work areas. The difference between the types of work areas is user access. "My Workspace" is a private work area. Only you can access it and view or modify its contents. "Workgroups" are for collaborative projects. Their contents can be accessed by a group of users and each of the group members can view or modify its contents. "Workgroups" must be created by you. You can create multiple workgroups, one for each project. Both of these types of work areas are described in this module.

3.3.1.1.1 My Workspace

Every person with a Connexions account has their own "My Workspace". It was created automatically when their Connexions account was set up. "My Workspace" is your personal work area in Connexions where you can work independently of anyone else. In this work area you can keep your own modules, files, and collections that no one else can view or modify. Many authors prefer to work on modules or collections in "My Workspace" before moving them to a workgroup (Section 3.3.1.1.2: Workgroups) where a group of collaborators can make comments or edit the material.

NOTE: It is best to keep only one working copy of any one content; therefore be sure to use "Cut" and "Paste" to move content to a Workgroup instead of "Copy" and "Paste".

You can display your personal work area from a Connexions web page by clicking on the My Workspace link in the navigation sidebar (Section 3.2.1: Navigation Bar) on the left side of the page.

3.3.1.1.2 Workgroups

A workgroup is a private "scratch area" where a group of Connexions authors can collaborate on a common set of modules. This allows authors working together to see their progress without letting the rest of the world see their work.

WARNING: Remember that workgroups are not a substitute for communication between collaborators. The members of a workgroup must communicate openly because all the members in a workgroup have the same level of privilege. Any

---

4This content is available online at <http://cnx.org/content/m37462/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
member in a workgroup may add or remove another member and modify, add, or remove content. The creator of a workgroup has no extra privileges.

Modules, collections, and other uploaded files can be included in each workgroup. The members of the workgroup can create new content in the workgroup, or can checkout existing content (or derive a copy (Section 4.5) of content) from the Content Repository. Once content is placed in the workgroup, any member of the workgroup may modify it. To publish modified content, a person must have the maintainer role for that module or collection. For more on this see Editing Module Roles.

The names of the workgroups of which you are a member are listed in the navigation sidebar (Section 3.2.1: Navigation Bar) on the left side of the page. You can display the contents of a workgroup by clicking on the name of the workgroup.

3.3.1.2 Creating a New Workgroup

To create a workgroup, use the following steps:

1. Click Create a workgroup in the "Workspaces" sidebar. The "Create Workgroup" screen displays.
2. Type the name of the workgroup in the "Name" field.
3. (Optional) Type a group e-mail address in the "Email" field. This should be a common e-mail address or mailing list that is shared by all members of the workgroup.
4. (Optional) Type a description of the workgroup in the "Description" field. (optional)
5. Click Create. The screen for the new workgroup displays.

Since it is a new workgroup, it has no content and you are its only member. See Managing Work Area Contents (Section 3.3.2) for a description of how to add content to the new workgroup. See Managing Workgroup Members (Section 3.3.4) for a description of how to add members to the new workgroup.

3.3.1.3 Deleting a Workgroup

A workgroup is deleted when it no longer has any members. If you want to delete a workgroup, you must remove all the members of the workgroup, including yourself. To delete a workgroup, use the following steps:

1. Click on the green "Members" tab at the top of the Workgroup.
2. Select the checkbox in the "Remove" column next to every member’s name, including your own.
3. Click Remove selected users from group. The "Delete Workgroup" confirmation screen displays.
4. Click Yes, Delete it. All members are removed from the Workgroup, the Workgroup is deleted, and the Workgroup name is removed from the navigation sidebar (Section 3.2.1: Navigation Bar) of all members of the Workgroup. Any published

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
content located in the Workgroup will not be deleted; to edit or modify it, you will need to check it out to another Work Area.

3.3.2 Contents

To display a Work Area, click on the link in the navigation sidebar (Section 3.2.1: Navigation Bar). If this is the first time you have displayed this Work Area or if you have just created the Work Area, there will be no content in the Work Area.

The display for your Personal Workspace (Section 3.3.1.1.1: My Workspace) has a green "Contents" and "Properties" tab at the top. Workgroups have "Contents", "Properties", and "Members" tabs. The "Content" tabs for both work areas are the same, except for the title at the top of the page. The following figure (Figure 3.7) shows the "Contents" tab for a Personal Workspace.

![Image not finished]

Figure 3.7: "Contents" tab for "My Workspace".

You can perform the following actions from the "Content" tab:

- Creating New Items (Section 3.3.2.1: Creating New Items)
- Checkout Content from the Repository (Section 3.3.2.2: Checkout Content from the Repository)
- Viewing the Work Area Contents (Section 3.3.2.3: Viewing the Work Area Contents)
- Removing Items from a Work Area (Section 3.3.2.4: Removing Items from a Work Area)
- Renaming Items in a Work Area (Section 3.3.2.5: Renaming Items in a Work Area)
- Cutting or Copying Items in a Work Area (Section 3.3.2.6: Cutting or Copying Items in a Work Area)

3.3.2.1 Creating New Items

To create new modules or collections in the work area, use the following steps:

1. Click the down arrow in the text box next to the Create New Item button. A drop-down list of valid item types appears.

---

5This content is available online at <http://cnx.org/content/m37476/1.3/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2. Select either the "Collection" or "Module" item type and click Create New Item. The "New content: Licensing" screen displays.

3. Select the checkbox to agree to the license and click Next. If you selected "Collection", the "Metadata" tab on the "Edit Collection" screen displays. See the How to Use the Collection Composer (Section 4.4.3) for a description of how to edit a collection. If you selected "Module", the "Metadata" tab on the "Edit Module" screen displays. See Editing Modules (Section 4.1) for a description of how to edit a module.

To create new files in the work area, use the following steps:

1. Click the down arrow in the text box next to the Create New Item button. A drop-down list of valid item types appears.
2. Select the "Image/Other File" item type and click Create New Item. The "Edit File" screen displays.
3. Click Browse and locate the file to be added.
4. Select the item and click OK to upload the file.
5. Click Save to add the file to the work area.

The "File" item type allows you the freedom to upload materials that you want to use in your content. These materials must exist before you can import them into Connexions. When you create the file in the work area, you are actually creating an instance of the item in Connexions.

When you create a new image file, once you click Save a preview of the image appears and a message that your image has been saved displays. When you create any other type of new file and click Save, information about the file displays and the Click here to get the file link appears.

3.3.2.2 Checkout Content from the Repository

Besides creating new content (Section 3.3.2.1: Creating New Items), you can also checkout published content from the Content Repository. This places a working copy into one of your Work Areas for editing. In this way you can reuse information that appears elsewhere in Connexions. Once you make changes to the content and republish it, the published content updates to the new version.

NOTE: It is possible to check out multiple copies of an item by checking it out to multiple work areas. If this happens, changes made to one working copy will not be automatically integrated into the other working copies. Because of this, it is highly recommended that you only keep one working copy of a published work. If you need to move the item to work on it in a new area, do not check out a second copy, but instead cut and paste (Section 3.3.2.6: Cutting or Copying Items in a Work Area) the existing copy into the new area.
You can checkout content in one of several ways:

1. Directly from the content viewing page. (Section 3.3.2.2.1: From the content page)
   This method also allows you the option to easily derive a copy.
2. From the search/browse returns listing. (Section 3.3.2.2.2: From the search/browse returns listing) This method allows you to checkout multiple items at once.

3.3.2.2.1 From the content page

Search (Section 2.4.1: Searching the Content Repository) or browse (Section 2.4.2: Browsing the Content Repository) to the content you want to checkout. Click on the title of the content you want to copy.

Near the top right corner, select the menu reading **Reuse / Edit** and click the "Reuse or edit" link for the content you want to copy. (If you are viewing a module in the context of a collection, there are "Reuse or edit" links for both the collection and the module.)

---

**Image not finished**

**Figure 3.8:** The ‘Reuse / Edit’ menu in the Content Actions bar.

---

You will be prompted to log in if you have not already done so. **If you do NOT have permission to edit the content,** you will be prompted either to:

- **Derive a copy of the content.** Deriving a copy (Section 4.5) allows you to adapt this content and then publish your adaptation.
- **Or check it out anyway.** You might choose to check out the content, even though don’t have permission to publish changes. You may ask to be a co-author later, derive a copy later (Section 4.5.2.2: From a checked out copy), or just want to see how the content is constructed.

---

**Image not finished**

**Figure 3.9:** The pop-up with options to derive a copy or check it out anyway.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Click **Check it out anyway** to checkout the content.

You will be prompted to select a work area to edit the content in.

![Image not finished](image)

**Figure 3.10**: ‘Choose a work area to check out content to’ popup.

If you leave the "Edit the content now" checkbox selected, you will be redirected to the work area in order to immediately start editing the content.

### 3.3.2.2.2 From the search/browse returns listing

1. Search (Section 2.4.1: Searching the Content Repository) or browse (Section 2.4.2: Browsing the Content Repository) the repository.
2. Select the checkboxes next to the items you want to checkout, or use the checkbox in the title bar to select all the results.
3. Click the down arrow in the "Add selected content to" text box at the top of the returns listing. A drop-down list of all the work areas to which you belong displays. Select a work area from the list.
4. Click **Add**. The selected items will be added to the work area you selected and the "Contents" tab for that work area will display.

![Image not finished](image)

**Figure 3.11**: Adding the results of a search to your Personal Workspace.

### 3.3.2.3 Viewing the Work Area Contents

A work area can contain modules, collections, and files (such as images). The current contents of a work area can be seen on the "Contents" tab for the work area. The following information is displayed for each item in the work area:

- **Title** - This column displays the title and the filename or ID of the item you have created or added to the work area group.
- **Type** - This column displays the type of item listed. The types are: files, modules, and collections.
• Size - This column displays the size of the item in kilobytes. No size is displayed for modules or collections.
• Modified - This column displays the date and time the item was last modified. If the item is new, this column displays the creation date and time.
• Status - This column displays the current status of the item. There are four possible statuses:
  - Published - The item has been checked into the Content Repository and is now accessible by the public.
  - Checked Out - The item was copied from the Content Repository and placed in your work area for editing purposes.
  - Created - The item has been created and placed in your work area, but no copy of the item exists in the Content Repository.
  - Modified - The item has been checked out and updated, and the changes have been saved. When you create a new item and make changes to it, its status changes from "Created" to "Modified".

3.3.2.4 Removing Items from a Work Area

To remove an item from a work area, select the checkbox next to the title of the item you want to remove and click Remove. You can select multiple items from the work area and remove them in one operation.

3.3.2.5 Renaming Items in a Work Area

To rename an item in a work area, use the following steps:

1. Select the checkbox next to the title of the item you want to rename and click Rename. The "Rename item" screen displays.
2. Enter the new name for the item in the "New Name" field.
3. Enter the new title for the item in the "New Title" field.
4. Click Rename All. You can rename multiple items at one time by selecting more than one item by clicking on the work area contents screen.

The items displayed on the "Contents" tab are updated to reflect the changes.

3.3.2.6 Cutting or Copying Items in a Work Area

To copy and paste an item in a work area, use the following steps:

1. Select the checkbox next to the title of the item you want to cut or copy and click Cut or Copy. Cut removes the original item at the end of this process. Copy leaves the original item in its present location.
2. Display the location where you wish to paste the object, such as another work area, and click Paste. If you paste a copied item into the workgroup where the original item resides, you will have a copy of the original item.

3.3.3 Properties

You can edit certain properties of a work area, such as the name of a workgroup or the description of your Personal Workspace. These changes are accessed on the work area’s "Properties" tab at the top of the page.

3.3.3.1 Editing My Workspace Title and Description

To edit the title or description of My Workspace, use the following steps:

- Click My Workspace to display your personal work area screen.
- Click on the Properties tab. The "Edit Folder Properties" screen displays.
- Type a new title or name for your personal work area in the "Title" field.
- Type a description of your personal work area in the "Description" field.
- Click Save.

3.3.3.2 Editing a Workgroup Title, E-mail Address, and Description

To edit the title, e-mail address, and description of a workgroup you of which you are a member, use the following steps:

- Click the name of the Workgroup in the navigation sidebar (Section 3.2.1: Navigation Bar) to display that Workgroup.
- Click on the Properties tab. The "Properties" screen displays.
- Type a new title or name for the workgroup in the "Name" field.
- Type a group e-mail address in the "Email" field. This should be a common e-mail address or mailing list that is shared by all members of the workgroup.
- Type a description of the workgroup in the "Description" field.
- Click Update.

3.3.4 Members

What are group members?

A Workgroup is a collaborative Work Area (Section 3.3.1) where content may be reviewed or edited by the Workgroup members prior (or after) being published. A Workgroup is not visible to the general public: you can become a Workgroup member by being added to the Workgroup by an existing member.

---

6This content is available online at <http://cnx.org/content/m37467/1.2/>.
7This content is available online at <http://cnx.org/content/m37472/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
NOTE: Only a person who has a Connexions account (Section 1.2.1: Account Requests) can be a member of a Workgroup.

Workgroup members can perform any of the allowed actions to manage Workgroup contents (Section 3.3.4), including adding and removing modules and collections, viewing content, modifying content, and renaming content in the Workgroup. For this reason, it’s important to communicate with other group members so that no content is modified or deleted that shouldn’t be. However, content may not be published by a group member unless that group member also has the maintainer role for that content.

**Workgroup Uses**

1. Solicit reviews and comments on content before you publish it by moving the content to a Workgroup and adding your reviewers as Workgroup members.
2. Keep projects separate by creating a different Workgroup for each project.
3. Keep your Personal Workspace less cluttered by moving contents to different Workgroups.

The names and e-mail addresses of the members of the workgroup are listed on the Members tab (Figure 3.12). This tab is not available in the "My Workspace" work area since that is your personal work area. You are the only person who can access that work area.

---

*Figure 3.12: Workgroup "Members" tab.*

---

You can perform the following actions from the "Members" tab:

- Adding Members to a Workgroup (Section 3.3.4.1: Adding Members to a Workgroup)
- Removing Members from a Workgroup (Section 3.3.4.2: Removing Members from a Workgroup)
- E-mailing the Entire Workgroup (Section 3.3.4.3: E-mailing the Entire Workgroup)
- E-mailing an Individual Member (Section 3.3.4.4: E-mailing an Individual Member)
- Leaving a Workgroup (Section 3.3.4.6: Leaving a Workgroup)

Joining a workgroup requires an invitation. See Joining a Workgroup (Section 3.3.4.5: Joining a Workgroup) for more information.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
3.3.4.1 Adding Members to a Workgroup

To add new members to a workgroup you are in, use the following steps:

1. Type the name of the person you wish to add to the list of members in the text box next to the Search button.
2. Click Search. A list of the Connexions account holders that match the search string appears. If the search returns no results, try the search again using just a username or a last name.
3. Select the checkbox next to the name of the person or persons you wish to add to the workgroup.
4. Click Add selected users. The "Current Group Members" table is updated to include the new member or members. When that user logs on, the Workgroup will be listed in the blue navigation sidebar (Section 3.2.1: Navigation Bar).

3.3.4.2 Removing Members from a Workgroup

To remove members from your workgroup, use the following steps:

1. Select the checkbox in the "Remove" column for the member you wish to remove from the workgroup.
2. Click Remove selected users from group. The "Current Group Members" table is updated to show the removal of the member.

3.3.4.3 E-mailing the Entire Workgroup

To e-mail the entire workgroup, use the following steps:

1. Click on the email group link next to the table title "Current Group Members". Your e-mail application is launched displaying a new message addressed to the group members.
2. Type in the text of the message and send it.

3.3.4.4 E-mailing an Individual Member

In addition to e-mailing the entire group, you can e-mail an individual member of the workgroup. To e-mail an individual member, use the following steps:

1. Click on the address in the "E-mail" column next to the name of the member to which you want to send an e-mail. Your e-mail application is launched displaying a new message addressed to the group member.
2. Type in the text of the message and send it.
3.3.4.5 Joining a Workgroup

To join a group, a current member of the group must add you to the workgroup member list. If you wish to join a workgroup, contact a member of that workgroup and ask them to add you to the workgroup.

3.3.4.6 Leaving a Workgroup

If you leave a workgroup or remove yourself from the member list, you are no longer a member of that workgroup. To become a member again, a current member of the workgroup must add you to the list of members again. To leave a workgroup, use the following steps:

1. Display the "Members" tab for the workgroup you wish to leave.
2. Select the checkbox in the "Remove" column next to your name.
3. Click Remove selected users from group. The "Leave Workgroup" confirmation screen displays.
4. Click Yes, Leave. You are removed from the workgroup and the workgroup name is removed from the "Workspaces" sidebar on your Connexions web page.

Once every member of a group has left the group, the group will be deleted. The last member to leave will be asked to confirm this step.
Chapter 4

Authoring Content

4.1 Create new module

4.1.1 What is a Module?

A module is the basic building block of a Connexions course, textbook, or other type of collection. You can think of it as a small knowledge chunk that addresses a single topic or a specific aspect of a topic. Every author determines the size of their module, ranging from a few paragraphs to an entire textbook chapter. To a student or reader viewing a course or collection, a module is simply a web page in the collection. Modules allow readers to follow the information path arranged by the author or instructor or to branch off and discover their own path. To an instructor putting a collection together, having topics in different modules allows easy selection and arrangement of the information. An instructor can include existing modules from other courses or other academic disciplines that are important to the presentation of the course subject.

4.1.2 Creating Modules

You create and edit modules within one of your Work Areas (Section 3.3.1). There are two ways to create modules:

1. Create a new module from scratch. (Section 3.3.2.1: Creating New Items)
2. Create an adaptation (Section 4.5) of an existing module in the repository.

Below are the steps to create a module from scratch.

You can create a new module from the "Create New Item" dropdown in one of your work areas (Section 3.3.1), or you can click the "Create a new module" link on the MyCNX home page:

---

1This content is available online at <http://cnx.org/content/m10887/2.31/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 4. AUTHORING CONTENT

Figure 4.1: Go the "MyCNX" tab and select the "Create a new module" link.

From there you must agree to the license for this module, select a work area in which to edit the content, and then enter the module’s initial metadata.

After accepting the license, you will be asked to enter some basic information, or metadata, about your module before getting started. (If you haven’t already selected a work area in which to work on the module, you will be asked to do so.) The basic metadata include:

• A title for the module
• The primary language used in the module
• The subject categories that apply to the module
• Terms that describe the most important topics in the module
• A brief description of the module. You may markup the summary section with inline CNXML tags.

Figure 4.2: Enter basic information about your module here.

Click the Save button. The metadata entries are saved and the "Contents" tab displays with the title of the collection. This tab is where you do the actual editing of the module content.

4.1.3 Editing Modules

Once you have opened a new or adapted module within one of your Work Areas, you will need to edit the module to add content and files (such as images). These operations are performed with the links and buttons on the "Manage Files" screen or the "Module Status" sidebar. The recommended sequence to perform these operations in is described in this module and appears in the following list:

1. Edit the CNXML (Section 4.3.1.1) file.
2. Add (Section 4.3.2.1) any other files that you want to associate with the module.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
3. Edit the metadata (Section 4.3.3.1) to enter the appropriate search keywords and the module title.
4. Edit the module roles (Section 4.3.4.1).
5. Add links (Section 4.3.5.1) to the module.
6. Preview (Section 4.3.6.1) the module in both print and on-line versions.
7. Publish (Section 4.3.6.2) the module.

4.2 Import/Export

4.2.1 Importing and Exporting to Connexions

Our editors allow you to work with your content online, anywhere, anytime, without needing to keep track of files on your local hard drive. While this works in most cases, there are also times when authors or developers would prefer working in an offline environment. Similarly, many authors with existing Word, OpenOffice, or LaTeX documents do not want to have to start over with a new document. The following sections describe the import and export options available to you through our platform, along with links to resources to help get you started.

Contents

- Importing a Module (Section 4.2.1.1: Importing a Module)
- Importing a Collection (Section 4.2.1.2: Importing a Collection)
- Exporting a Module (Section 4.2.1.3: Exporting a Module)
- Exporting a Collection (Section 4.2.1.4: Exporting a Collection)

4.2.1.1 Importing a Module

There are several types of importers (Section 4.2.1.1.1: Types of importers) to help authors quickly upload and convert existing word processor documents into CNXML modules. These importers were designed to accommodate as many common cases as possible, but do require a certain degree of preparation to ensure that the documents are imported as accurately and completely as possible. If you are looking to import content, please be sure to read all of the documentation and template information provided before using the importer (Section 4.2.1.1.2: Using the module importers).

4.2.1.1.1 Types of importers

The following importers are provided for working with existing or external content:

---

2This content is available online at <http://cnx.org/content/m38952/1.1/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 4. AUTHORING CONTENT

4.2.1.1.1 Word/OpenOffice Importer

You can use the Word/OpenOffice importer to import *.doc files and convert them to CNXML modules. You can use either Microsoft Word or an open source editor like OpenOffice Writer that allows you to save in the *.doc format to prepare your document. In either case, be sure to save with file extension *.doc, or as the Microsoft Word 97-2003 compatible version. The following links provide resources to help get you started:

- View the Word Importer documentation (Section 4.2.2)
- Download the Word Importer template (.dot)³
- Download the Word Importer example document (.doc)⁴

Remember that using the Word/OOo importer will overwrite any existing content, so only use this method when working with a new module or when you are sure you don’t mind starting over.

4.2.1.1.2 LaTeX Importer

You can use the LaTeX importer to import .tex files (packed as a .zip file along with embedded media files) and convert them to CNXML modules. The following links provide resources to help get you started:

- View the LaTeX Importer documentation (Section 4.2.3)
- Download the LaTeX Importer template (.tex)⁵

Remember that importing a LaTeX document into a module will overwrite any existing content, so only use this method when working with a new module or when you are sure you don’t mind starting over.

IMPORTANT: Please be sure to read all of the instructions in the help file and template very carefully, particularly the list of supported LaTeX packages. The vast majority of LaTeX importer errors are the result of using an unsupported package.

4.2.1.1.3 Plain CNXML Importer

For authors wishing to develop CNXML files in an external editor, you can upload your files (saved as plain text files with the *.cnxml extension) using the Plain CNXML Importer. You can learn more about the structure of a CNXML document in the CNXML Tutorial⁶.

³http://cnx.org/content/m38952/latest/..../help/authoring/templates/Connexions-Document.dot
⁴http://cnx.org/content/m38952/latest/..../help/authoring/templates/WordExample.doc
⁵http://cnx.org/content/m38952/latest/..../help/authoring/templates/LaTeX_template.tex
⁶CNXML Tutorial <http://cnx.org/content/col10121/latest/>

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
This importer will overwrite the current CNXML module with the contents of the file provided, so only use this method when working with a new module or when you are sure you don’t mind starting over.

Several authors have asked if we provide support for offline CNXML editing. While we do not currently provide such support, we are interested in developing this at some point in the future provided there is sufficient interest for our authors. If you would be interested in contributing to this development, please email techsupport@cnx.org and let us know!

4.2.1.1.4 ZIP Importer

The ZIP Importer is unique in that it is the only importer that does not necessarily overwrite the contents of the module. The contents of the .zip file are uploaded to the Files tab of the module, replacing/updating any contents of the same name that already exist.

- If the ZIP file contains a file called index.cnxml, then the module contents are overwritten just as they are with the Plain CNXML importer.
- If the ZIP file contains files with the same name as previously attached files, then those files are replaced with the new version. This feature makes the ZIP importer a convenient way to update several images, downloads, or other supporting resources at once.
- If the ZIP file contains files which are not already attached to the module, then those files are added just as if they had been added manually through the Files tab. This feature makes the ZIP importer a convenient way to "batch upload" several new supporting resources at once.

NOTE: The module file structure does not support folders or subdirectories. Any folders or subdirectories in the zip file that you import into a module will be ignored.

4.2.1.1.2 Using the module importers

1. If you have not already done so, create (Section 4.1) a new blank module.
2. Use the help documents linked above to ensure you have prepared your document correctly; otherwise you may get an error.
3. The importer dropdown menu is accessible on both the module Content tab and the Files tab when editing the module. Choose the importer type from the dropdown menu and click Import.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4. On the next page, browse to your saved file on your harddrive and click Import.

You can mix and match any of the different importer types to meet your needs; for example, you can upload new files while simultaneously updating existing images, replace the index.cnxml file without affecting other existing resources, etc. This importer can be used as a batch uploader, as an extension of the Plain CNXML importer, or in conjunction with the ZIP exporter (list, p. 55).

4.2.1.2 Importing a Collection

Importing collections is currently not supported; however, we have recently begun laying the groundwork to make this feature available at some point in the future. If you are a developer and would like more information regarding the current support of exported or third-party CollXML documents, please contact techsupport@cnx.org for assistance.

4.2.1.3 Exporting a Module

You can export a module for offline editing, to create a backup of your content, or to use with an external platform that supports the use of CNXML documents. You can export a module from your workgroup by checking it out and using the "Export" menu in the module editor:
There are two options available for exporting modules:

- **Plain CNXML** allows you to download the CNXML\(^9\) file for the module, which contains all of the module text and CNXML markup. The exported file is titled "<moduleid>-plain.cnxml", where <moduleid> is the module’s ID number.

- **Zip File** allows you to download the module’s CNXML file (titled "index.cnxml") along with all attached resource files (such as embedded images, downloadable handouts, etc.).

You can also access the CNXML source code for any published module by appending "source" to the module’s URL (e.g. http://cnx.org/content/m10884/latest/source (Section 1.1)).

### 4.2.1.4 Exporting a Collection

You can use the collection export feature to download a copy of a collection for external use. There are two versions of the collection available:

- The **complete version**, which includes information about the collection structure as well as the complete contents of all component modules.

- The **CollXML-only version**, which includes only the structural information for the collection and does not include component modules.

When viewing the published version of a collection online, you can export both versions of the collection from the metadata page - simply scroll to the bottom of the collection home page and click on the 'Metadata' link, and locate the appropriate link at the bottom of the 'Metadata' section.

---

\(^9\) CNXML Tutorial [http://cnx.org/content/col10121/latest/]

Available for free at Connexions [http://cnx.org/content/col10151/1.27>
Collection authors can also download both versions of the collection export file from their workgroup after publishing a collection, and can download the CollXML-only version from any checked-out collection.

**IMPORTANT:** There is currently no support for importing collections, including those exported through this feature. At present, the collection export feature is provided primarily for developers interested in taking advantage of existing content for use with external platforms supporting CollXML/CNXML documents.

### 4.2.1.4.1 The Collection Export File (complete version)

The complete exported collection is packaged as a ZIP file titled "<collectionid>_<version>_complete.zip", where <collectionid> and <version> are the collection’s ID and version number, respectively. Once expanded, this version of the exported collection contains the following:

- A CollXML\(^{10}\) document (titled "<collectionid>_<version>_collection.xml") describing the collection’s structure.
- For each component module, a folder titled "<moduleid>" (the module’s ID). Each of these folders contains:
  - The CNXML\(^{11}\) document for the module (titled "index.cnxml").
  - Any resource files, such as embedded images or downloadable handouts, that are attached to the module.

### 4.2.1.4.2 The Collection Export File (CollXML-only version)

The structure-only version of the exported collection is available as a downloadable CollXML\(^{12}\) document titled "<id>_<version>_collection.xml", where <id> and <version> are the collection’s ID and version number, respectively. The CollXML file contains information about the collection including references to component modules, the order in which they are presented in the collection, and chapter/section information, along with a copy of the collection metadata.

For more information regarding the contents of the collection export file, please see the CollXML help page\(^{13}\).

---

\(^{10}\)http://cnx.org/content/m38952/latest/xml#collxml

\(^{11}\)CNXML Tutorial <http://cnx.org/content/col10121/latest/>

\(^{12}\)http://cnx.org/content/m38952/latest/../help/authoring/xml#collxml

\(^{13}\)http://cnx.org/content/m38952/latest/../help/authoring/xml#collxml

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.2.2 Import Word/OOo

4.2.2.1 Overview

One of the easiest paths for populating a module is the Word/OOo importer, which converts a *.doc document to CNXML. The importer overwrites any existing CNXML, so it is most useful as an initial import. We suggest using the online Edit-in-Place feature to make any further edits; any re-import of a Word or OpenOffice document will erase any other changes you have made using Edit-in-Place or the Full Source Editor.

Who should use the Word/OOo importer?

You should use the Word/OOo importer if you already have a Word or OpenOffice document saved that you wish to publish to the repository. The importer will correctly convert objects like section headings (designated by innate Word/OOo styles like Heading 1 or Heading 2), paragraphs (separated by two line breaks), tables, lists, images, footnotes, and even mathematical objects.

You can go one step further and apply the special CNXML styles in our template to your document so that you are able to import CNXML-specific objects like examples, problems, notes, citations, quotes, and more. The Word/OOo importer can produce most of the CNXML tags available for authors to use in their modules (see What can I import? (Section 4.2.2.4: What can I import?) below) and several combinations of tags. Any use case not available as a Word/OOo style must be added after import, either using the Edit-in-Place interface (recommended) or full-source editing.

How does it work?

1. First you need to apply the CNXML styles from our template to your document. We have sections below with specific instructions for both Word (Section 4.2.2.2: Preparing documents in Microsoft Word) and OpenOffice (Section 4.2.2.3: Preparing documents in OpenOffice Writer).
2. Next, you need to prepare your Word or OpenOffice document for import. The importer will ignore any user-defined styling, and requires specific markup in order to import other objects correct. We recommend that you read through the What can I import? (Section 4.2.2.4: What can I import?) section to learn what can be imported and make sure that your document has been properly prepared.
4. Create a new blank module in one of your Work Areas.
5. Use the importer on either the Edit or Files tab to import your saved document.

---

14 This content is available online at <http://cnx.org/content/m13056/1.5/>.
15 http://cnx.org/content/m13056/latest/..//eip-help/tags

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.2.2.2 Preparing documents in Microsoft Word

Download a copy of the document template (*.dot)\(^\text{17}\).

You may also want to download a copy of a sample document (*.doc)\(^\text{18}\) that has been prepared to properly exercise all the features of the template. We encourage you to have a look at the document and practice uploading it (see "Uploading Your Document" (Section 4.2.2.5: Uploading your document)) to get a feel for how the Word import process works.

4.2.2.2.1 Creating a new document with the template

To create a new document in Microsoft Word to import into Connexions, download and open either the template\(^\text{19}\) or the sample document\(^\text{20}\) from above. In Microsoft Word go to the File menu and select Save As. In the "Save As" dialog box that appears make sure that "Word Document (*.doc)" or "Word 97-2003 (*.doc)" appears in the Save as Type drop-down box. Give your document a unique name in the File Name box, and click the Save button. You have now created a new Word document with the template styles included in it. Delete the default text from the template and begin typing your document, adding styles as you go as described below.

Alternatively, you can install the template file into Word so that it is available through the New option in the Word File menu. Instructions to install this template are specific to your version of Word, so we recommend that you consult Word’s Help feature if you wish to do so.

4.2.2.2.2 Applying Styles to an Existing Document

If you already have a document that you wish to import, you will need to apply the template styles to the document. To apply the template to an existing document, first open that document in Word. Then, follow the steps below:

1. Go to the Tools menu and click Templates and Add-Ins.
2. Click the Attach button, and use the file browser to navigate to the folder where you saved the template file.
3. Select the "Connexions_Document.dot" icon in that folder and click Open.
4. In the "Templates and Add-Ins" window, make sure the "Automatically Update document styles" box is checked and click Ok. (Figure 1 (Figure 4.7))

---

\(^\text{17}\)http://cnx.org/help/authoring/templates/Connexions-Document.dot
\(^\text{18}\)http://cnx.org/help/authoring/templates/WordExample.doc
\(^\text{19}\)http://cnx.org/help/authoring/templates/Connexions-Document.dot
\(^\text{20}\)http://cnx.org/help/authoring/templates/WordExample.doc

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
As an alternative, you can also create a new blank document with the Word Importer template styles as described in "Creating a New Document with the Template" (Section 4.2.2.2.1: Creating a new document with the template). Once you have done so, you can copy and paste the text from your original document into the new blank document as follows. Go to the Edit menu in the original Word document and select the Select All option. Then go to the Edit menu in the original Word document again and select the Copy option. This copies the text of the original document to your system clipboard. Next, go to the Edit menu in the new blank Word document with the Word Importer template styles and select the Paste option. The text of the original document is now copied into your new document, and you can begin modifying it with Word Importer style information as described in the next section.

4.2.2.2.3 Using styles in Microsoft Word

If you are using Microsoft Word 97, you can find the style drop-down box on your format toolbar at the top of the screen (it usually displays the default style Heading 1 or Normal). In newer versions of Word the style menu is located under the "Home" tab. Click the pop-out icon in the lower right corner of the Styles menu to see a list of all styles.

To apply a style to a specific piece of text, you follow the instructions below:

1. Highlight the word, phrase, or paragraph you want to semantically mark up.
2. Locate the styles menu.
3. Click the name of the semantic style you want to apply to the highlighted text. The text that was highlighted is automatically given the style you selected.
4.2.2.3 Preparing documents in OpenOffice Writer

Download a copy of the document template (*.dot)\(^{21}\).

You may also want to download a copy of a sample document (*.doc)\(^{22}\) that has been prepared to properly exercise all the features of the template. We encourage you to have a look at the document and practice uploading it (see "Uploading Your Document" (Section 4.2.2.5: Uploading your document) below) to get a feel for how the Word import process works.

4.2.2.3.1 Creating a New Document with the Template

To create a new document in OpenOffice Writer to import into Connexions, download and open either the template\(^{23}\) or the sample document\(^{24}\) from above. In OpenOffice Writer go to the File menu and select Save As. In the "Save As" dialog box that appears make sure that "Word Document (*.doc)" or "Word 97-2003 (*.doc)" appears in the Save as Type drop-down box. Give your document a unique name in the File Name box, and click the Save button. You have now created a new document with the template styles included in it. Delete the default text from the template and begin typing your document, adding styles as you go as described below.

4.2.2.3.2 Applying the Template to an Existing Document

To apply the template to an existing document, create a new blank document with the OpenOffice Importer template styles as described in "Creating a New Document with the Template" (Section 4.2.2.3.1: Creating a New Document with the Template). Once you have done so, you can copy and paste the text from your original document into the new blank document as follows. Go to the Edit menu in the original OpenOffice Writer document and select the Select All option. Then go to the Edit menu in the original OpenOffice Writer document again and select the Copy option. This copies the text of the original document to your system clipboard. Next, go to the Edit menu in the new blank OpenOffice Writer document with the OpenOffice Importer template styles and select the Paste option. The text of the original document is now copied into your new document, and you can begin modifying it with OpenOffice Importer style information as described in the next section.

4.2.2.3.3 Using styles in OpenOffice Writer

You can see the styles available to you with the template by going to the Format menu and selecting the Styles and Formatting option. This brings up the "Styles and For-

\(^{21}\)http://cnx.org/help/authoring/templates/Connexions-Document.dot
\(^{22}\)http://cnx.org/help/authoring/templates/WordExample.doc
\(^{23}\)http://cnx.org/help/authoring/templates/Connexions-Document.dot
\(^{24}\)http://cnx.org/help/authoring/templates/WordExample.doc

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Paragraph styles (such as Heading 1, Heading 2, and CNXML Quote) are available by clicking on the leftmost icon at the top of the box. Character styles (such as CNXML Emphasis, CNXML Code, and the rest of the application-specific styles) are available by clicking on the second-leftmost icon at the top of the box.

---

**Figure 4.9:** (a) Paragraph styles. (b) Character styles.

---

To apply a style to a specific piece of text, you follow the instructions below:

1. Select the appropriate style from the "Styles and Formatting" box.
2. Make sure the "Fill Format Mode" tool is selected in the "Styles and Formatting" box (appears as the paint bucket icon in the figure (Figure 4.9) above.
3. Highlight the text to which you wish to apply the style, and it will be applied.

### 4.2.2.4 What can I import?

- Text markup using the template styles (Section 4.2.2.4.1: Text markup using the template styles)
- Other objects (Section 4.2.2.4.2: Other objects)
- Caution to users of MathType 5 (Section 4.2.2.4.3: Caution to users of MathType 5)
- Some common mistakes (Section 4.2.2.4.4: Some common mistakes)

#### 4.2.2.4.1 Text markup using the template styles

We can use the text styles provided by the template to generate Connexions text markup tags when the document is imported.

**Emphasis**

The CNXML Emphasis style is used to give emphasis\(^\text{25}\) to important text. Use this style where you would have used bold or italics to emphasize particular words or phrases. Note that **boldface** and *italicized* text will both import as an emphasis tag with the appropriate effect attribute.

\(^{25}\text{http://cnx.org/eip-help/emphasis}\)

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 4. AUTHORING CONTENT

Foreign
The CNXML Foreign style allows us to designate a term which is in another language than the rest of the text. We can also point its usage at a website of our choice by marking it as a hyperlink in addition to CNXML Foreign. This will produce foreign\(^{26}\) text in a module.

Code
With the CNXML Code style, we can designate inline text as code\(^ {27}\) written in a computer programming language. Use CNXML Code (Block) to designate blocks of code that should be set off from surrounding text.

Cite
The CNXML Cite style allows us to refer to non-electronic sources. If you are using Microsoft Word to prepare your document, you can cite a hyperlinked source by marking it as a hyperlink and then applying CNXML Cite. This is produce a citation\(^ {28}\) in Connexions.

Term
With the CNXML Term style, we can designate key terms\(^ {29}\) in our document. If you are using Microsoft Word to prepare your document, you can also point this term at an example of its usage by marking it as a hyperlink and then applying CNXML Term.

Quote
Using the CNXML Quote (Block) style, we can designate a section of text as a block quote\(^ {30}\) from another source. Use CNXML Quote (Inline) to designate inline quotes that should not be set apart from the surrounding text. If you are using Microsoft Word to prepare your document, you could also point this at an online source by marking it as a hyperlink and then applying CNXML Quote (Inline).

Hyperlinks
Using the Hyperlink style, we can create a hyperlink pointing to a URL. This will become a link\(^ {31}\) element in your Connexions module. We can also point at specific locations within the document by pointing, for example, at any section title created using the H1, H2, \ldots styles, or at any bookmark we have inserted into the document.

Exercise
We can make exercises\(^ {32}\) that only have problems by marking them up using the CNXML Exercise (Problem) style. We can also make exercises that have problems and solutions by following the text marked CNXML Exercise (Problem) with some more text marked CNXML Exercise (Solution).

\(^{26}\)http://cnx.org/eip-help/foreign
\(^{27}\)http://cnx.org/eip-help/code
\(^{28}\)http://cnx.org/eip-help/cite
\(^{29}\)http://cnx.org/eip-help/term
\(^{30}\)http://cnx.org/eip-help/quote
\(^{31}\)http://cnx.org/eip-help/link
\(^{32}\)http://cnx.org/eip-help/exercise

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Theorems
We can have theorems with or without proofs. To make a theorem statement, we can use the CNXML Theorem (Statement) style. To construct a theorem with both a statement and a proof, we use CNXML Theorem (Statement) followed by CNXML Theorem (Proof).

Definitions
We can have terms with definitions. To do so, we mark the term using the CNXML Definition (Term) style, and we follow it with the explanation marked using the CNXML Definition (Meaning) style.

Notes
We can create note boxes using the CNXML Note style.

Examples
We can create example boxes using the CNXML Example style.

4.2.2.4.2 Other objects
We can do more than just mark up text using the Word/OOo importer. We can also add images to our document, create tables, add lists, and even insert mathematical expressions...

Sections
H1 applied to a section title produces a new section. The section lasts until the next H1-marked title. An H2-marked title in between produces a subsection (which lasts until the next H2- or H1-marked title).

Paragraphs
You can also use certain features of the Word or OpenOffice editor to produce other CNXML tags. Pressing Enter gives a new line and produces a new paragraph.

Images
You can insert images into your Word document as you would normally, and they will import into Connexions. It’s best to stick to image types such as with .eps, .jpg, .png, and .gif (or any format with a valid image type); avoid formats such as .wmf and .svg, and avoid Clip Art.

33 http://cnx.org/eip-help/rule
34 http://cnx.org/eip-help/term
35 http://cnx.org/eip-help/definition
36 http://cnx.org/eip-help/note
37 http://cnx.org/eip-help/example
38 http://cnx.org/eip-help/section
39 http://cnx.org/eip-help/para
40 http://cnx.org/eip-help/image
41 http://www.duke.edu/websrv/file-extensions.html

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Use only images that are locally available on your machine. Make sure either that (1) the images are already available under a creative commons (CC) attribution license\footnote{http://creativecommons.org/licenses/by/3.0/}, or (2) you own the image copyrights and are willing to make them available under a CC attribution license.

**Figure**

We can precede images with text marked with the \texttt{CNXML Figure Title} style to give them a title, and follow them with text marked in the \texttt{CNXML Figure Caption} style to give them a caption. This will produce a figure\footnote{http://cnx.org/eip-help/figure} in the module.

**Tables**

Using the built-in Word table editor, we can create tables\footnote{http://cnx.org/eip-help/table} which are easily imported into Connexions.

**Lists**

We can create both bulleted and numbered lists\footnote{http://cnx.org/eip-help/list} using Word or OpenOffice and bring them into Connexions using the importer.

**Mathematics**

We can include mathematics in a line of text. Mathematics set off from other text in its own paragraph will import as an equation\footnote{http://cnx.org/eip-help/equation}.

In OpenOffice, choose Insert $>$ Object from the menu and then select Formula.

In Word we can use the built-in Equation Editor 3.0. To do this, choose Insert $>$ Object from the menu and select Equation Editor 3.0. **WARNING:** It is very important that you insert and equation through the Object menu. Newer versions of Microsoft Word have an option to insert an Equation, located under the Symbols menu. Do not use this as it will not import correctly.

\hspace{\textwidth}

---

**Image not finished**

---

**Figure 4.10:** Insert equations through the Object menu.

\footnote{http://cnx.org/content/col10151/1.27>
If you can’t find the Equation Editor 3.0 option in the list, you must install Equation 3.0 from your Microsoft Office installation CD. Please consult your Microsoft Office help for more information on this procedure.

**WARNING**: If you have MathType 5 installed on your system, math will not import correctly. See below (Section 4.2.2.4.3: Caution to users of MathType 5) for how to remedy this.

---

**Figure 4.11**: Microsoft Office object insert window.

---

**Footnotes**
Footnotes inserted using the native footnote tool will import as Connexions footnotes\(^{47}\).

**Glossary**
Finally, we can designate a glossary\(^{48}\) section for our module. We must first give it a title and mark that with the **CNXML Glossary Section** style. This must be the last section designation used at the end of your word document, and it cannot be followed by H1, H2, etc. styles. Then, for each glossary element we wish to include, we mark the term portion with **CNXML Definition (Term)** and the meaning portion with **CNXML Definition (Meaning)**, as we did before in the body of our document.

**4.2.2.4.3 Caution to users of MathType 5**

MathType 5 is a stand-alone equation editing package written by Design Science, makers of the Microsoft Equation 3.0 editor. When MathType 5 is installed on your system, equations written with the Microsoft equation editor will be saved as MathType 5 equations rather than Equation 3.0 equations. Even merely editing equations from a document with Equation 3.0 equations on a machine with MathType 5 will cause those equations to be saved in MathType 5 format. This is problematic, since the Word importer cannot properly convert MathType 5 equations to Presentation MathML.

We recommend un-installing MathType 5 from your system if it is currently installed. This will cause Word to revert to using Equation 3.0 for equation editing, which will allow the Word importer to properly generate mathematics.

\(^{47}\)http://cnx.org/eip-help/footnote  
\(^{48}\)http://cnx.org/eip-help/glossary

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Should you inadvertently include a MathType 5 equation in your Word document, in the Edit-in-Place interface where the mathematics should appear you will instead see the warning message shown in Figure 5 (Figure 4.12).

![Image not finished]

Figure 4.12: Edit-in-Place warning for MathType mathematics import.

You can go replace each of these warning messages by hand using the following procedure. To begin, double click on the MathType equation in Word to open the MathType editor.

If this is your first time to use this procedure, you must configure MathType to properly generate MathML. Go to the Preferences menu and select the Translators option. In the window that appears, make sure that “Translation to other language (text)” is selected, and in the drop-down box next to “Translator:”, select the “MathML 2.0 (m namespace)” option. Also, make sure that “Include translator name in translation” and “Include MathType data in translation” are NOT selected (see Figure 6 (Figure 4.13) below). Click OK to complete the configuration. This sets the translation preferences permanently, so you will not have to repeat these steps again.

![Image not finished]

Figure 4.13: Setting the translator in MathType to copy MathML to the system clipboard.

Once the Translators preferences are set, you can copy the MathType equation’s MathML equivalent by highlighting the equation in the MathType editor and copying to the clipboard (Edit menu, Copy option). This stores a copy of the MathML on your system’s clipboard.

Return to your browser and find the appropriate placeholder for the equation in the Edit-in-Place interface (appearing as in Figure 5 (Figure 4.12)). Double click to edit the paragraph and delete the text “***SORRY, THIS MEDIA TYPE IS NOT SUPPORTED***”

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
(Figure Figure 7 (Figure 4.14) below). Paste the contents of the clipboard into is place (Figure Figure 8 (Figure 4.15)), and click Save. Your equation should now appear in the Edit-in-Place interface (Figure Figure 9 (Figure 4.16)).

### Image not finished

**Figure 4.14:** Editing the paragraph with the warning text

### Image not finished

**Figure 4.15:** Pasting the MathML copied from the MathType editor into the paragraph where the equation should appear.

### Image not finished

**Figure 4.16:** MathML equation appearing in the edit-in-place interface.

#### 4.2.2.4 Some common mistakes

It is important that you follow the instructions on this page for each object. Do not use other objects or text, including:

1. Do not add line breaks between paragraphs using `cnt1-Enter`. This will not import as a new paragraph.
2. Do not use tabs. This formatting will not import.
3. Do not use lists or tables to invisibly format the document, as these constructs are literally imported as lists and tables into CNXML.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4. Do not continue list numbering across separate lists.
5. Do not insert images whose source files are remotely available via a hyperlink.
6. Do not use Word’s image captioning option; the caption will import as a new paragraph.
7. Do not use fonts such as “Wingdings” to produce symbols: They will be interpreted as text.
8. Do not use the MathType editor inside Word to create equations. This is the default (and only) option if MathType is installed on your system. See the Notes on Including Mathematics (Section 4.2.2.4.3: Caution to users of MathType 5) section for more details.

4.2.2.5 Uploading your document

1. If you have not already done so, create a new blank module.
2. Use the help documents linked above to ensure you have prepared your document correctly; otherwise you may get an error.
3. The importer dropdown menu is accessible on both the module Content tab and the Files tab when editing the module. Choose the importer type from the dropdown menu and click Import.

4. On the next page, browse to your saved file on your harddrive and click Import.

---

Figure 4.17: The module import menu (expanded).

---

Figure 4.18: Browse to file.

---

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.2.3 Import LaTeX

4.2.3.1 How to use the LaTeX Importer

One of the first lessons learned when using the LaTeX importer is to abide by the rules. The importer is a rather faithful utility, but it will not function properly unless the LaTeX file is properly prepared. So, the first step to importing your LaTeX document is preparation.

Connexions has created a template that will aid you in this process. The template is available for download here.

4.2.3.1.1 Using the LaTeX Template

The importer uses Tralics to translate LaTeX to CNXML. Tralics supports a large number of LaTeX packages and commands, but not all of them. Using the Connexions LaTeX Template will increase your chances for a successful import. Follow the steps below to prepare your document.

1. Begin preparing your document by pasting the body of your document between the \begin{document} and \end{document} commands in the template.
2. Un-comment (remove the "%" sign) only the \usepackage{} commands required by your LaTeX document. These are located in the preamble of the Connexions LaTeX template.
3. If you have any "user-defined macros" you may insert these after the \usepackage{} commands.
4. At the end of the document you may supply your .bib files. Make sure these files are at the same directory level as the file that is referencing it.

   NOTE: An issue has been noted with .bib file names containing capital letters. Ensure that your .bib files are labeled in lower-case letters (including the extensions).

5. If your document contains images, make sure these files are at the same directory level as the file that is referencing it.
6. Save your new template-enhanced file. Make sure to rename the files and place it in the folder with your .bib and image files.
7. Run LaTeX (pdflatex) on the new file using only the \usepackage{} commands allowed by the template.
8. Include the new file, all image files and .bib files in a new .zip file with the same name as your template-enhanced .tex file. Their names must match and there should not be any sub-folders.

---

50 This content is available online at <http://cnx.org/content/m17310/1.6/>.
51 http://cnx.org/help/LaTeX_template.tex
52 http://www-sop.inria.fr/apics/tralics/
53 http://cnx.org/help/LaTeX_template.tex

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.2.3.1.2 Importing your LaTeX Document

Once you have successfully completed the steps above, your document should be properly prepared for import in to the Connexions depository. Follow the steps below:

1. In your workspace or workgroup, create a new blank module.
2. Then from the edit tab of that module, select LaTeX from the import drop-down list and click import.

3. Click browse to locate your .zip file.
4. Click import to upload your document.

Figure 4.19

4.2.3.2 Troubleshooting your Import

This section of the module will serve as a consistently updated resource documenting typical importer issues and errors as well as known solutions.

4.2.3.2.1 The Document Imports but there are Errors with the Content

Links to outside of the module are missing: \begin{itemize}
\item \texttt{\textless link\textgreater} tags are CNXML’s analog to LaTeX \texttt{\textbackslash ref} links. When you chop a larger LaTeX document up to make modules, the \texttt{\textbackslash ref} links pointing to objects still inside the new, smaller document will become \begin{itemize}
\item \texttt{\textless links\textgreater}.
\item Those \texttt{\textbackslash ref} links pointing to objects now outside of the modularized document will be preserved as links, but links will not be pointing to anything. All non-internal links will have to be inserted manually using \texttt{\textless link\textgreater}\textsuperscript{54} tags. Also, it is
\end{itemize}
\end{itemize}

\textsuperscript{54}\url{http://cnx.org/eip-help/link}

Available for free at Connexions \url{http://cnx.org/content/col10151/1.27}
important to note that inter-module links can not be established until the modules have been published and assigned a module number.

**Images are too large or small:** The importer displays the image at the size of the originally imported image, but it is possible to manipulate the image size inside Connexions. Click on the image and the image parameters will be displayed. Where it says `<image mime-type="image/png" src="fig15.png" id="uid4_onlineimage" width="500"/> <image mime-type="application/postscript" src="fig15.eps" id="uid4_printimage"/>`, you can change the size of the image by adjusting the number.

**Math text does not retain white space:** The importer converts all math into presentation MathML. When text is written within the `\begin{equation}` and `\end{equation}` tags in the LaTeX file, the text will often be interpreted as an "identifier" `<m:mi></m:mi>`). The result of this error will look like the equation below.

\[
\delta = \min\{\delta_i\}, \text{forall pass and stop bands} \tag{4.1}
\]

To remedy this error, simply remove the MathML that is "writing" the text and replace it with `<m:mtext>your text here</m:mtext>`.

**Two equations are referenced as a single equation:** On occasion, the importer will misinterpret two consecutive equations as a single equation. This occurs quite frequently when there is nothing separating the equations. The solution to this error is to simply copy the MathML for the second equation and pasting it into a new equation box.

**An equation has assimilated everything following it into MathML:** There is a noted bug with Tralics where the converter can not understand where certain equations end. This is especially common with conditional statements such as this:

\[
a(n) = \begin{cases} 
2h(n) & \text{for } 0 \leq n \leq M - 1 \\
h(M) & \text{for } n = M \\
0 & \text{otherwise}
\end{cases} \tag{4.2}
\]

Statements like this may LaTeX on your computer, but they often result in errors when used with Tralics. Unfortunately, this issue is not a simple fix. Follow the steps below to fix this issue.

a. Locate the culprit equation in the .tex file and comment it out.
b. Save the file, and place the newly saved file back in your .zip file containing your .bib and image files.
c. Import the file again.

**NOTE:** Warning, importing the content again will overwrite any work that you have done previously!

d. Make certain that the import was successful, paying special attention to the content following the omitted equation.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
e. Manually input the MathML for the conditional statement.

**NOTE:** Hint, using the `<m:mtable></m:mtable>` command will help you create this type of conditional statement.

### 4.2.3.2.2 The Document Does NOT Import

These errors are much more difficult to remedy. As of right now, the error message received when a file does not import is generic and gives very little information about what actually caused the import to fail, but this section will provide you with a few known issues that might result in failure.

**Known Tralics Don’ts:** Certain practices allowed by LaTeX in general are not supported through Tralics. These must be avoided, as they will break the importer. They include:

- Placing tables within other tables
- Omitting braces for LaTeX commands when they take a single argument. As an example, consider use of the `\sqrt{}` command without braces, as in `\sqrt{\pi}` instead of `\sqrt{\pi}`. The latter usage is the only supported technique.
- Using different capitalizations on the filenames in `\includegraphics{}` statements than are used in the actual files they reference in the .zip folder. Though this may work in development on a Windows platform, it will cause an error in the Linux-based importer.

**Packages Supported by the Tralics importer:** As mentioned previously, Tralics supports several LaTeX packages, but not all of them. For a complete list of supported packages and sub-packages, visit the Tralics website.

**Use only the packages you need:** Occasionally, using packages that are not required by your document can result in errors. Make certain that all of the un-commented `\usepackage{}` commands are actually required by your document. If you locate any that are not necessary comment them out, save your file and attempt to import the file again.

**mbox Issues:** The `\mbox{}` command is known to cause issues, especially when more than of these commands occurs in close succession. For example the following lines of LaTeX resulted in the a module’s failure to import, even though it was possible to successfully build the file on a Windows platform.

\[
b e\nn_{\phi} = 2K_1i \quad \mid \mid \mid \mbox{for } i = 0, 1,\]

---

55http://www-sop.inria.fr/apics/tralics/packages.html

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
From (\ref{cc15}), this gives

\begin{equation}
\omega_{zi} = \text{sn} \left[ 2K_1i/n, k \right], \quad i = 0, 1, \ldots
\end{equation}

This can be reformulated using (\ref{cc18}) so that $n$ and $K_1$ are not needed. For $N$ odd, the zero locations are

\begin{equation}
\omega_{zi} = \text{sn} \left[ 2Ki/N, k \right], \quad i = 0, 1, \ldots
\end{equation}

Fortunately, commenting out this section of code made it possible to successfully import the module. The MathML for these equations was added manually after the import of the rest of the module.

**Empty Sections:** An empty section is a portion of the LaTeX file that contains a \section{your text here} command, but lacks any content following it. This sort of empty section is permitted anywhere inside the document except as the final section if the section is followed by a bibliography file. This will break the importer. There is a simple remedy for this issue. If you do not need the section, simply comment it out and attempt to import again, but if you do need the section header resulting from the command, add text following the marker but before the bibliography. Once you import the file, you can manually delete the text from the paragraph under the section marker, but you can not delete the paragraph. This is prohibited by Connexions. A section must contain something.

**4.2.3.3 Conclusion**

Hopefully this short guide has been helpful. Obviously, this list does not cover all of the possible issues, but this module will serve as a living document continuing to be updated and refined as new issues and solutions become known. If you come across an error that you think is a Tralics based error, please let Connexions know using the bug submission form\textsuperscript{56}.

\textsuperscript{56}http://cnx.org/bug_submit_form
4.3 Edit Module

4.3.1 Content

4.3.1.1 Module Content

The "Edit" tab for the module displays the contents of the "index.cnxml" file. This file contains the text that appears in the module, plus the CNXML instructions to include any image, sound, or other media files that appear in the module. The default content display and edit is with the Edit-In-Place editor. Alternatively, you can display and edit the content with the Full Source editor by clicking Switch to Editing Full Source. From this tab you can perform the following actions on the file:

- Import content into the index.cnxml file (Section 4.3.1.1.1: Import Content into the "index.cnxml" File)
- Export the index.cnxml file to an external XML editor (Section 4.3.1.1.2: Export the "index.cnxml" File to an External XML Editor)
- Edit the index.cnxml file with the Edit-In-Place editor (Section 4.3.1.1.3: Edit the index.cnxml file with the Edit-In-Place Editor)
- Edit the index.cnxml file with the Full Source editor (Section 4.3.1.1.4: Edit the "index.cnxml" File with the Full Source Editor)

Figure 4.20: The "Edit" tab displaying contents of the "index.cnxml" file in the Edit-In-Place editor mode.

4.3.1.1.1 Import Content into the "index.cnxml" File

To import content into the "index.cnxml" file, use the following steps:

1. Select the format of file you want to import from the drop-down list next to the Import button. The valid formats are: Microsoft Word, OpenOffice Writer, XMLSpy/Authentic, and Plain XML.

---

57This content is available online at <http://cnx.org/content/m38193/1.1/>.
58http://cnx.rice.edu/help/UsingMSWord
59http://cnx.org/help/UsingOOWriter
60“Editing CNXML with Altova’s Authentic” <http://cnx.org/content/m11160/latest/>
2. Click Import. A screen displays with a text box for the pathname of the file you want to import.
3. Type the pathname of the file in the text box or use Browse to display the name of the pathname in the text box.
4. Click Import. The content of the file you specified is copied into the "index.cnxml" file in your module.

WARNING: When you click Import in the last step, the contents of the "index.cnxml" file are overwritten by the contents of the imported file.

4.3.1.1.2 Export the "index.cnxml" File to an External XML Editor

You can export the "index.cnxml" file so that you can edit it with a software package that is specifically designed to edit XML files. The "index.cnxml" file must be altered to allow the XML editor to read it. And once you have completed your edits, the file must be altered again to remove the changes for editor readability. The Connexions Export and Import functions automatically add and remove the changes for editor readability. Currently, the only XML editor format supported by Connexions is the format used by the Altova® xmlspy® editor or the Altova authentic® editor.

To export the "index.cnxml" file to be edited with an XML editor, use the following steps:

1. Select the format of the file you want to export from the drop-down list next to the Export button. The valid formats are: XMLSpy/Authentic, and Plain XML.
2. Click Export. A dialog box displays asking if you want to open the file or save it to a disk.
3. Specify that you want to save the file to a disk.
4. Specify the pathname of where you want to save the file.
5. Launch your XML editor and edit the file.

To import the "index.cnxml" file back into Connexions, use the instructions for Import Content from Outside Connexions (Section 4.3.1.1.1: Import Content into the "index.cnxml" File) mentioned earlier in this module.

4.3.1.1.3 Edit the index.cnxml file with the Edit-In-Place Editor

The Edit-In-Place editor is the default editor on the "Edit" tab. You can insert new text and media objects into a module or modify the existing contents of a module with this editor. You do not have to be familiar with the CMXML tags to use Edit-In-Place. It inserts the appropriate CNXML tags for you when you insert a new item in the file. In addition, it has a help feature that describes the CNXML tags for the items you insert or edit with it.

61 http://www.altova.com
62 "Editing CNXML with Altova’s Authentic" <http://cnx.org/content/m11160/latest/>
4.3.1.1.3.1 Inserting Content in an Empty Module

To insert text and media objects into a new module that contains no content, use the following steps:

1. Click in the white box that says "Click To Edit Paragraph" in the Edit-In-Place editor.

![Image not finished](image)

*Figure 4.21:* The "Click To Edit Paragraph" text box in the Edit-In-Place editor.

The white box is replaced by a blue editing box.

2. Type the text you want to add in the blue editing box.

![Image not finished](image)

*Figure 4.22:* The editing box in the Edit-In-Place editor.

3. Click Save to save the text you typed. The blue editing box is replaced by a white box that displays your entry.

4.3.1.1.3.2 Editing Existing Content with Edit-In-Place

To edit the content of a module, use the following steps:

1. Display the module you want to edit with Edit-In-Place.
2. Scroll down to display the white text box that contains the item you want to edit.
3. Click in the text box. The white box is replaced by a blue editing box that displays the text of the element.

![Image not finished](image)

*Figure 4.23:* The blue text box in the Edit-In-Place editor.

4. Make the necessary changes to the item.
5. Click Save to save the entry or click Cancel to clear the entry in the blue editing box.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
6. Repeat steps 2 through 5 for any item you want to edit.

NOTE: In the white text boxes mathematical equations are displayed as they appear in the module. In the blue editing box mathematical equations are displayed in the MathML markup language. Edit equations by making changes to the MathML.

4.3.1.1.3.3 Adding Content Items with Edit-In-Place

You can add new content items to a module with Edit-In-Place. The types of items you can add are: sections, paragraphs, examples, exercises, notes, lists, code blocks, equations, and tables. Edit-In-Place places the opening and closing CNXML tags around the item and it generates a unique item ID for each item you add. To add new items, use the following steps:

1. Display the module you want to edit with Edit-In-Place.
2. Scroll down to display the location in the module in which you want to insert the content item. You can insert the new item above or below and existing item by using the Insert... link above or below the existing item.
3. Select and click the type of item you want to enter from the drop-down list next to the Insert... link. These items are described in the help text, which can be accessed by selecting "Other Elements" from the drop-down list.
4. An empty blue editing box displays for the item you selected. You can display help text for the item you selected by clicking Help editing xxx in the upper right corner of the blue box, where xxx is the CNXML tag for the item you selected.
5. Type the text that is appropriate for the type of item in the blue editing box.
6. Click Save to save the entry or click Cancel to clear the entry in the blue editing box.
7. Repeat steps 2 through 6 for each new content item you want to add to the module.

4.3.1.1.4 Edit the "index.cnxm" File with the Full Source Editor

You can edit the content and CNXML tags in the "index.cnxm" file within Connexions with the Full Source editor. This method of editing is recommended for quick changes only. This is a simple text editor and it does not have the advanced functions (for example, spell checking and text searching) that are available with commercially available text editors. To edit the file with the Full Source editor, use the following steps:

1. Click Switch to Editing Full Source on the "Edit" tab. The contents of the "index.cnxm" file are displayed in the Full Source editor mode.
CHAPTER 4. AUTHORING CONTENT

Figure 4.24: The "Edit" tab displaying the contents of the "index.cnxml" file in the Full Source editor mode.

2. Scroll down to display the location in the module in which you want to edit.
3. Click in the text window and begin editing.
4. Click Save to save your changes.

As you edit the "index.cnxml" file, click Save at any point to save your changes. The page reloads and validates your CNXML markups every time you save your changes. If there are no CNXML tag errors, your markup is valid and "Saved" appears in an information box at the top of the tab. If there are errors in your CNXML markup, they are listed above the file content.

**NOTE:** Saving and publishing are two different operations. When you save your editing changes in the Full Source editor, your changes are saved in your work area; they are not entered into the Connexions repository. Your module must be published to be entered into the repository and be accessible by visitors to Connexions. For information about publishing, see the topic Publishing the Module later in this module.

4.3.1.1.4.1 Validating Your Work in the Full Source Editor

Any "index.cnxml" file that you publish in the Connexions repository must be a valid CNXML document. This means that the file must be well formed and contain no CNXML errors. If it does contain errors, a description of each error (Figure 4.25) appears when you click Save. Your changes are not saved until the errors are corrected.

**WARNING:** You cannot publish the module in Connexions until all the CNXML errors are corrected and the "index.cnxml" file is validated.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.1.2 Adding Multimedia

Contents

- Introduction (Section 4.3.1.2.1: Introduction)
- Making Multimedia Accessible for All Users (Section 4.3.1.2.2: Making Multimedia Accessible for All Users)
- Embedding Media Elements in Your Module (Section 4.3.1.2.3: Embedding Media Elements in Your Module)
  - The <media> Element (Section 4.3.1.2.3.1: The <media> Element)
  - The <figure> Element (Section 4.3.1.2.3.2: The <figure> Element)
  - The <subfigure> Element (Section 4.3.1.2.3.3: The <subfigure> Element)
  - Including Alternate Media for Print (Section 4.3.1.2.3.4: Including Alternate Media for Print)
- Media Subtypes (Section 4.3.1.2.4: Media Subtypes)
  - Images (Section 4.3.1.2.4.1: Images)
  - Video (Section 4.3.1.2.4.2: Video)
  - Flash objects (Section 4.3.1.2.4.3: Flash Objects)
  - Audio (Section 4.3.1.2.4.4: Audio Files)
  - Java Applets (Section 4.3.1.2.4.5: Java Applets)
  - LabVIEW Demonstrations (Section 4.3.1.2.4.6: LabVIEW Demonstrations)
  - Parameters (Section 4.3.1.2.4.7: Parameters)
- Using Third-Party Services (Section 4.3.1.2.5: Using Third-Party Services)
  - Embedding YouTube Videos (Section 4.3.1.2.5.1: Embedding YouTube Videos)
  - Embedding SlideShare Presentations (Section 4.3.1.2.5.2: Embedding SlideShare Presentations)

4.3.1.2.1 Introduction

Connexions modules can contain many different types of embedded multimedia files. This document explains how to use these elements to create media-rich modules. Though you are welcome to skip ahead at any time, we strongly recommend that you read the sections on Accessibility, Uploading Media Files, and Embedding Media Elements in Your Module before moving on to a specific medium as these sections cover several important aspects common to all media types.

63This content is available online at <http://cnx.org/content/m12660/1.20/>.
NOTE: This module contains a number of code examples designed to illustrate specific aspects of the CNXML language. As a result, these code snippets will not always take advantage of all of the available options and attributes for every element. Feel free to experiment with different combinations and structures as you build your modules to learn how these options can enhance your module’s multimedia content.

Connexions supports the use of any media format that has a valid MIME type (see this page\(^64\) for more information). In order to take advantage of the media, however, end-users must also have any necessary software and/or browser plugins installed on their browsers. For this reason, it is advisable to stick with common file formats when authoring in Connexions. This module provides a number of CNXML code snippets that cover several typical use cases, though this list is not exhaustive.

TIP: If you need assistance adding multimedia to your modules, have questions about an object not listed in this module, or have any other authoring issues that you need help with, please contact Connexions technical support at techsupport@cnx.org\(^65\).

4.3.1.2.2 Making Multimedia Accessible for All Users

It is important for authors to remember that not all users are able to experience media the same way. Visually impaired users will not be able to view graphical content, while those hard of hearing will not be able to experience audio content as intended. As an author, should always strive to make their content accessible to as many audiences as possible; in addition to allowing more users to enjoy and benefit from your work, you will also make it easier for the many organizations required by policy or law to meet accessibility standards to use and adopt your content.

Alt Text

Alt text allows authors to provide a short description of the visual content of a media element. These descriptions can be read by screen reading software packages used to assist visually impaired users. Keep your alt text short but descriptive; imagine somebody trying to describe an image you cannot see and think about the information you would find most important. A few tips:

- Don’t be redundant. The user is already aware that this is an image, so don’t start your description with "This image is . . . " or "A picture of . . . "
- Keep it short and sweet. Describe only the relevant aspects of the image; for example, it might be important to describe the subject of a photograph as having freckles, but it is probably not necessary to count how many freckles she has on each cheek.

\(^{64}\)http://www.ietf.org/rfc/rfc1341.txt?number=1341

\(^{65}\)techsupport@cnx.org

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Use correct grammar and spelling. For the majority of users requiring alt text descriptions, this content will be read out loud; these descriptions should adhere to the same quality and style standards used elsewhere in your work.

IMPORTANT: Alt text is required for all `<media>` elements.

longdesc
A longdesc (or "long description") file is used to describe images that are too complex to summarize in alt text. Rather than including the description in the CNXML markup, the full, detailed description is contained in a separate file that is linked from the media element. This allows you to describe the image in as much detail as necessary without needing to keep the description short. A longdesc file:

- Can be any file type, though a basic text-only HTML file is most appropriate.
- Can be loaded into a Connexions module like any other image or resource file.
- Is not a replacement for alt text, but instead is a chance for the author to elaborate on alt text.
- Is only necessary when the image is so complicated that an alt text description is not sufficient to describe it.

The following example shows how to link to a longdesc file titled `earnings_report_desc.html`:

```
<media id="longdesc_example"
   alt="An earnings graph showing steady growth of 6% over the past two quarters."
   longdesc="earnings_report_desc.html">
   <image mime-type="image/png" src="earnings_report.png" />
</media>
```

In this example, `earnings_report_desc.html` would most likely be a simple text file describing the details of the chart, including the data points, overall trends, and any other relevant information. While the alt text describes "steady growth of 6% over the past two quarters," the longdesc file can talk about specific features of the chart, such as a dip that coincided with a recent merger or a spike that represented a major product release. A screen reader user would have the option of following this link in order to get a detailed understanding of the data that would otherwise be conveyed and understood by sighted users through visual information.

TIP: Remember, longdesc files are not common, and are usually only used when the media element contains important visual information that is too complex and/or detailed for alt text.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Captioned Video and Audio Transcripts
Captioning is another important consideration when dealing with video content. Users who are hearing impaired may not be able to understand all of the information being conveyed in a video, limiting their access to the material being described. This can be addressed by submitting videos that are captioned, allowing those users to read along with the spoken materials. If you are not able to caption your videos, another option is to include audio transcripts of the materials as a module resource that can be easily accessed using a link. While there are no specific tools or standards for providing this type of content, the important thing to remember is to make it as easy as possible for as many people as possible to access the information you are trying to communicate.

NOTE: You can read more about how Connexions addresses accessibility concerns in m17212 - Accessibility Features of Connexions.66

4.3.1.2.3 Embedding Media Elements in Your Module

4.3.1.2.3.1 The <media> Element
Once you’ve attached your multimedia files to your module, the next step is to embed the media files in your content. This can be done by using the <media> element.

The <media> element is the primary building block for including multimedia files in your module. In its simplest form, a <media> element contains three pieces of information:

- A unique id
- Alt text
- A media subtype element, such as <image>

The code snippet below shows a simple example of <media> element:

```html
<media id="media_example" alt="a dog on a bed">
  <image mime-type="image/jpeg" src="image1.jpg"/>
</media>
```

This code results in the following display:

---

66"Accessibility Features of Connexions" <http://cnx.org/content/m17212/latest/>
Note that the resulting image displays inline; when included as the child of another element, such as <para>, the media element will display in line with the surrounding text by default:

```html
<para id="myparagraph">
  <media id="media_example2" alt="a dog on a bed">
    <image mime-type="image/jpeg" src="image1.jpg"/>
  </media>

  This is my dog. Isn’t he cute?
</para>
```

This code results in the following display:

This is my dog. Isn’t he cute?

Each <media> must contain a media subtype element from the following list:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.1.2.3.2 The <figure> Element

The <figure> element allows you to set a media element apart from the surrounding text and highlight it as a labeled figure within the module. In its simplest form, a <figure> contains two pieces of information:

- A unique ID
- A <media> element

The code snippet below shows an example of a <figure> element:

```html
<figure id="figure_example">
    <media id="dog_on_bed" alt="a dog on a bed">
        <image mime-type="image/jpeg" src="image1.jpg"/>
    </media>
</figure>
```

This example results in the following:
You’ll notice that the figure is labeled as "Figure 4.26" - this information was not provided in the CNXML code, but is instead supplied automatically by Connexions based on the figure’s place within the document. As you add, move, and delete figures in your module, you do not have to worry about keeping track of the figure numbering - this is done for you when the page is displayed to the reader. Also, unlike the <media> example shown previously, the image in this example is displayed in block mode rather than inline mode.

TIP: When referring to figures in your Connexions modules, you can take advantage of the figure’s unique ID to create a dynamic reference that is automatically labeled for you. To do so, simply create an empty <link> that points to the <figure>; in this case,

```
<link target-id="figure_example" />
```

results in the following link: Figure 4.26.

Connexions will automatically label the text of the link to match the figure’s label. Dynamic references can also be created for other "numbered" elements, such as <equation>, <example>, <exercise>, etc.

The <figure> also provides additional options that allow you to further highlight and describe an enclosed <media> element. The following example illustrates several additional elements:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
• `<title>`, which adds a title to the figure.
• `<caption>`, which adds a caption allowing the author to provide a description or context for the figure.
• `<label>`, which overrides the default ‘Figure’ label; this is especially useful for modules not written in English.

```xml
<figure id="figure_example_extended">
  <label>Figura</label>
  <title>Mi Perro Benny</title>
  <media id="dogpic" alt="Perro sentado en la cama">
    <image mime-type="image/jpeg" src="image1.jpg" />
  </media>
  <caption>Este es mi perro Benny haciendo lo que hace mejor.</caption>
</figure>
```

This example results in the following display:

```
Figura 4.27: Este es mi perro Benny haciendo lo que hace mejor.
```

**TIP:** Remember, the `<figure>` element is not necessary to display a `<media>` element. Use `<figure>` when you want to set your multimedia content apart from the surrounding text, add a title or caption, or create a numbered figure label for your media. If you just want your media to display inline, only use the `<media>` element.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.1.2.3.3 The <subfigure> Element

You can place two or more images within the same figure using the subfigure tag. This is useful when you have two related images that you want to display side-by-side or one above the other. The following CNXML snippet can be used to display two subfigures side-by-side:

```xml
<figure id="figure-2" orient="horizontal">
  <subfigure id="subfigure-1">
    <title>Blue on Blue</title>
    <media id="sub_example" alt="An envelope with a blue page">
      <image mime-type="image/png" src="Xenvelope-blue-on-blue.png"/>
      <caption>Subfigure 1.</caption>
    </media>
  </subfigure>
  <subfigure id="subfigure-2">
    <title>Orange on White</title>
    <media id="sub_example2" alt="An envelope with a white page">
      <image mime-type="image/png" src="Xenvelope.png"/>
      <caption>Subfigure 2.</caption>
    </media>
  </subfigure>
  <caption>Two images displayed horizontally in one figure.</caption>
</figure>
```

The resulting figure is displayed below:

![Two images displayed horizontally in one figure. (a) Subfigure 1. (b) Sub-figure 2.](http://cnx.org/content/col10151/1.27)

The optional orient attribute determines how the subfigures are arranged respective to one another. This attribute can have a value of either horizontal (default) or vertical. Modifying the previous example so that orient="vertical" results in the following:
4.3.1.2.3.4 Including Alternate Media for Print

Connexions allows you to insert an alternate media file in your module that can be formatted and sized specifically for the print version of your content. For example, you may want to provide a color image for online use and a black-and-white image for print. You can also use this feature to provide alternate media types, such as using a Java applet for online use and an image for print.

To create an alternate image for print, simply include two image elements (or, say, a video element and an image element) inside the media tag. Use the for attribute to specify whether this element should be used for the online version, the PDF version, or used as the default selection.

The following code gives one example for including an alternate print image. In this case, online users will see image.png, while image.eps will be used for the print version of the module.

```xml
<figure id='printimage'>
  <media id="print_image_example">
```

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Possible values for the for attribute include:

**default:** (Default) This element is used for both PDF and online versions unless another, version-specific element is present.

**online:** This element is used only for the online version of the content.

**pdf:** This element is used only for the pdf version of the content.

Note that it is not necessary to specify a media element for both online and print versions. For example, given the code snippet above, the first image ("image.png") is the default and will be automatically used for the online version of the content, while the second image ("image.eps") is explicitly assigned to the PDF version of the content. If no default had been specified, then no image would have appeared in the online version of the content.

**TIP:** You can use any two media objects in this way to create alternative media for the online and PDF versions of your content. When two media subtype objects are included inside a media element, use the for attribute to specify when that object should be displayed.

### 4.3.1.2.4 Media Subtypes

Within each `<media>` element, at least one media subtype is required. The following sections describe how to create the various possible subtype elements.

#### 4.3.1.2.4.1 Images

Some of the image file types and their corresponding MIME types used in Connexions modules are:

- **eps** - mime-type="application/postscript"
- **png** - mime-type="image/png"
- **jpg** - mime-type="image/jpeg"
- **gif** - mime-type="image/gif"

This list is not all inclusive. Any image with a valid MIME type\(^{67}\) can be used in a module.

---

\(^{67}\)[http://www.duke.edu/websrv/file-extensions.html]
4.3.1.2.4.1.1 Embedding Images

To embed images in your module, insert CNXML entries similar to the following example into the "index.cnxml" file for your module:

```xml
<figure id="figure-01">
  <title>Example Figure</title>
  <media id="figureexample" alt="an envelope with a blue page">
    <image mime-type="image/png" src="Xenweope-blue-on-blue.png"/>
  </media>
  <caption>A graphic image displayed as a figure.</caption>
</figure>
```

The src attribute in the image tag gives the location or source of the image that you want to appear in the module. After loading the image file into the 'Files' tab on your module, simply provide the full file name (e.g. "envelope-taller.png") to specify which image should be used.

The actual display (Figure 4.30: Example Figure) for the previous CNXML example is:

![Example Figure](image.png)

**Figure 4.30:** A graphic image displayed as a figure.

4.3.1.2.4.1.2 Adjusting the Size of the Image

You can use optional size attributes to control the size of the image displayed in both online and print versions of the content. There are a few things to keep in mind when specifying image sizes in CNXML:

- Images that are displayed larger than their original size will look fuzzy or grainy.
- Resizing an image online relies on the web browser to determine how to redraw the image. This can sometimes lead to unpredictable and undesirable results that vary from browser to browser. Whenever possible, resize the image to the desired dimensions before loading the file into Connexions to avoid this problem.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
• It is not necessary to specify both the height and width of an image. If you supply only one of these, the other value will be calculated to preserve the same height-width ratio as the original image. You only need to supply both when you wish to change the shape of the image.

• You can choose to resize an image for online viewing, printing, or both by supplying the corresponding attributes as described below. Sizes specified for one version have no effect on the other. If you do not supply sizing information, the image will be displayed in its original size by default.

4.3.1.2.4.1.2.1 Adjusting the Size of an Online Image

You can adjust the size of the image in the on-line version of your document by using the height and width attributes.

The following code will display the image in its original, default size:

```xml
<figure id="element-439">
<media id="tajmahal" alt="The Taj Mahal">
  <image src="TajMahal_medium.jpg" mime-type="image/jpeg"/>
</media>
<caption>
  The photograph of the Taj Mahal in this figure and the following figures was taken by <link url="http://www.flickr.com/photos/babasteve/">Steve Evans</link>. It is licensed for public use under the Creative Commons Attribution License.
</caption>
</figure>
```

This code results in the following image:
Figure 4.31: The photograph of the Taj Mahal in this figure and the following figures was taken by Steve Evans. It is licensed for public use under the Creative Commons Attribution License.

The following CNXML code shows how to display the same image with a specific size (in this case, 250 pixels wide by 250 pixels tall):

```cnxml
<figure id="figure_size_adjusted">
  <media id="TajMahal" alt="The Taj Mahal">
    <image mime-type="image/jpeg"
      src="TajMahal_medium.jpg"
      height="250"
      width="250" />
  </media>
</figure>
```

The resulting figure is displayed below:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
When specifying width and height, enter the value in pixels. For example, the code above will produce a print image that is 250 pixels tall and 250 pixels inches wide.

**TIP:** It is not necessary to specify both the height and width of an image. If you supply only one of these, the other value will be calculated to preserve the same height-width ratio as the original image. You only need to supply both when you wish to change the shape of the image.

**TIP:** The height and width attributes do not affect the size of the image in the print (PDF) version of the module. To change the printed size of the image, use the print-width attribute (see below). You can use any combination of print and online sizing attributes for an image.

### 4.3.1.2.4.1.2.2 Adjusting the Size of a Printed Image

You can adjust the size of the image in the print (PDF) version of your document by using the print-width attribute. The height of the image will be adjusted proportionately to the width set in print-width. There is no print-height attribute.
The following CNXML code shows how to display the Taj Majal image above with a specific printed size (in this case, 4.5 inches wide):

```xml
<figure id="figure_size_adjusted">
  <media id="TajMahal" alt="The Taj Mahal">
    <image mime-type="image/jpeg"
      src="TajMahal_medium.jpg"
      print-width="4.5in" />
  </media>
</figure>
```

When specifying `print-width` include the unit of measurement in the value. This can be in (inches), cm (centimeters), pt (points) or any other value supported in LaTeX\(^{69}\). The code above will produce a print image that is 4.5 inches wide and a height proportional to its original dimensions.

TIP: The `print-width` attribute does not affect the size of the image when viewing the version of the module. To change the online size of the image, use the `height` and `width` attributes. You can use any combination of print and online sizing attributes for an image.

### 4.3.1.2.4.1.2.3 Creating a Linked Thumbnail Image

A thumbnail is a small version of an image that, when clicked, displays the larger, full-size version of that image. You can create a thumbnail in your modules by completing the following steps:

1. Using image-editing software, create a thumbnail-sized version of your image file. Make sure to change the name of the thumbnail image so that you know which is which (e.g. "myimage_thumb.png").
2. Add the thumbnail image to your module using the files tab as you would with any image file.
3. Create the `media` and `image` elements as you would normally do for a full-sized image.
4. Use the optional `thumbnail` attribute to the `image` element as shown in the example below.

Here is an example of the CNXML code required to include a thumbnail image that links to a full size version:

```xml
<figure id='thumbnail'>
  <media id="thumbnailmedia" alt="The Taj Mahal (thumbnail version)">
    <image mime-type="image/jpeg"
      src="TajMahal_thumb.jpg"
      print-width="1.5in" />
  </media>
</figure>
```

\(^{69}\)http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/teTeX/latex/latex2e-html/ltx-86.html

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
This code results in the following linked image:

![Taj Mahal](TajMahal_medium.jpg)

**Figure 4.33**

### 4.3.1.2.4.2 Video

#### 4.3.1.2.4.2.1 Embedding Videos Attached to the Module

Movies can be added to your module using the same basic structure as with other media types.

Available for free at Connexions [http://cnx.org/content/col10151/1.27]
<figure id='moviefig2'>
<media id="movie" alt="Building on the Past">
<video mime-type="video/mpeg"
    src="Building_on_the_Past.mpg"
    width="350"
    height="300"
    autoplay="false" />
</media>
<caption>
The Creative Commons movie: "Building on the Past".
Click the Play button to start the movie.
</caption>
</figure>

The movie is displayed below:

This media object is a video file. Please view or download it at
<Building_on_the_Past.mpg>

**Figure 4.34:** The Creative Commons movie: "Building on the Past". Click the Play button to start the movie.

### 4.3.1.2.4.3 Flash Objects

You can insert Flash objects in your module. Here is an example of the CNXML code required to include a Flash object:

<figure id='flashfig'>
<media id="new_flash" alt="flash animation of the connexions logo">
<flash mime-type="application/x-shockwave-flash"
    src="flash.swf"
    height="250"
    width="250" />
</flash>
</media>
<caption>"Welcome to Connexions" example Flash object.</caption>
</figure>

The Flash object (Figure 4.35) appears in your module the same way in which a movie or an image appears:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
This media object is a Flash object. Please view or download it at <flash2.swf>

Figure 4.35: "Welcome to Connexions" example Flash object.

4.3.1.2.4.4 Audio Files

Audio files such as mp3, real audio, and wav files can be inserted into your module quickly and easily. To include audio files in your document, upload the corresponding audio file and include code similar to the following in your "index.cnxml" document:

```xml
<figure id='musicscale'>
  <media id="music_example" alt="chromatic scale slurred">
    <audio mime-type="audio/x-wav" src="chromatic_slurred.wav"/>
  </media>
  <caption>A chromatic scale performed on clarinet by Michael Lawrence.</caption>
</figure>
```

The actual display (Figure 4.36) for the previous CNXML example is:

This media object is an audio file. Please view or download it at <chromatic_slurred.wav>

Figure 4.36: A chromatic scale performed on clarinet by Michael Lawrence.

4.3.1.2.4.5 Java Applets

Java applets are another media type you can use in your Connexions modules:

```xml
<figure id="javafig">
  <media id="java_example" alt="a complex sinusoid">
    <java-applet mime-type="application/x-java-applet"
                code="PhasorDemo.class"
                width="430"
                height="500"/>
  </media>
</figure>
```

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.1.2.4.6 LabVIEW Demonstrations

For information on including LabVIEW demonstrations in your module, please see Creating LabVIEW demonstrations for Connexions\(^ {71}\).

4.3.1.2.4.7 Parameters

Some multimedia objects need options or parameters to display properly. You can pass this information to the multimedia objects with the \texttt{param} tag. The \texttt{param} tag allows you to specify the run-time settings for an object inserted into XHTML documents.

Here is an example of a \texttt{param} tag for an audio (wav) object:

\begin{verbatim}
<figure id="audio2">
  <media id="music_example2" alt="Slurred chromatic scale">
    <audio mime-type="audio/x-wav" src="chromatic_slurred.wav">
      <param name="title" value="Chromatic Scale"/>
    </audio>
  </media>
</figure>
\end{verbatim}

The actual display (Figure 4.38) for the previous example is

\begin{verbatim}[http://java.com]
\end{verbatim}

\begin{verbatim}[http://cnx.org/content/m11601/latest/]
\end{verbatim}

\begin{verbatim}[http://cnx.org/content/col10151/1.27]
\end{verbatim}

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.1.2.5 Using Third-Party Services

It is possible to embed videos and other multimedia files from sites such as YouTube\(^{72}\) or SlideShare\(^{73}\) into your module, just as you might embed them on your own personal web site. Please keep in mind, however, that unlike media files attached to Connexions modules, there is no guarantee that these resources will be always be available to your readers. Also, content hosted on third-party websites is not necessarily released under the same open licenses as Connexions content, meaning that other authors won’t have the same rights to edit and modify content as they would with an attached media file. For these reasons we strongly encourage you to consider providing your own attached media files whenever possible.

Most resources hosted by a third-party service can be embedded using the same process as an attached video or flash file as described earlier, with the \texttt{src} attribute pointing to the URL of the video rather than a local file. Some of these external resources, however, require a little extra work by the author.

4.3.1.2.5.1 Embedding YouTube Videos

If you choose to include YouTube videos in your module, you will need to first copy the information provided in the \textbf{Embed} HTML snippet provided by YouTube for the video you wish to use. As an example, consider this video\(^{74}\) of Neil Armstrong walking on the moon. The Embed snippet for this video is as follows:

\begin{verbatim}
<object width="425" height="344">
  <param name="movie" value="http://www.youtube.com/v/RMINSD7MmT4&hl=en&fs=1&rel=0"/>
  <param name="allowFullScreen" value="true"/>
  <param name="allowScriptAccess" value="always"/>
  <embed src="http://www.youtube.com/v/RMINSD7MmT4&hl=en&fs=1&rel=0"
type="application/x-shockwave-flash" allowscriptaccess="always"
allowfullscreen="true" width="425" height="344">
\end{verbatim}

\(^{72}\)http://www.youtube.com/
\(^{73}\)http://www.slideshare.net
\(^{74}\)http://www.youtube.com/watch?v=RMINSD7MmT4

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Since this code is designed for use on websites and not the CNXML language, the majority of this snippet is not necessary. The part that is important is the embed tag:

```xml
<embed src="http://www.youtube.com/v/RMINSD7MmT4&hl=en&fs=1&rel=0" type="application/x-shockwave-flash" allowscriptaccess="always" allowfullscreen="true" width="425" height="344">
```

The CNXML example below illustrates how to embed a YouTube video in your module. Notice that it is implemented like a standalone video, with a few key features:

- The media subtype is `<iframe>`, not `<video>` - this is because the YouTube player uses iframe for embeds. In practice, the `<video>` element would work just fine to embed a simple YouTube player, but some features (such as being able to view the movie in full screen mode) would not be available.
- The `src` attribute is set to the embed URL for the video, not the URL for the video on the YouTube website. The embed URL is the one found in the code snippet as shown above and is of the form `http://www.youtube.com/embed/[VideoID]` (where `[VideoID]` is the ID of the video you wish to include in your module).
- After you copy the embed URL, you need to replace the `&` characters with `&amp;`. The `&` character is a special character in CNXML and must be escaped in this manner, otherwise you will get an error when you try to save your module. In this example,

```xml
src="http://www.youtube.com/embed/RMINSD7MmT4&rel=0"
```

The final CNXML code for embedding this movie is as follows:

```xml
<figure id="moonlanding-youtube">
  <title>One Small Step for (a) Man</title>
  <media id="yt-media" display="block" alt="Video of the Neil Armstrong on the Moon">
    <iframe width="425" height="344"
      src="http://www.youtube.com/v/RMINSD7MmT4&rel=0"/>
  </media>
</figure>
```

And results in the following:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
TIP: Though the steps described above may look intimidating, the actual process isn’t actually that hard. To embed your YouTube video, just perform the following steps:

Step 1. Copy the example code above into your module.
Step 2. Replace the URL with the one provided in the <embed> element in the HTML snippet provided by YouTube.
Step 3. Escape the ampersand characters in the URL by replacing & with &amp;.
Step 4. Adjust the height and width parameters to change the size of the video (optional).

If you are including more than one YouTube video and are using the code provided, don’t forget to change the id for the <figure>, <media>, and <flash> elements each time you add a new video.

4.3.1.2.5.2 Embedding SlideShare Presentations

If you choose to include SlideShare presentations in your module, you will need to first copy the information provided in the Embed HTML snippet provided by SlideShare for the presentation you wish to use. As an example, consider this slideshow\(^\text{75}\) taken from our standard Connexions presentation. The Embed snippet for this presentation is as follows:

```
<iframe src="http://www.slideshare.net/slideshow/embed_code/1888644?rel=0" width="427" height="356" frameborder="0" marginwidth="0" marginheight="0" scrolling="no" style="border:1px solid #CCC;border-width:1px 1px 0;margin-bottom:5px" allowfullscreen webkitallowfullscreen mozallowfullscreen">
<br />
</iframe>

<div style="margin-bottom:5px">
  <strong>
    <a href="http://www.slideshare.net/cnxorg/cnxdemo" title="CNXDemo" target="_blank">CNXDemo</a>  
    from  
    <strong>
      <a href="http://www.slideshare.net/cnxorg" target="_blank">Connexions</a> 
    </strong>
  </strong>
</div>
```

Since this code is designed for use on websites and not the CNXML language, the majority of this snippet is not necessary. The part that is important is the iframe tag. You will copy that, but delete all the options but height and width:

\(^75\)http://www.slideshare.net/cnxorg/cnxdemo

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
The CNXML example below illustrates how to embed a SlideShare player in your module.

- The `src` attribute is set to the **embed URL** for the slideshow, **not the URL for the page on the SlideShare website**. The embed URL is the one found in the code snippet as shown above and is of the form "http://www.slideshare.net/slideshow/embed_code/[PresentationID]?rel=0" (where [PresentationID] is the ID and title of the slideshow you wish to include in your module).
- After you copy the embed URL, you need to replace any `&` characters with `&amp;`. The `&` character is a special character in CNXML and must be escaped in this manner, otherwise you will get an error when you try to save your module.

The final CNXML code for embedding this presentation is as follows:

```xml
<figure id="slidesharefigure">
  <media id="slidesharemedia" alt="Slide show introducing the ideas behind Connexions.">
    <iframe src="http://www.slideshare.net/slideshow/embed_code/1888644?rel=0" width="425" height="355"/>
  </media>
</figure>
```

And results in the following:

---

**Connexions: Create Globally, Educate Locally**

**Figure 4.40**

---

**TIP** Though the steps described above may look intimidating, the actual process isn’t actually that hard. To embed your SlideShare presentation, just perform the following steps:

Step 1. Copy the example code above into your module.
Step 2. Replace the URL with the one provided in the `<embed>` element in the HTML snippet provided by SlideShare.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Step 3. Escape the ampersand characters in the URL by replacing `&` with `&amp;`.

Step 4. Adjust the height and width parameters to change the size of the presentation (optional).

If you are including more than one SlideShare presentation and are using the code provided, don’t forget to change the `id` for the `<figure>`, `<media>`, and `<flash>` elements each time you add a new video.

### 4.3.1.2.5.3 Embedding Prezi Presentations

If you choose to include Prezi presentations in your module, you will need to first copy the information provided in the **Embed** HTML snippet provided by Prezi for the presentation you wish to use. As an example, consider this presentation\(^{76}\) taken from a presentation developed by UniqU\(^{77}\). The Embed snippet for this presentation is as follows:

```html
<div class="prezi-player">
<style type="text/css" media="screen">
.prezi-player { width: 550px; }
.prezi-player-links { text-align: center; }
</style>
<object id="prezi_n5c7h_bctk1c" name="prezi_n5c7h_bctk1c" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="550" height="400">
<param name="movie" value="http://prezi.com/bin/preziloader.swf" />
<param name="allowfullscreen" value="true" />
<param name="allowscriptaccess" value="always" />
<param name="bgcolor" value="#ffffff" />
<param name="flashvars" value="prezi_id=n5c7h_bctk1c&lock_to_path=1&color=ffffff&autoplay=no" />
<embed id="preziEmbed_n5c7h_bctk1c" name="preziEmbed_n5c7h_bctk1c" src="http://prezi.com/bin/preziloader.swf" type="application/x-shockwave-flash" allowfullscreen="always" width="550" height="400" bgcolor="#ffffff" flashvars="prezi_id=n5c7h_bctk1c&lock_to_path=1&color=ffffff&autoplay=no" />
</object>
</div>
<p>
<a title="How to think about Connexions!" href="http://prezi.com/n5c7h_bctk1c/">UniqU: Prezi</a></p>
</div>

Since this code is designed for use on websites and not the CNXML language, the majority of this snippet is not necessary. The part that is important is the **embed** tag and the **param** tags:

```html
<param name="movie" value="http://prezi.com/bin/preziloader.swf" />
<param name="allowfullscreen" value="true" />
```

---

\(^{76}\) [http://prezi.com/n5c7h_bctk1c/](http://prezi.com/n5c7h_bctk1c/)

\(^{77}\) [http://theuniqu.com](http://theuniqu.com)

Available for free at Connexions [http://cnx.org/content/col10151/1.27]
The CNXML example below illustrates how to embed a Prezi player in your module. Notice that it is implemented like a standalone flash object, with a few key features:

- The src attribute is set to the embed URL for the slideshow, not the URL for the page on the Prezi website. The embed URL is the one found in the code snippet as shown above and is of the form "http://prezi.com/bin/preziloader.swf).  
- Copy all param tags from the embed snippet.

The final CNXML code for embedding this presentation is as follows:

```
<embed id="preziEmbed_n5c7h_bctk1c" name="preziEmbed_n5c7h_bctk1c"
src="http://prezi.com/bin/preziloader.swf" type="application/x-shockwave-flash"
allowfullscreen="true" allowscriptaccess="always" width="550" height="400" bgcolor="#ffffff"
flashvars="prezi_id=n5c7h_bctk1c&lock_to_path=1&color=ffffff&autoplay=no" />
```

And results in the following:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
**TIP:** Though the steps described above may look intimidating, the actual process isn’t actually that hard. To embed your Prezi presentation, just perform the following steps:

Step 1. Copy the example code (p. 104) above into your module.
Step 2. Replace the URL with the one provided in the `<embed>` element in the HTML snippet provided by Prezi.
Step 3. Replace the param tags from the example with those that are provided in the `<embed>` element in the HTML snippet provided by Prezi.
Step 4. Escape the ampersand characters in the URL by replacing `&` with `&amp;`, if any are present.
Step 5. Adjust the height and width parameters to change the size of the presentation (optional).

If you are including more than one Prezi presentation and are using the code provided, don’t forget to change the `id` for the `<figure>`, `<media>`, and `<flash>` elements each time you add a new video.

### 4.3.1.3 Using Basic CNXML in Edit-In-Place

**PLEASE NOTE:** This module also contains information derived from The Advanced CNXML (Section 5.5) by Ricardo Radaelli-Sanchez.

---

**4.3.1.3.1 Starting with CNXML**

To create the bare bones of content in Connexions, the author interface provides a variety of creation tools: the Document Importer, Edit-In-Place, and even a full-source editor. However, a basic knowledge of our markup language can help make small edits into tremendous enhancements to your material!

---

78This content is available online at <http://cnx.org/content/m14394/1.4/>.
79http://cnx.org/member_profile/rars

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Connexions\textsuperscript{80} uses the Connexions Markup Language (CNXML) as its primary language for marking up and storing documents. CNXML is lightweight XML\textsuperscript{81} for marking up educational content. Unlike the well-known HTML, the goal of CNXML is to convey the content of the material and not a particular presentation. For example, say you have the following sentence: I like cupcakes very much. However, you feel that your enthusiasm for cupcakes has not been fully expressed. In HTML, you would use bold, underline, italic, etc.; in CNXML, you would use the emphasis (Section 4.3.1.3.2.1: Emphasis) \textbf{tag}.

### 4.3.1.3.2 Inline Tags

Inline tags, such as emphasis, are used to embed content and functionality inside the structural tags, such as paragraphs. Some of the more commonly used tags are discussed below.

#### 4.3.1.3.2.1 Emphasis

As mentioned previously (p. 106), the emphasis tag is used to accent certain text. Note that this refers to semantic emphasis and not a typeface. Different \textit{stylesheets} can render emphasis with different typefaces.

**Example 4.1**

```xml
<para id='intro'>
  Gardenias are my absolute <emphasis>favorite</emphasis>
  flower. Their petals are soft, and their bloom has an
  absolutely <emphasis>heavenly</emphasis> scent.
</para>
```

The above markup will display as:

Gardenias are my absolute **favorite** flower. Their petals are soft, and their bloom has an absolutely ***heavenly*** scent.

#### 4.3.1.3.2.2 Term

The term tag is used to mark words or phrases which are being defined. However, its use is confined to either a para (Section 4.3.1.4.1: Para) or definition (Section 5.4.6: Definition) tag. The \texttt{term} tag has one optional attribute: \texttt{URL} - a URL specifying the source or definition of the term.

\begin{itemize}
  \item \textsuperscript{80}http://cnx.org
  \item \textsuperscript{81}http://en.wikipedia.org/wiki/XML
\end{itemize}

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Example 4.2

Gardenias can be tricky to maintain. The soil around the roots of a gardenia must remain moist always, but too much water can damage the plant. Also, gardenias enjoy the sun, but if the foliage gets wet to bring the gardenia into the shade.

The above markup will display as:

Gardenias can be tricky to maintain. The soil around the roots of a gardenia must remain moist always, but too much water can damage the plant. Also, gardenias enjoy the sun, but if the foliage gets wet to bring the gardenia into the shade.

4.3.1.3.2.3 Note

The note tag creates an "out of line" note to the reader. The type of note is specified by an optional type attribute. If a type is not specified, the default is Note. The type attribute can contain any of the following values:

- note
- aside
- warning
- tip
- important

Example 4.3

Receiving flowers is, on the whole, a wonderful thing. However, sometimes pollen from the flowers can cause problems. In particular, the clean up of a bit of pollen can be tricky. <note type="Important">Do not use water when cleaning up pollen! This can lead to counter-top and clothing stains!</note> Your best bet is to use a dry method of cleaning with a paper-towel.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
The above markup will display as:

Receiving flowers is, on the whole, a wonderful thing. However, sometimes pollen from the flowers can cause problems. In particular, the clean up of a bit of pollen can be tricky.

**IMPORTANT:** Do not use water when cleaning up pollen! This can lead to countertop and clothing stains!

Your best bet is to use a dry method of cleaning, with a paper-towel, for example.

### 4.3.1.3.2.4 Link

The `link` tag is the tag in CNXML used for linking to other Connexions modules or collections as well as external links.

- **strength:** The Strength attribute can contain the value 1, 2, or 3 specifying the relevance of the link.
- **window:** The Window attribute determines the manner in which the link location will be opened. It can contain the values "Replace" or "New". "Replace" will result in the link location opening in the current window replacing the page with the link. "New" will result in the link location opening in a new browser window.
- **url:** The URL attribute can contain the web address of the link you wish to reference.
- **document:** The Document attribute is used to reference the ids of other Connexions modules or Collections.
- **target-id:** The Target-id attribute is used to reference the ids of specific elements within Connexions modules.
- **resource:**
- **version:** The Version attribute is used to reference a specific version of a Connexions module or collection.

The `target` and `document` attributes can be used together or alone. If both are used then you will link to a particular tag in another document. If only `document` is used, you will link to another document. If only `target` is used, you will link to a particular tag within the current document.

### 4.3.1.3.2.5 Cite

The `cite` tag is used to refer to non-electronic materials within a document, and primarily contains the title, the author, and/or a page number of a work.

**Example 4.4: Cite Example**

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Finally, a good resource is the <cite>Garden Lover’s Cookbook -- William M. Rice; Paperback</cite>.

The above markup will display as:

Finally, a good resource is the *Garden Lover’s Cookbook – William M. Rice; Paperback*.

4.3.1.3.2.6 Quote

The `quote` tag is used to denote that some text directly quotes another source. The `quote` tag has a `display` attribute which denotes whether the quote is *inline* or *block*.

**Example 4.5: Quote Example**

```html
<para id='plantquote'>
Every plant needs a different amount of water in order to grow well. <quote display="inline">"If you water each plant the same, you will always water too much and too little."</quote> Also, remember the words of Lou Erickson:
<quote id="quote_example" display='block'>"Gardening requires lots of water - most of it in the form of perspiration."</quote>
</para>
```

Every plant needs a different amount of water in order to grow well. “If you water each plant the same, you will always water too much and too little.” Also, remember the words of Lou Erickson:

"Gardening requires lots of water - most of it in the form of perspiration."

4.3.1.3.2.7 Foreign

The `foreign` tag is used to denote that a word or phrase foreign to the language of the document is being used.

**Example 4.6: Foreign Example**

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 4. AUTHORING CONTENT

All flowers have a scientific name, often derived from Latin. *Gardenia augusta* is the name of a type of gardenia found in Japan.

All flowers have a scientific name, often derived from Latin. *Gardenia augusta* is the name of a type of gardenia found in Japan.

4.3.1.3.2.8 Code

The `code` tag is used to insert example computer output/input as either inline text within a paragraph or as a block of text. The `code` tag has a `display` attribute with two possible values:

- `inline` (default) - used to specify code that is inline.
- `block` - used to specify code that should be in a separate block of text.

**Example 4.7: Inline Code Example**

For now, take a look at what the inline code looks like:

```
<para id='copy'>
In a unix terminal the command to copy a file is
<code display='inline'>cp original copy</code>.
</para>
```

In a unix terminal the command to copy a file is `cp original copy`

You will see more about code blocks in Advanced CNXML using Edit-In-Place (Section 4.3.1.4.6: Code).

4.3.1.4 Advanced CNXML using Edit-in-Place

4.3.1.4.1 Para

When working in Edit-in-Place, notice that the first item of the "Add Here" drop-down menu is "Paragraph". When you select this item and click Add Here, a text box (Figure 4.42: A Paragraph Box) will appear. You can now insert text in the white box, including inline tags (Section 4.3.1.3.2: Inline Tags). Note the `id="element-143"` in the upper

82This content is available online at <http://cnx.org/content/m14395/1.5/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
left hand part of the blue box in Figure 4.42 (A Paragraph Box). element-143 is the paragraph’s unique ID, which you can use to refer to the paragraph directly using a link (Section 4.3.1.3.2.4: Link) tag. Also, you can find some helpful tips in the upper right-hand corner of the blue box: "Help editing <para>".

---

**A Paragraph Box**

![A Paragraph Box](image)

**Figure 4.42:** When you click "insert and choose paragraph", a box like this should appear.

---

**Example 4.8: Submitted by J. Cameron Cooper**

<para id='intro'>
Working on trees or bushes can generate a lot of limbs and branches to haul away. If you just carry them, it’ll take all day. Instead, make a sledge.
</para>

<para id='intro2'>
Find a large, complex branch to make the base of your sledge. It should be relatively flat, and broad and long enough to make a decent pile; that is, as big or bigger than

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Working on trees or bushes can generate a lot of limbs and branches to haul away. If you just carry them, it’ll take all day. Instead, make a sledge.

Find a large, complex branch to make the base of your sledge. It should be relatively flat, and broad and long enough to make a decent pile; that is, as big or bigger than anything else you need to haul away. Green branches from hardwoods are best. Place it with the cut end pointing the way you want to go. If no single branch is good enough, two can be used. Just place their cut ends a couple feet apart.

Then pile on the remaining branches. Most will naturally weave together; if not, give 'em a little help. Once the pile it a few layers deep, smaller waste, like weeds or maybe even leaves can be added to the pile. If it gets unstable, another big branch will help.

When you’re done, grab the cut end of the bottom branch, and maybe the base of one of the other big branches in the pile, and drag the thing where you want to go. You’ll be surprised how much one person can drag!

If you have a lot of leaves or similar small stuff to move, you can use a similar technique. Get a tarp, toss the leaves and weeds and whatnot in the middle, and then drag the whole thing away.

which displays as the following:

Working on trees or bushes can generate a lot of limbs and branches to haul away. If you just carry them, it’ll take all day. Instead, make a sledge.

Find a large, complex branch to make the base of your sledge. It should be relatively flat, and broad and long enough to make a decent pile; that is, as big or bigger than anything else you need to haul away. Green branches from hardwoods are best. Place it with the cut end pointing the way you want to go. If no single branch is good enough, two can be used. Just place their cut ends a couple feet apart.

Then pile on the remaining branches. Most will naturally weave together; if not, give 'em a little help. Once the pile it a few layers deep, smaller waste, like weeds or maybe even leaves can be added to the pile. If it gets unstable, another big branch will help.

When you’re done, grab the cut end of the bottom branch, and maybe the base of one of the other big branches in the pile, and drag the thing where you want to go. You’ll be surprised how much one person can drag!
If you have a lot of leaves or similar small stuff to move, you can use a similar technique. Get a tarp, toss the leaves and weeds and whatnot in the middle, and then drag the whole thing away.

4.3.1.4.2 List

To insert a new list, select "list" from the "insert" drop-down menu. As with adding a paragraph, adding a list will insert a blue box (Figure 4.43: Lists Available in Edit-in-Place), with the list’s unique ID in the upper left-hand corner and a helpful link in the upper right-hand corner.

![Lists Available in Edit-in-Place](image)

(a) Enumerated List

(b) Bulleted List

**Figure 4.43:** (a) After you add a list, you will see this blue box. You can then select the type of list you wish to use. Here an enumerated list has been selected (b) Here a bulleted list has been selected.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Example 4.9: Enumerated List

<list id='sledge' list-type='enumerated'>
  <title>Making a Sledge</title>
  <item>
    Find a large, complex branch to make the base of your sledge. It should be relatively flat, and broad and long enough to make a decent pile; that is, as big or bigger than anything else you need to haul away. Green branches from hardwoods are best. Place it with the cut end pointing the way you want to go. If no single branch is good enough, two can be used. Just place their cut ends a couple feet apart.
  </item>
  <item>
    Then pile on the remaining branches. Most will naturally weave together; if not, give 'em a little help. Once the pile it a few layers deep, smaller waste, like weeds or maybe even leaves can be added to the pile. If it gets unstable, another big branch will help.
  </item>
  <item>
    When you're done, grab the cut end of the bottom branch, and maybe the base of one of the other big branches in the pile, and drag the thing where you want to go. You'll be surprised how much one person can drag!
  </item>
</list>

The resulting list will look like:

Making a Sledge

1. Find a large, complex branch to make the base of your sledge. It should be relatively flat, and broad and long enough to make a decent pile; that is, as big or bigger than anything else you need to haul away. Green branches from hardwoods are best. Place it with the cut end pointing the way you want to go. If no single branch is good enough, two can be used. Just place their cut ends a couple feet apart.

2. Then pile on the remaining branches. Most will naturally weave together; if not, give 'em a little help. Once the pile it a few layers deep, smaller waste, like weeds or maybe even leaves can be added to the pile. If it gets unstable, another big branch will help.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
3. When you’re done, grab the cut end of the bottom branch, and maybe the base of one of the other big branches in the pile, and drag the thing where you want to go. You’ll be surprised how much one person can drag!

**Example 4.10: Bulleted List**

```xml
<list id="ex-bulleted-list" list-type="bulleted">
  <item>branches</item>
  <item>leaves</item>
  <item>sweat</item>
  <item>lemonade</item>
</list>
```

- branches
- leaves
- sweat
- lemonade

### 4.3.1.4.3 Equation

The *equation* tag is used to set off and number equations in CNXML documents. If you have MathML (Section 4.3.1.6.1) enabled for your document, you will only be able to place MathML equations within the *equation* tags. Otherwise, to write the actual equations, you can use ASCII or images.

**NOTE:** Connexions strongly encourages the use equation with MathML (Section 4.3.1.6.1) tags when displaying math.

If you look at Figure 4.44 (Adding an Equation), you will find the equation’s unique ID in the upper left-hand corner and a helpful link in the upper right-hand corner.
As with lists, you can add an optional title at the beginning of each equation.

**Example 4.11: Using Images as Equations**

\[
\langle\text{equation id="eqn14"}\\
\langle\text{media id="img12" display="block" alt="1+2=3"}\\
\langle\text{image mime-type='image/gif' src='euler.gif'} /\rangle\\
\langle\text{/equation}\\
\langle\text{Save} \quad \langle\text{Cancel}\rangle
\]

displays as:

\[1+2=3\quad (4.3)\]

**Example 4.12: ASCII equations**

\[
\langle\text{equation id='eqn15'}\\
\\langle\text{/equation}\\
\]

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Simple Arithmetic

\[ 11 + 27 = 38 \]

This equation will display as:

Simple Arithmetic

4.3.1.4.4 Exercise

The exercise tag allows authors to add practice problems into their documents. When you initially add an exercise, you will see the familiar blue box (Figure 4.45: Adding an Exercise), with the unique ID and the helpful link in the top corners. However, also notice that new tags have been premade in your text box: problem and solution.

---

**Adding an Exercise**

```xml
<exercise id="eip-704">
  <problem id="eip-132">
    <para id="eip-766">
      Insert Problem Text Here
    </para>
  </problem>

  <solution id="eip-871">
    <para id="eip-994">
      Insert Solution Text Here
    </para>
  </solution>
</exercise>
```

---

**Figure 4.45**

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
To continue utilizing edit-in-place to edit your exercise, press the Save button (see Figure 4.46 (A New Exercise after Saving)). You can now add various block tags to your problem and solution, including paragraphs and lists!

**A New Exercise after Saving**

![Exercise 1: Insert Problem Text Here](image)

**Solution**

Insert Solution Text Here

---

*Figure 4.46:* If you save immediately after creating a new exercise, you can continue to edit the exercise using the familiar edit-in-place interface.

---

To create more complex exercises, such as multiple-choice, multiple-response, ordered-response, and free-response questions, QML (Questions Markup Language) may used in place of the problem and solution tags. For more information, please see the information about QML.\(^{83}\)

**Example 4.13**

```xml
<exercise id='hyd_test'>
<problem id="id9">
<para id='hyd_testp1'>
```

\(^{83}\)“QML 1.0” [http://cnx.org/content/m10136/latest/](http://cnx.org/content/m10136/latest/)

Available for free at Connexions [http://cnx.org/content/col10151/1.27](http://cnx.org/content/col10151/1.27)
The color of a hydrangea changes with the pH of the soil. What color would the hydrangea be if the soil were highly acidic? Highly basic? Neutral?

This code will display as:

**Problem**
The color of a hydrangea changes with the pH of the soil. What color would the hydrangea be if the soil were highly acidic? Highly basic? Neutral?

**Solution**
Highly acidic soil produces blue flowers. Highly basic soil produces pink flowers. Neutral soil produces very pale cream flowers.

### 4.3.1.4.5 Figure
The *figure* tag provides the structure for creating a figure within a document. They can contain either two or more subfigure (Section 5.4.3: Subfigure) tags, or a single media (Section 4.3.1.2), table (Section 4.3.1.5), or code (Section 4.3.1.4.6: Code) tag.
Adding a Figure

Figure 4.47: Adding a figure will create this familiar blue box, with a helpful link in the upper right corner and the figure’s unique ID in quotes in the upper left corner.

The optional first tag of the figure tag is title (Section 5.3.2.2: Title) which is used to title a figure.

The title tag is followed by any of the tags listed above; however, the most commonly used tag is media, which is used to include any sort of media such as images, video, music, or java applets. For more information on what media you can add to your content, and how to add it, see Adding Multimedia to Your Connexions Content (Section 4.3.1.2).

The final tag is the optional caption which is used to add a small caption to the figure.

Example 4.14: Example of a Figure

<figure id='blossom'>

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
This code will display as:

Momosa Blossom

Figure 4.48: Picture taken by Jenn Drummond (CC Attribution).
4.3.1.4.6 Code

As seen in Using Basic CNXML in Edit-in-Place (Section 4.3.1.3.2.8: Code), you can add inline code to your document; edit-in-place also allows you to insert a block of code (Figure 4.49: Adding a Block of Code), separate from text.

Adding a Block of Code

![Adding a Block of Code](image)

Figur 4.49: Note that code has a required unique ID if and only if the display attribute is block.

If you need to use the > and < symbols in your block of code, you must either use the unicode for these characters (&gt; and &lt;, if you have MathML enabled), or use the CDATA method. To utilize the CDATA method, insert <!--[CDATA[ before your code and ]]> after it, as seen in Example 4.15 (A Block of Code, Using CDATA).

Example 4.15: A Block of Code, Using CDATA

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Using CDATA in a Code Block

When saved, Figure 4.50 (Using CDATA in a Code Block) will display as:

```xml
<para id='copy'>
  In a unix terminal the command to copy a file is
  <code display='inline'>cp original copy</code>.
</para>
```

4.3.1.4.7 Note

As mentioned in Using Basic CNXML in Edit-in-Place (Section 4.3.1.3.2.3: Note), the note tag creates an "out of line" note to the reader. You can also insert a note using the drop-down box in Edit-in-Place; however, unless you edit the full source, the type of note will be set to the default.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Adding a Note using Edit-in-Place

![Screenshot of the Edit-in-Place note feature]

**Figure 4.51:** As with code, notes require a unique ID when the display attribute is "block".

**Example 4.16**

```
<note>
  Gardening requires a lot of intense physical exertion.
  Please drink plenty of water to avoid dehydration!
</note>
```

The above markup will display as:

**NOTE:** Gardening requires a lot of intense physical exertion. Please drink plenty of water to avoid dehydration!
4.3.1.4.8 Example

As is often the case in textbooks, authors will include examples in the middle of a chapter or section. For this reason CNXML provides the example (Figure 4.52: Adding an Example Using Edit-in-Place) tag that allows an author to include examples in a document.

---

Adding an Example Using Edit-in-Place

![Adding an Example Using Edit-in-Place](image)

Figure 4.52

---

Example 4.17

Here is the code for Example 4.16:

```xml
<example id="notexamp">
  <code id="codeseg1" display="block">
    <note>
      Gardening requires a lot of intense physical exertion. Please drink plenty of water to avoid dehydration!
    </note>
  </code>
</example>
```

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
The above markup will display as:

\[
\text{Gardening requires a lot of intense physical exertion. Please drink plenty of water to avoid dehydration!}
\]

4.3.1.4.9 CALS Table

The final element you can add using Edit-in-Place is \texttt{table}. To learn more about adding and editing tables using Edit-in-Place, see CALS Table (Section 4.3.1.5). For a more complete description of the CALS Table consult the CALS Table Spec\textsuperscript{84}.

4.3.1.5 CALS Table\textsuperscript{85}

CNXML uses the industry standard CALS Table Model\textsuperscript{86} for including tables into CNXML documents. When you first insert a table using Edit-in-Place, you will see a blue window similar to the one shown in Figure 4.53 (Adding a Table using Edit-in-Place).

\begin{center}
\textbf{Adding a Table using Edit-in-Place}
\end{center}

\begin{center}
\textit{Image not finished}
\end{center}

Figure 4.53

Using the Edit-in-Place interface, you can add a name and change how many rows or columns your table needs.

\textbf{IMPORTANT}: Be sure to set the correct dimensions before you start editing the entries of the table, as the Resize table button will reset the entries to their default values.

\textsuperscript{84}\url{http://www.oasis-open.org/specs/a502.htm}
\textsuperscript{85}This content is available online at \url{<http://cnx.org/content/m14396/1.5/>}. 
\textsuperscript{86}\url{http://www.oasis-open.org/specs/a502.htm}

Available for free at Connexions \url{<http://cnx.org/content/col10151/1.27>
Once your table is correctly sized, you can start editing the entries of your table, as shown in Figure 4.54 (Editing Entries of a CALS Table).

Provided below is a brief description of the CALS tags, their attributes, and children (along with a helpful example). For a more complete description of the CALS Table consult the CALS Table Spec\(^87\).

4.3.1.5.1 table

The `table` tag marks the beginning of a table. It has an optional first child of name (Section 5.3.2.2: Title) (as you can see in the Edit-in-Place interface) and must contain one or more `tgroup` (p. 127) tags. The Edit-in-Place interface, however, only supports the utilization of one `tgroup`. The `table` tag also has many attributes, to find out more information consult the CALS Table Spec\(^88\).

4.3.1.5.2 tgroup

The `tgroup` tag marks the beginning of a new portion of a table (p. 127). It has a required attribute `cols` which is the number of columns in the `tgroup`; as mentioned previously (p. 126), this attribute can be set using the Edit-in-Place interface. Its children tags are zero, one, or more `colspec` (p. 127) or `spanspec` (p. 128), zero or one `thead` (p. 128) or `tfoot` (p. 128), and one `tbody` (p. 128) tag.

4.3.1.5.3 colspec

The `colspec` tag is an empty tag that specifies the column of a table (p. 127) or `entrytbl` (p. 128). The names and numbers specified as attributes are used for referencing by other tags.

\(^{87}\)http://www.oasis-open.org/specs/a502.htm

\(^{88}\)http://www.oasis-open.org/specs/a502.htm

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.1.5.4 spanspec

The spanspec tag is an empty tag that identifies a horizontal span of columns and associated attributes that can subsequently be referenced by its spanname for repeated use in entry (p. 128) or entrytbl (p. 128) in different rows (p. 128).

4.3.1.5.5 thead

The thead tag identifies the heading row (p. 128) of a tgroup (p. 127) or entrytbl (p. 128). The thead tag can have zero, one, or more colspec (p. 127) tags and one or more row (p. 128).

4.3.1.5.6 tfoot

The tfoot tag identifies the rows (p. 128) of footer information that are displayed after the tbody (p. 128). The tfoot tag can have zero, one, or more colspec (p. 127) tags and one or more row (p. 128).

4.3.1.5.7 tbody

The tbody tag identifies the body of a tgroup (p. 127) or entrytbl (p. 128). The tbody tag must have one or more row (p. 128) tags.

4.3.1.5.8 row

The row tag identifies the row of information in a thead (p. 128), tbody (p. 128), or tfoot (p. 128). The row tag must have one or more entry (p. 128) or entrytbl (p. 128) tags.

4.3.1.5.9 entrytbl

The entrytbl tag takes the place of an entry (p. 128), but fits into a single row (p. 128) of tbody (p. 128) in a tgroup (p. 127). The content model is the same as that of a tgroup (p. 127) except that tfoot (p. 128) is ommitted and entrytbl is self-excluding. Its children tags are zero, one, or more colspec (p. 127) or spanspec (p. 128), zero or one thead (p. 128) or tfoot (p. 128), and one tbody (p. 128) tag.

4.3.1.5.10 entry

The entry tag identifies an entry in a row (p. 128). The entry tag contains ASCII text and zero, one, or many cite (Section 4.3.1.3.2.5: Cite), term (Section 4.3.1.3.2.2: Term), cnxn (Section 4.3.1.3.2.4: Link), link (Section 4.3.1.3.2.4: Link), code (Section 4.3.1.3.2.8: Code), emphasis (Section 4.3.1.3.2.1: Emphasis), or media (Section 4.3.1.2).
4.3.1.5.11 Using CALS Tables

It might sound a little confusing but I think that the best way to understand a table is to look at Example 4.18. For more information, consult the CALS Table Spec\(^ {89} \).

Example 4.18

![Image not finished](http://www.oasis-open.org/specs/a502.htm)

Figure 4.55

```xml
<thead>
  <row>
    <entry>Month</entry>
    <entry>Flower</entry>
  </row>
</thead>
<tbody>
  <row>
    <entry align="center">January</entry>
    <entry align="center">Carnation</entry>
  </row>
  <row>
    <entry align="center">February</entry>
    <entry align="center">Violet</entry>
  </row>
  <row>
    <entry align="center">March</entry>
    <entry align="center">Daffodil</entry>
  </row>
  <row>
    <entry align="center">April</entry>
    <entry align="center">Sweet Pea</entry>
  </row>
  <row>
    <entry align="center">May</entry>
    <entry align="center">Sunflower</entry>
  </row>
```

\(^ {89}\)http://www.oasis-open.org/specs/a502.htm

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
<table>
<thead>
<tr>
<th>Month</th>
<th>Flower</th>
</tr>
</thead>
<tbody>
<tr>
<td>June</td>
<td>Honeysuckle</td>
</tr>
<tr>
<td>July</td>
<td>Larkspur</td>
</tr>
<tr>
<td>August</td>
<td>Lily</td>
</tr>
<tr>
<td>September</td>
<td>Forget-Me-Not</td>
</tr>
<tr>
<td>October</td>
<td>Marigold</td>
</tr>
<tr>
<td>November</td>
<td>Chrysanthemum</td>
</tr>
<tr>
<td>December</td>
<td>Narcissus</td>
</tr>
</tbody>
</table>

This code will end up looking like this (Table 4.1: Birth Flowers):
Birth Flowers

<table>
<thead>
<tr>
<th>Month</th>
<th>Flower</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>Carnation</td>
</tr>
<tr>
<td>February</td>
<td>Violet</td>
</tr>
<tr>
<td>March</td>
<td>Daffodil</td>
</tr>
<tr>
<td>April</td>
<td>Sweet Pea</td>
</tr>
<tr>
<td>May</td>
<td>Sunflower</td>
</tr>
<tr>
<td>June</td>
<td>Honeysuckle</td>
</tr>
<tr>
<td>July</td>
<td>Larkspur</td>
</tr>
<tr>
<td>August</td>
<td>Lily</td>
</tr>
<tr>
<td>September</td>
<td>Forget-Me-Not</td>
</tr>
<tr>
<td>October</td>
<td>Marigold</td>
</tr>
<tr>
<td>November</td>
<td>Chrysanthemum</td>
</tr>
<tr>
<td>December</td>
<td>Narcissus</td>
</tr>
</tbody>
</table>

Table 4.1

4.3.1.6 MathML Editor

4.3.1.6.1 Content MathML

The authoritative reference for Content MathML is Section 4 of the MathML 2.0 Specification. The World Wide Web Consortium (W3C) is the body that wrote the specification for MathML. The text is very readable and it is easy to find what you are looking for. Look there for answers to questions that are not answered in this tutorial or when you need more elaboration. This tutorial is based on MathML 2.0.

In this document, the m prefix is used to denote tags in the MathML namespace. Thus the <apply> tag is referred to as <m:apply>. Remember all markup in the MathML namespace must be surrounded by <m:math> tags.

---

90This content is available online at <http://cnx.org/content/m9008/2.15/>.
91http://www.w3.org/TR/MathML2/chapter4.html
4.3.1.6.1.1 The Fundamentals of Content MathML: Applying Functions and Operators

The fundamental concept to grasp about Content MathML is that it consists of applying a series of functions and operators to other elements. To do this, Content MathML uses prefix notation. **Prefix notation** is when the operator comes first and is followed by the operands. Here is how to write "2 plus 3".

```
<m:math>
  <m:apply>
    <m:plus/>
    <m:cn>2</m:cn>
    <m:cn>3</m:cn>
  </m:apply>
</m:math>
```

This would display as \(2 + 3\).

There are three types of elements in the Content MathML example shown above. First, there is the apply tag, which indicates that an operator (or function) is about to be applied to the operands. Second, there is the function or operator to be applied. In this case the operator, plus, is being applied. Third, the operands follow the operator. In this case the operands are the numbers being added. In summary, the apply tag applies the function (which could be sin or \(f\), etc.) or operator (which could be plus or minus, etc.) to the elements that follow it.

4.3.1.6.1.1.1 Tokens

Content MathML has three tokens: ci, cn, and csymbol. A **token** is basically the lowest level element. The tokens denote what kind of element you are acting on. The cn tag indicates that the content of the tag is a number. The ci tag indicates that the content of the tag is an identifier. An **identifier** could be any variable or function; \(x\), \(y\), and \(f\) are examples of identifiers. In addition, ci elements can contain Presentation MathML. Tokens, especially ci and cn, are used profusely in Content MathML. Every number, variable, or function is marked by a token.

**csymbol** is a different type of token from ci and cn. It is used to create a new object whose semantics is defined externally. It can contain plain text or Presentation MathML. If you find that you need something, such as an operator or function, that is not defined in Content MathML, then you can use csymbol to create it.

Both ci and csymbol can use Presentation MathML to determine how an identifier or a new symbol will be rendered. To learn more about Presentation MathML see Section 3 of

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
the MathML 2.0 Specification\(^{92}\). For example, to denote "\(x\) with a subscript 2", where the 2 does not have a more semantic meaning, you would use the following code.

\[
\begin{align*}
\text{\textless m:math} & \\
\text{\textless m:ci} & \\
\text{\textless m:msub} & \\
\text{\textless m:mi}\text{x}\text{\textgreater} & \\
\text{\textless m:mn}\text{2}\text{\textgreater} & \\
\text{\textgreater} & \\
\text{\textless /m:msub} & \\
\text{\textless /m:ci} & \\
\text{\textless /m:math} & 
\end{align*}
\]

This would display as \(x_2\).

The \textit{ci} elements have a type attribute which can be used to provide more information about the content of the element. For example, you can declare the contents of a \textit{ci} tag to be a function (type='fn'), or a vector (type='vector'), or a complex number (type='complex'), as well as any number of other things. Using the type attribute helps encode the meaning of the math that you are writing.

4.3.1.6.1.1.2 Functions and Operators

In order to apply a function to a variable, make the function the first argument of an \textit{apply}. The second argument will be the variable. For example, you would use the following code to encode the meaning, "the function \(f\) of \(x\)." (Note that you have to include the attribute type='fn' on the \textit{ci} tag denoting \(f\)).

\[
\begin{align*}
\text{\textless m:math} & \\
\text{\textless m:apply} & \\
\text{\textless m:ci type='fn'} f\text{\textgreater} & \\
\text{\textless m:ci} x & \\
\text{\textgreater} & \\
\text{\textless /m:apply} & \\
\text{\textless /m:math} & 
\end{align*}
\]

This will display as \(f(x)\).

There are also pre-defined functions and operators in Content MathML. For example, sine and cosine are predefined. These predefined functions and operators are all \textit{empty tags} and they directly follow the \textit{apply} tag. "The sine of \(x\)" is similar to the example above.

\(^{92}\)http://www.w3.org/TR/MathML2/chapter3.html

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
This will display as $\sin(x)$.

You can find a more thorough description of the different predefined functions in Chapter 4 of the MathML specification.

In addition to the predefined functions, there are also many predefined operators. A few of these are plus (for addition), minus (for subtraction), times (for multiplication), divide (for division), power (for taking the $n$th-power of something), and root (for taking the $n$th-root of something).

Most operators expect a specific number of child tags. For example, the power operator expects two children. The first child is the base and the second is the value in the exponent. However, there are other tags which can take many children. For example, the plus operator merely expects one or more children. It will add together all of its children whether there are two or five. This is referred to as an n-ary operator.

Representing "the negative of a variable" and explicitly representing "the positive of a variable or number" has slightly unusual syntax. In this case you apply the plus or minus operator to the variable or number, etc., in question. The following is the code for "negative $x$.”

This will display as $-x$.

In contrast to representing the negative of a variable, the negative of a number may be coded as follows:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
This will display as $-1$.

To create more complicated expressions, you can nest these bits of apply code within each other. You can create arbitrarily complex expressions this way. "a times the quantity $b$ plus $c$" would be written as follows.

\[
\begin{align*}
\text{\textless m:math} \\
\text{\textless m:apply} \\
\text{\textless m:times/} \\
\text{\textless m:ci}a\text{\textgreater} \\
\text{\textless m:apply} \\
\text{\textless m:plus/} \\
\text{\textless m:ci}b\text{\textgreater} \\
\text{\textless m:ci}c\text{\textgreater} \\
\text{\textless /m:apply} \\
\text{\textless /m:apply} \\
\text{\textless /m:math}
\end{align*}
\]

This will display as $a(b + c)$.

The eq operator is used to write equations. It is used in the same way as any other operator. That is, it is the first child of an apply. It takes two (or more) children which are the two quantities that are equal to each other. For example, "a times b plus a times c equals a times the quantity $b$ plus $c$" would be written as shown.

\[
\begin{align*}
\text{\textless m:math} \\
\text{\textless m:apply} \\
\text{\textless m:eq/} \\
\text{\textless m:apply} \\
\text{\textless m:plus/} \\
\text{\textless m:apply} \\
\text{\textless m:times/} \\
\text{\textless m:ci}a\text{\textgreater} \\
\text{\textless m:ci}b\text{\textgreater} \\
\text{\textless /m:apply} \\
\text{\textless m:apply} \\
\text{\textless m:times/} \\
\text{\textless m:ci}a\text{\textgreater} \\
\text{\textless m:ci}c\text{\textgreater} \\
\text{\textless /m:apply} \\
\text{\textless /m:apply} \\
\end{align*}
\]
This will display as \( ab + ac = a(b + c) \).

### 4.3.1.6.1.2 Integrals

The operator for an integral is \( \text{int} \). However, unlike the operators and functions discussed above, it has children that define the independent variable that you integrate with respect to (bvar) and the interval over which the integral is taken (use either lowlimit and uplimit, or interval, or condition). lowlimit and uplimit (which go together), interval, and condition are just three different ways of denoting the integrands. Don’t forget that the bvar, lowlimit, uplimit, interval, and condition children take token elements as well. The following is "the integral of \( f \) of \( x \) with respect to \( x \) from 0 to \( b\)."

\[
\int_{0}^{b} f(x) \, dx
\]

This will display as \( \int_{0}^{b} f(x) \, dx \).

### 4.3.1.6.1.3 Derivatives

The derivative operator is \( \text{diff} \). The derivative is done in much the same way as the integral. That is, you need to define a base variable (using bvar). The following is "the
derivative of the function $f$ of $x$, with respect to $x$.

\[
\frac{d}{dx} f(x)
\]

This will display as $\frac{d}{dx} f(x)$.

To apply a higher level derivative to a function, add a degree tag inside of the bvar tag. The degree tag will contain the order of the derivative. The following shows "the second derivative of the function $f$ of $x$, with respect to $x$.

\[
\frac{d^2}{dx^2} f(x)
\]

This will display as $\frac{d^2}{dx^2} f(x)$.

4.3.1.6.1.4 Vector and Matrices

Vectors are created as a combination of other elements using the vector tag.
<m:apply>
  <m:plus/>
  <m:ci>x</m:ci>
  <m:ci>y</m:ci>
</m:apply>
<m:ci>z</m:ci>
<m:cn>0</m:cn>
</m:vector>
</m:math>

This will display as \[
\begin{pmatrix} x + y \\ z \\ 0 \end{pmatrix}.
\]

Matrices are done in a similar manner. Each matrix element contains several matrixrow elements. Then each matrixrow element contains several other elements.

<\m:math>
  <\m:matrix>
    <\m:matrixrow>
      <m:ci>a</m:ci>
      <m:ci>b</m:ci>
      <m:ci>c</m:ci>
    </m:matrixrow>
    <\m:matrixrow>
      <m:ci>d</m:ci>
      <m:ci>e</m:ci>
      <m:ci>f</m:ci>
    </m:matrixrow>
    <\m:matrixrow>
      <m:ci>g</m:ci>
      <m:ci>h</m:ci>
      <m:ci>j</m:ci>
    </m:matrixrow>
  </m:matrix>
</\m:math>

This will display as \[
\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & j \end{pmatrix}.
\]
There are also operators to take the determinant and the transpose of a matrix as well as to select elements from within the matrix.

4.3.1.6.1.5 Entities

NOTE: The use of MathML character entity references in Connexions content is deprecated.

MathML defines its own entities for many special characters used in mathematical notation. While the entity references have the advantage of being mnemonic with respect to the characters they stand for, they also entail some technical limitations, and so their use in Connexions content is deprecated. Please use the UTF-8-encoded Unicode characters themselves where possible, or, failing that, the XML Unicode character references for the characters. At some time in the future, the Connexions repository system will likely convert entity references and character references silently to the UTF-8-encoded Unicode characters they stand for. See 6.2.1 Unicode Character Data\(^93\) from the XML Specification for more information. The MathML specification contains a list of character entities with their corresponding Unicode code points\(^94\).

There are character picker utilities available to help you select and paste UTF-8 characters into applications like Connexions. If you are running Microsoft Windows, the Windows accessory Character Map can help you. The "Lucida Sans Unicode" font seems to have a good selection of mathematical operators and special characters. Under Linux, the charmmap utility and GNOME applet provide access to all Unicode characters.

4.3.1.6.1.6 Other Resources

There is a lot more that can be done with Content MathML. Especially if you are planning on writing a lot of Content MathML, it is well worth your time to take a look at the MathML specification\(^95\).

4.3.1.6.2 Introduction to the MathML Editor\(^96\)

4.3.1.6.2.1 Math Editor Features

This module explains how to open the Math Editor, create math, edit existing math, and keyboard shortcuts. There is also a separate tutorial page (Section 4.3.1.6.4) with examples showcasing the features.

\(^93\)http://www.w3.org/TR/2003/REC-MathML2-20031021/chapter6.html#chars.unicodechars
\(^94\)http://www.w3.org/TR/MathML2/bycodes.html
\(^95\)http://www.w3.org/TR/MathML2/
\(^96\)This content is available online at <http://cnx.org/content/m24561/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
At the end of this module are nuances and limitations (Section 4.3.1.6.2.15: Nuances / Limitations) of the editor. Please, let us know which ones you’d really like to see incorporated!

4.3.1.6.2.2 Opening the Editor

When editing a Module using Mozilla’s Firefox browser, click on a part of the module to open a blue editing box (Figure 4.56). On the top-right hand side of the box is a "MathML Editor" link which will open up the editor (Figure 4.57).

![Figure 4.56: Begin editing and in the top-right corner is a MathML Editor link](http://cnx.org/content/col10151/1.27)
4.3.1.6.2.3 Using the Popup Window

The popup window (Figure 4.57) can remain open while editing a module and can even stay open while editing several modules. At any point one can close the window, but the contents of the editor will be lost.

4.3.1.6.2.4 User Interface

The Editor has 4 main sections, detailed below. The toolbar (Section 4.3.1.6.2.5: Toolbar) provides a way to insert new operations, a navigation tree to show where the cursor is located, and standard buttons for undo, preview, and source editing.

The main editing area (Section 4.3.1.6.2.6: Editing Area) is located below the toolbar and contains the math that is being edited.
4.3.1.6.2.5 Toolbar

The toolbar contains a row of buttons representing categories of different mathematical operations. These are enabled when something is selected in the editing area (Section 4.3.1.6.2.6: Editing Area). **Note:** The editor does not infer multiplication and addition. See Nuances (Section 4.3.1.6.2.15: Nuances / Limitations) for how to insert next to existing math by wrapping (Section 4.3.1.6.2.15.2: Wrapping Math with Math) existing math. Explain the different sections, when it’s enabled, how things get inserted, and Keyboard Entry (Section 4.3.1.6.2.12: Keyboard Input) for things.

**Menu**
Clicking a category in the top row of the toolbar will open a menu of mathematical operations to insert. To the right of each operation is a name that can be entered from the keyboard at the cursor position in the main editing area (Section 4.3.1.6.2.6: Editing Area). For example, instead of using the toolbar to enter the symbol for pi, the user could enter "pi" or "&pi;" (the MathML version of pi) in the cursor and press the Enter key.

**Path**
Shows the path to the cursor location. Math is organized in a tree-like hierarchy (see Navigating Math (Section 4.3.1.6.2.7: Math Tree)) and the path represents where in the tree the cursor (Section 4.3.1.6.2.10: Cursor) currently is. The path (and context (Section 4.3.1.6.2.11: Context)) are important because they define what can be inserted and where it will go.

**Undo/Redo**
These buttons allow the user to undo an operation such as deletion or insertion. See Keyboard Shortcuts (Section 4.3.1.6.2.12: Keyboard Input) for details on using these features from just the keyboard.

**Preview**
Shows what math will look like when module is published. To resume editing, one must click the Preview button a second time.

**View Source**
Math in Connexions is represented in an XML format known as MathML. Clicking the View Source button will allow editing of the raw MathML.

4.3.1.6.2.6 Editing Area

This is the main area for creating math. It begins empty, but math can be pasted directly in here from Connexions. The tutorials (Section 4.3.1.6.4) contain instructions on moving math from Connexions to the math editor and back. The editing area is the most important part of the editor and as several subsections, outlined below:

- Math is structured like a tree (Section 4.3.1.6.2.7: Math Tree).
• Colors (Section 4.3.1.6.2.8: Colors) are used in this area to denote required information and contextual clues.
• Content vs Presentation Math (Section 4.3.1.6.2.9: Content vs. Presentation) discusses the two different types of math the editor supports.
• The cursor (Section 4.3.1.6.2.10: Cursor) is discussed in detail below, including navigation and different editing modes.
• Since the exact location of the cursor may at times be ambiguous, the context (Section 4.3.1.6.2.11: Context) provides visual cues.
• Keyboard (Section 4.3.1.6.2.12: Keyboard Input) strokes are discussed in detail.
• Finally, empty blocks (Section 4.3.1.6.2.14: Blocks) are discussed below.

4.3.1.6.2.7 Math Tree

Math in the editor is structured like a tree. It can be thought of as removing the precedence rules and just having parentheses. For example, the formula "a*x^2+b*x+c=0" which is displayed (using the editor) as \(ax^2 + bx + c = 0\) and as a tree would look like Figure 4.58. The equal sign has the least precedence and so is on the top. Similarly, \(x\) binds tighter to 2 through the power operation than to \(a\) through the times operation.

![Figure 4.58: \(ax^2 + bx + c = 0\) as a tree](http://cnx.org/content/col10151/1.27)
4.3.1.6.2.8 Colors

Color notation (legend)

\[
f = \begin{cases} 
  x_1 & \text{if } x < 0 \\
  x_2 & \text{if } x > 0 \\
  \text{otherwise}
\end{cases}
\]

- \( f \): The location where text is currently being entered is represented as a box with a blue border (see Text Input (Table 4.3: Text Input Examples) for more information on how to enter math).
- \( x \) and \( x_1 \): Content MathML is represented in black while Presentation MathML is in a dark green (See Content vs. Presentation (Section 4.3.1.6.2.9: Content vs. Presentation) for editing Presentation MathML).
- \( x < 0 \): The cursor context (when the cursor is next to a complex expression) is represented by having a gray background. See Context (Section 4.3.1.6.2.11: Context) for details.
- and : Empty blocks that need to be filled are denoted with a yellow background and optional blocks that can be filled but do not need to be filled are transparent with a dotted border. See Blocks (Section 4.3.1.6.2.14: Blocks) for details.
- \( x_2 \): The current selection is denoted by a light blue background. See Copy and Paste back to Modules (Categories, p. 147) for details.

4.3.1.6.2.9 Content vs. Presentation

There are two subsets of the MathML language; Content MathML and Presentation MathML. Content, as the name implies, focuses on expressing operations like addition, integration, matrices, etc. Presentation focuses on how precisely math is displayed and contains elements like tables and subscripts.
Comparing Content and Presentation MathML

Figure 4.59: Content MathML typically has fewer places to enter information and navigation is simpler, and Presentation MathML allows the user to tweak the way formulas are presented and is used by OCR and import software.

The editor supports creating and editing the Content Math subset while being able to navigate through Presentation MathML. Every thing that is entered into the Editor is entered as Content Math. For example, entering $a*x^2+b*x+c=0$ will be translated as the variable $a$ times $x$ to the power of 2 and added to $b$ times $x$ ...

4.3.1.6.2.10 Cursor

The Math Editor can be used entirely from the keyboard (See Keyboard Input (Section 4.3.1.6.2.12: Keyboard Input)). The cursor can be in one of four places. Either it is editing a variable or number, editing an empty block of text, next to a complicated expression, or has selected an expression. In each of these places there are several things that can be done.

Editing a variable, number, or block
At this point, the cursor is surrounded by a blue box and the user can type in expressions or even paste existing MathML. The expression will be parsed as soon as the cursor
leaves the box or presses the Enter key (in the case of an expression) or immediately when MathML is pasted in. The user can leave the box by pressing clicking on the toolbar or by pressing the Left, Right, or Tab key. See Keyboard Input (Section 4.3.1.6.2.12: Keyboard Input) for more on expressions.

Next to a Complicated Expression
When a cursor is next to a complicated expression, the expression is shown with a light gray background (See Context (Section 4.3.1.6.2.11: Context)). From this point, one of three things may be done. The user may add on to the expression. This is done by just typing. For example, if the cursor is to the left of \((-\pi)i\), the user may type \(-1 = e^{-\pi i}\) and parse the expression (Section 4.3.1.6.2.12: Keyboard Input) to yield \(-1 = e^{-\pi i}\)

One can select the expression by either pressing Shift+Right/Left (depending on whether the cursor is before or after the element), Delete, or Backspace key. See Selection (Section 4.3.1.6.2.10.1) for what can be done next.

4.3.1.6.2.10.1
Selection
When an expression is selected, several things can be done:

- Pressing the Delete or Backspace key will remove it
- Pressing Ctrl+X/C will cut/copy it
- Pressing Ctrl+V will replace the selection with the contents of the clipboard
- Clicking an item in the toolbar will replace the selected item

4.3.1.6.2.11 Context
Instead of using parentheses to denote which operations are grouped, the math editor highlights the current context for the operation. The context shows the position of the cursor (Section 4.3.1.6.2.10: Cursor) relative to existing math in the editing area (Section 4.3.1.6.2.6: Editing Area) and is displayed using a gray background (Section 4.3.1.6.2.8: Colors). An example of a confusing position can be shown using the following example. Suppose the editor contains the term \(a + bc\) and the cursor is just after the \(c\). If the user enters "^2" it is not clear what should be squared. At that position the user may want to square \(c, bc,\) or the entire term \(a + bc\). This produces very different math, namely \(a + bc^2, a + (bc)^2,\) and \((a + bc)^2\). In the above example, the context would highlight precisely the math that ended up being in parentheses. One can think of the context as defining where the parentheses should go once the new math is entered.

4.3.1.6.2.12 Keyboard Input
There are several places the user can enter text into the editor. Most of them behave the same way, but listed below are common uses and specifics:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Common for all Text Entry Points

- Pressing the Enter key or moving the cursor out of the text box (by pressing the Left/Right, Tab key, or clicking elsewhere) after entering will cause the Math to be parsed.
- If the text cannot be converted to Math, it will appear with a red dashed line beneath it (like a spelling error) and must be corrected before saving.
- Simple algebraic expressions, logic operations, trigonometric functions, and subscripts can be entered and will be converted into math.
- If a shorthand notation exists for an operation, it will show up in the toolbar next to the name of the operation (See Toolbar (Section 4.3.1.6.2.5: Toolbar)). Shorthand notation is usually more natural (the operation, like addition, is between its arguments, like \(a+2\))
- If a shorthand notation does not exist for an operation, one can still enter the operation using the keyboard by typing the name of the operation which is also found in the menu (See Toolbar (Section 4.3.1.6.2.5: Toolbar))

Categories
There are three categories of key presses and are enumerated in the table below.

- Shortcuts are preceded by pressing the Ctrl key (or the \(\text{U+2318}\) key on Apple computers)
- Navigation keys move the cursor through the math
- Modification keys change the math in some way

<table>
<thead>
<tr>
<th>Category</th>
<th>Key</th>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+ (Apple (\text{U+2318})+)</td>
<td>X</td>
<td>Math is selected</td>
<td>Cuts the selected Math to the clipboard and replaces it with an empty block (that can be deleted)</td>
</tr>
</tbody>
</table>

*continued on next page*
<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Math is selected</td>
</tr>
<tr>
<td>V</td>
<td>Math is selected</td>
</tr>
<tr>
<td>Z</td>
<td>Ctrl+Z was just pressed</td>
</tr>
<tr>
<td>Y</td>
<td>Redoes one step in the editor</td>
</tr>
<tr>
<td>E</td>
<td>Opens full-source editing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Navigation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab Shift+Tab</td>
<td>Moves to the next/previous free block</td>
</tr>
<tr>
<td>Left / Right</td>
<td>Moves to the previous/next element in the Math</td>
</tr>
<tr>
<td>Shift+Left / Shift+Right</td>
<td>Selects the Context element (right next to the cursor)</td>
</tr>
<tr>
<td>Enter</td>
<td>Attempts to parse the text entered next to the cursor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>After / Before the Context (Section 4.3.1.6.2.11: Context)</td>
<td>Selects the Context's parent</td>
</tr>
<tr>
<td>Before / After the Context</td>
<td></td>
</tr>
</tbody>
</table>

continued on next page

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
<table>
<thead>
<tr>
<th>Delete / Backspace</th>
<th>Cursor next to Math</th>
<th>Selects the Math Node (subsequent delete will remove the math)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math selected</td>
<td>Removes the node and replaces it with an empty block (a second press will remove the block as well)</td>
<td></td>
</tr>
<tr>
<td>Cursor in block</td>
<td>Removes the empty block if it is allowed in MathML (in &quot;a+b+c&quot; any one variable can be removed, but addition requires at least 2 things to add)</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.2**

**Text Input Examples**

<table>
<thead>
<tr>
<th>Type</th>
<th>Input</th>
<th>Math Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculator</td>
<td>(a x^2 + b x + c = 1/2)</td>
<td>(ax^2 + bx + c = \frac{1}{2})</td>
</tr>
<tr>
<td></td>
<td>(a \land b \lor c \neq a</td>
<td></td>
</tr>
</tbody>
</table><p>ightarrow b) | (a \land (b \lor c) \neq aightarrow b) |
|           | (\sin(x)^2 + \cos(x)^2 = 1) | (\sin^2(x) + \cos^2(x) = 1) |
|           | (x_1 + x_2 &lt; x_3) | (x_1 + x_2 &lt; x_3) |
| Templates | (\text{sum} = n*(n-1)/2) | (\sum = \frac{n(n-1)}{2}) |
| MathML    | (&lt;\pi/&gt;) | (\pi) |</p>

[continued on next page]
4.3.1.6.2.13 Text Entry

This is a text entry place. See shortcuts. can paste MathML (Ctrl+V from Mathematica, MathType, etc), or enter simple algebra (see Shortcuts). Moving away using Enter, Tab, Left, Right will cause the input to be parsed and converted into Math.

4.3.1.6.2.14 Blocks

Blocks are holes that may need to be filled. (Click or Tab to them). Required blocks have a yellow background and optional ones are transparent and have a dotted border.

Click, double-click, highlight, (only right-click inside a text box)

4.3.1.6.2.15 Nuances / Limitations

There are several nuances in the editor, and common ones are listed here, along with workarounds. Also listed are limitations of the editor and things we’d like to get working soon.

- If more than two things are added or summed together, one cannot select only a subset of them.
- One cannot easily change a "+" sign to "*". To do this, you will need to copy the entire "+" operation and paste it, then remove the unwanted children.
- Moving children around by dragging is not possible. Unfortunately, this currently requires copying and pasting to the clipboard.

4.3.1.6.2.15.1 Limitations

Unable to change the domain of operations like Sum, Max, and Integrals.

Operations like Sum, Max and Integrals may be over an interval, or when a certain condition holds (like \( x \in \mathbb{R} \)). The math editor allows editing these variations but does not always offer a way to create new operations. Currently, this must be done by hand by switching to the source edit (View Source, p. 142) view and manually replacing the \(<\text{interval}/>\) with a \(<\text{condition}/>\).
4.3.1.6.2.15.2 Wrapping Math with Math

Sometimes it is necessary to add to existing mathematical operations. For example, adding higher terms to a polynomial. This can be done either by using the keyboard (Keyboard Only, p. 151) or with help of the toolbar (Toolbar, p. 151). In the explanations below we start with "b*x+c=0" and create \( ax^2 + bx + c = 0 \)

**Keyboard Only**

To add the \( ax^2 \) term:

1. Move the cursor to the left of \( bx + c \) but make sure the context (Section 4.3.1.6.2.11: Context) is only around \( bx + c \) and that \( bx + c \) is not selected (Section 4.3.1.6.2.10.1). This can be done by clicking the "+" sign.

2. Enter "a*x^2+" (without the quotes) and press the Enter key.

**Toolbar**

Using only the toolbar to insert math is a bit more difficult because the editor does not infer multiplication or addition when pasting right next to existing math. We will need to "wrap" the existing math with the combiner operation (usually +, *, or ^) and then add in the new math.

1. Select \( bx + c \) but make sure only \( bx + c \) is selected (Section 4.3.1.6.2.10.1). This can be done by double clicking the "+" sign.

2. Cut ( Ctrl+ (Apple [U+2318]+) X Math is selected Cuts the selected Math to the clipboard and replaces it with an empty block (that can be deleted) ) the existing math. This should create an empty block.

3. From the toolbar (Section 4.3.1.6.2.5: Toolbar) select the combiner operation. This should create at least one empty block.

4. Paste ( V Math is selected Pastes MathML from the Clipboard (can be from other sources) ) the math that was cut earlier into one of the empty blocks.

5. Select another empty block.

6. From the toolbar, insert the operation.

4.3.1.6.3 MathML Editor: The Basics

4.3.1.6.3.1 Overview

This module covers some basic points of Connexions’ MathML editor, now in beta testing. It is organized as a tutorial and will lead you through basic usage of the editor, though you can skip around the sections if you wish. The tutorial covers:

- how to access (Section 4.3.1.6.3.2: Accessing the editor) and use the editor,
- how to create (Section 4.3.1.6.3.3: Creating expressions) MathML expressions,

\[^{97}\text{This content is available online at <http://cnx.org/content/m26312/1.2/>}.\]

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
• how to navigate (Section 4.3.1.6.3.4: Navigating through an expression) through the structure of an expression,
• how to edit (Section 4.3.1.6.3.5: Editing expressions) expressions,
• how to insert (Section 4.3.1.6.3.6: Inserting expressions into a module) expressions created in the editor into a module,
• how to delete (Section 4.3.1.6.3.7: Deleting expressions) expressions,
• how to use the menu buttons (Section 4.3.1.6.3.8: Creating expressions with the toolbar menus) to create and edit expressions

You will learn how to create the expression $ax^2 + bx + c$ using keyboard input (Section 4.3.1.6.3.3: Creating expressions) and also using the toolbar menus (Section 4.3.1.6.3.8: Creating expressions with the toolbar menus).

You will also learn about the basic layout (p. 154) of the MathML editor, the context (Section 4.3.1.6.3.4.1: What is going on here?) of your cursor within the editing area, and how to select or highlight (Section 4.3.1.6.3.6.1: Why can’t I select the MathML with my mouse?) part or all of the MathML within the editor with key strokes or the mouse.

All of these aspects of the editor and more are covered in more detail in the MathML Editor: Manual (Section 4.3.1.6.2).

4.3.1.6.3.2 Accessing the editor

Currently, the MathML editor is only supported in Mozilla Firefox\textsuperscript{98}. If you use Internet Explorer or another browser, you can download Firefox for free under the link.

When editing a module using Mozilla’s Firefox browser, click on an element within the module to open a blue editing box (Figure 4.60).

\textsuperscript{98}http://www.mozilla.com/firefox/
Figure 4.60: Begin editing and in the top-right corner is a MathML Editor link

On the top-right hand side of the box is a "MathML Editor" link which will open up the editor (Figure 4.61).
Figure 4.61: Once clicked, a popup window will appear containing the Math Editor.

The popup window (Figure 4.61) can remain open while editing a module and can even stay open while editing several modules. At any point one can close the window, but the contents of the editor will be lost.

The editor has two main sections: the toolbar (Section 4.3.1.6.2.5: Toolbar) and the main editing area (Section 4.3.1.6.2.6: Editing Area).

The toolbar consists of

- a Menu (Menu, p. 142) containing mathematical operators,
- the Path (Path, p. 142) bar to help you find the context of your cursor,
- buttons for Undo/Redo,
- a Preview (Preview, p. 142) button, which will display the math as it will look in a published module,
- and a View Source (View Source, p. 142) button, which displays the raw MathML code.
The editing area is the main area for creating math. It begins empty, but math can be pasted directly in here from Connexions, or created on the spot by either typing in expressions or using the toolbars. See the section Inserting expressions into a module (Section 4.3.1.6.3.6: Inserting expressions into a module) below for how to move math from Connexions to the math editor and back.

4.3.1.6.3.3 Creating expressions

When you first open (Section 4.3.1.6.3.2: Accessing the editor) the MathML editor, the Editing area should have one blank box in it, called a "block (Section 4.3.1.6.2.14: Blocks)". A block is an empty slot that can be filled with a number, variable, or longer expression consisting of at least one operator and its associated arguments.

Click inside of it to place your cursor inside the empty block.

There are two ways to insert math using the keyboard. The simplest way is to use keyboard shorthand notations for operators. These exist for simple algebra notations, and are similar to the buttons on a calculator (for instance, * for multiplication, - for negative or minus). A full list of Keyboard shortcuts is located here (Section 4.3.1.6.2.12: Keyboard Input).

The MathML editor will correctly parse a sequence of text and numbers and keyboard shortcuts into operators and arguments. For instance, 3-4 will be interpreted as "3 minus 4", and 3/(x+9) will be interpreted as "3 divided by the quantity x plus 9".

The second way to create operations using the keyboard is to type out the MathML name. This is analogous to choosing the operation from the toolbar menus and is discussed in the same section (Section 4.3.1.6.3.8: Creating expressions with the toolbar menus).

4.3.1.6.3.3.1 An example

With a blank MathML editor open (Section 4.3.1.6.3.2: Accessing the editor), try typing in the following (or copying it from here and pasting into the editor):

\[ x^2 + bx \]

Hit Enter. The MathML editor will display this as:

\[ x^2 + bx \]

Notice that we had to be explicit about our operations. Although most textbooks, and even our MathML editor, represents the product visually by printing \( b \) and \( x \) next to one another, we had to explicitly mark the multiplication. The editor will display the \textit{times} operation in different ways, depending on the surrounding operations.
If you try to represent the product of $b$ and $x$ with $bx$, that portion of your expression will be highlighted, indicating that there is a problem with the code that must be fixed before being used in a module. Simply click on $bx$ and replace it with $b\times x$.

### 4.3.1.6.3.4 Navigating through an expression

If you have just created (Section 4.3.1.6.3.3: Creating expressions) an expression and pressed Enter, tight now your cursor should be at the right edge of your expression. If it isn’t, click somewhere within your expression, and press the right arrow key until the cursor no longer moves and the path no longer changes in the Path (Path, p. 142) bar. In this position, the Path bar should display Path: / plus and the entire expression should have a light grey background.

![Figure 4.62: The context of the cursor is the entire plus operation.](image)

Press the left arrow key once. The cursor should still be on the right-hand side of the expression, but the Path bar should read: Path: / plus / times and now only the $bx$ term should have a light grey background.

![Figure 4.63: The context of the cursor is the times operation.](image)

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Press the left arrow again. Now the $x$ should appear within a block and be selected. The path will change again to $\text{Path: / plus / times / ci}$. The term $ci$ represents an "identifier", in this case our variable $x$.

**Figure 4.64:** The context of the cursor is the $ci$ block.

Continue pressing the left arrow key across your expression to see what happens. Notice that once you reach the left side of the $b$, the path should again read $\text{Path: / plus / times}$ and the product $bx$ should have a light grey background. When you reach the far left side of the expression, the path should again read $\text{Path: / plus}$ and the entire expression should have a light grey background.

4.3.1.6.3.4.1 What is going on here?

MathML treats operators and their arguments as nested elements. In our example of $x^2 + bx$, $b$ and $x$ are both arguments of the operation called $\text{times}$. The product "$b \times x$" is just one argument within the $\text{plus}$ operation. The other argument of the $\text{plus}$ operation is of course $x^2$. This $\text{power}$ operation itself has two arguments: $x$ and 2. In this way, $b$ is more closely bound to $x$ than it is to either argument of the $\text{power}$ operator.

We could picture these operators and arguments in a parenthetical structure:

\[
( ( (x)^{2} ) + ( (b)*(x) ) )
\]

The parentheses group arguments of each operator together. A set of parentheses encloses each number or variable separately, and a set of parentheses encloses each entire operation - for instance, the entire product $bx$ or the entire sum $x^2 + bx$.

This parenthetical structure can also be visualized as a tree-like structure. See the MathML Editor: Manual (Section 4.3.1.6.2) for an example (Section 4.3.1.6.2.7: Math Tree).

By moving our cursor left and right, we move deeper in or out of these parenthetical or tree-like structures, represented by a longer path name, and a smaller unit of shaded math. The Path and shading lets us know the context (Section 4.3.1.6.2.11: Context) of our
cursor, i.e. where within the parenthetical or tree-like structure our cursor is. Essentially, navigating to different contexts of the MathML expression lets us place new expressions anywhere we want, as we’ll show below (Section 4.3.1.6.3.5.2: Creating a new operation).

4.3.1.6.3.5 Editing expressions

4.3.1.6.3.5.1 Adding an argument to an existing operation

Be sure the editing area of the MathML editor contains the expression we entered (Section 4.3.1.6.3.3: Creating expressions), \( x^2 + bx \). Let’s change this to \( x^2 + bx + c \).

To do this, we will have to add an argument to the addition operation. Move your cursor to the end of the expression by clicking on the last \( x \) and then pushing the right arrow key a couple of times to move the cursor to the right of the \texttt{times} operation. Then type in \( +c \) and press \texttt{Enter}.

You do not always have to add on to the right-hand side of an expression. Try moving your cursor to the beginning of the expression, typing in \( c+ \) followed by \texttt{Enter}. Experiment with adding more arguments to the \texttt{plus} operation just after the \texttt{power} operation or just before the \texttt{times} operation.

4.3.1.6.3.5.2 Creating a new operation

In our example, \( x^2 + bx + c \), let’s change \( x^2 \) to \( ax^2 \). We are not adding an argument to an existing operation, because \( x \) and 2 are arguments of a \texttt{power} operator, while we want to make the entire \texttt{power} operation one argument of a \texttt{times} operation. The other argument will be the new element that we are adding, \( a \).

The underlying structure of our new expression will look something like this:

\[
( ( (a) ( (x)^{(2)} ) ) + ( (b)*(x) ) + (c) )
\]

For this we will need to choose the context (Section 4.3.1.6.2.10: Cursor) of our cursor carefully, or we will get different math. Make sure your cursor is on the left hand side of the \texttt{power} operation, the path says \texttt{Path: / plus / power} and that \( x^2 \) has a light grey background. This means the context of the cursor is the expression \( x^2 \), which is good. We only want to multiply \( x^2 \) by \( a \), nothing more and nothing less.

Then type in \( a* \) and press \texttt{Enter}. The MathML should display as:

\[
ax^2 + bx + c
\]

What if our cursor had been in a different context when we typed in \( a* \)? We could have moved our cursor farther to the left so that the context was the entire \texttt{plus} operation, where the path bar would read \texttt{Path: / plus} and the entire expression would have had
a light grey background. In this case, typing in \(a\) would have affected the entire grey area and would have resulted in this display:

\[ a \left( x^2 + bx + c \right) \]

If our cursor was next to the \(x\) within the \(x^2\) term, a block would appear around the \(x\) and the path bar would read Path: / plus / power / ci. Typing in \(a\) next to the \(x\) in the block would mean that now the quantity \(ax\) would be squared and would result in this display:

\[ (ax)^2 + bx + c \]

### 4.3.1.6.3.6 Inserting expressions into a module

Let’s put our expression \(ax^2 + bx + c\) into a module we’re editing.

The MathML editor won’t automatically insert your MathML expression into a module. You will have to copy and paste the MathML code. You can paste it into any module you wish, not only the module that you first accessed the MathML popup window from. The popup window can remain open even if you edit several different modules, but once you close it all contents of the editor will be lost.

You can press Ctrl+A (or \([U+2318]+A\) on Macs) to select your entire expression. You can also double-click in the white box outside of your expression to select it all. If you want to select only a portion of the expression, hold the Shift key down and use the left and right arrow buttons, or double click on the operator of just the portion you want to select (i.e. double click + to select everything within the plus operator, or double click on two multiplied objects to just select that product).

Copy your expression by pressing Ctrl+C ([U+2318]+C). Open your module and click on a segment to edit. Press Ctrl+V ([U+2318]+V) to paste the MathML code directly into your module.

### 4.3.1.6.3.6.1 Why can’t I select the MathML with my mouse?

There are two ways to select expressions within the editor. By using one of the methods above, you copy the underlying MathML code. This is what you need to copy MathML code over to a module, or to select a portion of your code to delete.

By clicking and dragging over an expression within the edit box, you select only the text. Like any other text you enter, if you copy part of your code and paste it as an argument of an operator within the editing area, the MathML editor will correctly parse it and generate the underlying code. This is also called "wrapping (Section 4.3.1.6.2.15.2: Wrapping Math with Math)" your existing math.
4.3.1.6.3.7 Deleting expressions

There are several ways to delete part or all of your expression. You can either move your cursor to the context (Section 4.3.1.6.2.11: Context) of the portion of your expression you want to delete, then push Delete, or you can select (Section 4.3.1.6.3.6: Inserting expressions into a module) the portion of your code you want to delete, and press Delete.

To delete $bx$ from our expression $ax^2 + bx + c$, move your cursor until the context is only $bx$. The $bx$ portion of the expression should have a light grey background, and the path bar should read Path: / plus / times.

Press Delete once, and that part of your expression should be highlighted in blue.

Press Delete again and that portion will delete.
You should be left with a blank block where the expression $bx$ used to be. By clicking Delete you have deleted the content of one argument of the plus operator, but not the argument itself. You can type in a different expression that will replace $bx$, or if you simply want the expression $ax^2 + c$, press Delete one more time to delete the block. Notice that the context automatically moves to the next argument, which in our case is the $c_i$ element $c$.

### 4.3.1.6.3.8 Creating expressions with the toolbar menus

The toolbar menus are only activated when something is selected within the editing box. This can be either an entire operator, like the expression $bx$, or it could be just one argument of an operator, like $b$ or an empty block. Since the editing box contains only an empty block (Figure 4.61) when you first open it, the block is automatically highlighted and the toolbar menus are accessible.

However, note that if you select a portion of your MathML expression and then insert an operator from the toolbar, the selected portion will be overwritten. The operator will **not** appear behind or in front of the selected expression. New users are often confused by this behavior. Essentially, keyboard input and the toolbar function in different ways.

Remember that each operator in MathML has a specific number of arguments that can be associated with it. For instance, the operator plus must have at least two arguments (as in $x + y$) up to an unlimited number of arguments (as in $1 + 2 + 3 + 4 + 5 + 6...$). The operator root has one required argument (the radicand, as in $\sqrt{3}$), as well as one optional argument for specifying which root power (as in $\sqrt[3]{3}$).

When you insert a series of characters into the editor, the editor will correctly parse some of the symbols you enter as operators, and some as arguments of the operators. Furthermore, it correctly determines **which** operator you have typed in, and correctly associates the surrounding symbols as arguments of that operator.
The menu buttons in the toolbar will paste an operator over your highlighted expression, and insert blank blocks for all the required and optional arguments of the operator. This means that keyboard input and toolbar insertion are useful for different things.

**The toolbar is recommended:**

- When you are just beginning a new expression and already know the relations of the arguments to one another. For instance, if you have a written version of an equation in front of you and want to recreate it in the MathML editor so that you can paste the MathML code into a module. We recreate the equation \( ax^2 + bx + c \) in the sections below (Section 4.3.1.6.3.8.1: Creating a new expression) using only the toolbar, or find a more complicated and in-depth example in the module MathML Editor: Intermediate (Section 4.3.1.6.4).
- When the operator has many arguments associated with it.
- When the operator does not have a keyboard shortcut, or if you do not know the keyboard shortcut. To the right of each menu entry, there is the keyboard shortcut for that operator. For complicated operators, you often must type in the name of the element (for instance, "root"), and the editor will provide a visual representation (the radical sign, in this case) and blank boxes for argument entry.

**Keyboard input is recommended:**

- For most simple MathML entry.
- When you want your existing expression in the editing area to become one argument of a new operation. To add a new argument to the plus operator \( x + y \), you need only type +z. The editor interprets z as one argument of the plus operator, and the previously existing expression as the other argument. Toolbar insertion cannot do this. Some examples of workarounds are given below.

The following sections will show you how to create an expression from scratch using mainly toolbar entry, and how to use part or all of your existing expression as one argument of a new operator.

### 4.3.1.6.3.8.1 Creating a new expression

The toolbars are useful when creating new expressions because you work "from the top down". The operator with the broadest scope is inserted from the menus first. Then, the arguments are filled in with variables and numbers, or with another complex expression. If one complex expression is itself an argument of an operator, you can use the toolbar to insert the operator first, and then fill in the arguments again. Continue this iteration with successively narrower scoped operators, until only number and variables are left to be inserted as arguments.

Let's create \( ax^2 + bx + c \) using the toolbar menus, and in the same order that we created it with keyboard input above (Section 4.3.1.6.3.3: Creating expressions). We'll first insert
\[ x^2 + bx \] (p. 163), then add \( c \) (Section 4.3.1.6.3.8.2: Adding an argument to an existing operation) as an argument to the existing \texttt{plus} operator, and finally add \( a \) (Section 4.3.1.6.3.8.3: Creating a new operation) as an argument in a new operation.

Make sure the editing area in the MathML editor is blank. It should already be blank if you’ve just opened it. If you have some text entered, delete (Section 4.3.1.6.3.7: Deleting expressions) it first.

The expressions \( x^2 \) and \( bx \) stand in relation to one another as arguments of a \texttt{plus} operator, so we will start by inserting a \texttt{plus} operator and then inserting successively smaller chunks of the expression.

Step 1. Click the \texttt{plus} operator from the toolbar.
Step 2. To create the \( x^2 \) expression, click within the first block in the \texttt{plus} operation, then select the \texttt{power} operator from the toolbar.
Step 3. Click within the first block in the \texttt{power} operation, and type in \( x \).
Step 4. Click within the second block in the \texttt{power} operation, and type in \( 2 \).
Step 5. To create the \( bx \) expression, click within the remaining block, which should be the second argument of the \texttt{plus} operation, and select the \texttt{times} operator from the toolbar.
Step 6. Click within the first block of the \texttt{times} operation, and type in \( b \).
Step 7. Click within the second block of the \texttt{times} operation, and type in \( x \).

### 4.3.1.6.3.8.2 Adding an argument to an existing operation

There are two ways we can add the final argument \( c \) to the \texttt{plus} operator. One way would have been to add an extra argument to the \texttt{plus} operation immediately after we inserted it from the toolbar menu. After Step 1 above, click in either blank block on each side of the + sign. Then insert another \texttt{plus} operation. Essentially we are inserting the entire operator as an argument of the first \texttt{plus} operator, but the editor will correctly condense this and turn all three blocks into arguments of one \texttt{plus} operator. The same thing occurs with keyboard input when, say, you replace the 6 in \( 6 + 7 \) with \( 5 + 6 \).

The second method would be to want to add a new argument to the \texttt{plus} operator after we already created the expression \( x^2 + bx \). In this case, we essentially want to make \( x^2 + bx \) the first argument of a new \texttt{plus} operator, and then add \( c \) as the second operator. This is known as ”wrapping (Section 4.3.1.6.2.15.2: Wrapping Math with Math)” existing math in a new operation.

Step 1. Select the entire expression and cut it by pressing Ctrl+X (or \[U+2318\]+X on a Mac), so that you are left with a blank editing area.
Step 2. Select the \texttt{plus} operator from the toolbar menu.
Step 3. Paste in the expression \( x^2 + bx \) as the first argument of the operation.
Step 4. Click in the remaining block and type in \( c \).

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.1.6.3.8.3 Creating a new operation

You will also have to wrap existing math when inserting a new operation. Let’s change the first expression $x^2$ to $ax^2$.

Step 1. Select and cut the power operation $x^2$. You should be left with a blank block $+ bx + c$.
Step 2. Insert a times operator from the toolbar menu.
Step 3. Click on the first block of the times operator and type in $a$.
Step 4. Click on the second block of the times operator and paste in the expression that you cut.

4.3.1.6.4 MathML Editor: Tutorials

4.3.1.6.4.1 Introduction

This module presents step-by-step instructions for creating, and then editing well-known formulas to illustrate how to use the editor. The example and what properties of the editor it illustrates are listed below. All of these assume you have a blank Math Editor open.

- Quadratic Equation (Section 4.3.1.6.4.2: Quadratic Equation) (Simple text entry, Special characters)
- Quadratic Equation using mostly the mouse (Section 4.3.1.6.4.2.2: Method 2: Toolbar)
- Advanced Text Entry (Section 4.3.1.6.4.2.3: Advanced Editing)

4.3.1.6.4.2 Quadratic Equation

As a simple example, we’ll step through several ways of writing the well-known Quadratic Equation (with real or complex coefficients):

$$ax^2 + bx + c = a \left( x - \frac{-b + \sqrt{b^2 - 4ac}}{2a} \right) \left( x - \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right)$$

4.3.1.6.4.2.1 Method 1: Pure Keyboard

Probably the quickest way to enter math is by using the keyboard. This method requires entering a total of 3 statements and a few Tab key presses.

Step 1. Start off with a blank editor.

---

99This content is available online at <http://cnx.org/content/m24560/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Step 2. Enter the following into the main editing area
"a*x^2+b*x+c=a*(x-(-b+\sqrt{b^2-4*a*c})/(2*a))*(x-(-b+\sqrt{b^2-4*a*c})/(2*a)) ". See below for details (p. 165).

Step 3. Press the Enter key. This will cause the text to be parsed and converted into math.

Most of the text in step 2 should look similar to the notation used in calculators, except for \"root\". Many calculators follow different conventions for entering complicated math operations like integrals and vectors. For this version of the editor we decided to wait for feedback from users on which convention to adopt. Until one is chosen, any math element defined in the W3C MathML Specification can be entered. The toolbar also provides a way to see the available commands.

**Finish Entering Equation**

At this point the editor should have 2 remaining boxes that need to be filled out, and 2 optional ones (the degree of the root). To fill in the rest, you will need to do the following:

1. Press Shift+Tab four times to move to the first empty block. (That is, hold down the Shift key, press the Tab key, and release the Shift key four times).
2. Enter \"b^2-4*a*c \" into the empty block under the radical.
3. Press the Enter key to convert the input into math.
4. Press Shift+Left arrow key to select \(b^2 - 4ac\)
5. Press Ctrl+C to copy the selection to the clipboard.
6. Press the Tab key twice to move to the other empty block.
7. Press Ctrl+V to paste the selection into the current block.

Now, the equation should be complete. In the previous steps we used the Tab key to navigate to empty blocks that still needed information in them, skipping over optional ones. We used Shift+ arrow keys to select math and Ctrl+C and Ctrl+V to copy and paste that math.

**Paste into Connexions**

Finally, we need to copy the math and paste it back into a module. We already used the same technique above. Right now, the cursor should be just to the right of the second \(b^2 - 4ac\). The following steps will place the newly created quadratic equation back into the Connexions module.

Step 1. Press Ctrl+A to select the entire formula, or Shift+Right (or Shift+Left) until the math you want to copy is selected.
Step 2. Press Ctrl+C to copy it to the clipboard.
Step 3. Switch back to the window where you were editing the module.
Step 4. Place the cursor at the location you want to insert the quadratic formula.
Step 5. Press Ctrl+V to insert the formula.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Summary
In this tutorial we entered the quadratic equation entirely through the keyboard. We used the Tab and arrow keys to navigate through math content, the Return key to convert input text into math, and Ctrl+C and Ctrl+V to copy and paste both within the editor and between the editor and the main module editor.

Next, we will do the same example using the mouse and toolbar.

4.3.1.6.4.2.2 Method 2: Toolbar
This method requires a bit more time because we will need to click the toolbar for every character (like "+", "*", or "/" in the previous method). Instead of doing the entire equation, this tutorial will step through creating only part of it: \[ \frac{-b + \sqrt{b^2 - 4ac}}{2a} \]

Step 1. Start off with a blank editor.
Step 2. Click the "Arithmetic" category in the toolbar.
Step 3. Click the "Divide" operation in the menu.
Step 4. Click the top empty block (numerator).
Step 5. Click the "Arithmetic" category in the toolbar and the "Plus" operation in the menu.
Step 6. Click the left empty block.
Step 7. Click the "Arithmetic" category and the "Negate" operation in the menu.
Step 8. Enter "b" in the top-left empty block.
Step 9. Click the other empty block in the numerator.
Step 10. Click the "Arithmetic" category and the "Root" operation in the menu.
Step 11. Click the empty block under the radical.
Step 12. Click the "Arithmetic" category and the "Minus" operation in the menu.
Step 13. Click the left empty block.
Step 14. Click the "Arithmetic" category and the "Power" operation in the menu.
Step 15. Enter "b" and "2".
Step 16. ...

Using the toolbar is a bit more tedious, but serves as a way to find operations that can be expressed in MathML. Some operations have variations (A sum can take a variable and limits, or a variable and a condition) but see Limitations on how to enter them in.

Paste into Connexions
Pasting the Math back into Connexions can be done the same way as before, or can be done via the Edit menu in the browser. Again, we must select (Section 4.3.1.6.2.10.1) the entire equation. This can be done by highlighting the equation using the mouse, or double-clicking the division bar (since it is the outer-most operation). Once highlighted, you can Click Edit, and either Cut or Copy from the main browser menu bar. If you switch back to the Connexions module editor, you can Click Edit and then Paste again from the menu to paste the newly created math back into a module.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Summary
In this tutorial we entered a part of the quadratic equation using the mouse and toolbar
buttons. We used the mouse to select move the cursor and select math, the toolbar to
insert new operations, and the browser’s Edit menu to copy and paste math between the
editor and the main module editor.

Next, we will discuss some more advanced math editing.

4.3.1.6.4.2.3 Advanced Editing

So far we’ve gone through creating math from scratch. In this section, we will look at how
to insert more elaborate symbols, change how variables look (Presentation MathML), and
customize some of the operations provided in the toolbar.

Elaborate Symbols
So far we’ve used simple characters available from the keyboard. The quadratic formula
is frequently written with a plus-minus sign like:

\[-b \pm \sqrt{b^2 - 4ac} \]

\[\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]

In order to get this, we will need a little bit of Presentation MathML. This is because
plus-minus is not an operation represented in Content MathML.

Step 1. We will start with one part of the formula (as described above (Section 4.3.1.6.4.2.1:
Method 1: Pure Keyboard) )

Step 2. To save some time, copy the \[\sqrt{b^2 - 4ac} \] into the clipboard.

Step 3. Highlight everything in the numerator (it should be the entire plus operation)

Step 4. Replace it with "mrow " and press the enter key

Step 5. Enter "-b " and move to the next block

Step 6. Enter "± " and move to the next block

Step 7. Paste the part of the formula we copied earlier

mrow is used to control how Math is displayed to the user. In this case we used it to
insert a plus-minus symbol between \(-b \) and \[\sqrt{b^2 - 4ac} \]. The Unicode standard defines
many characters but the Unicode Mathematical Operators\(^{100}\) document may be a useful
reference.

Customize the Look of Variables
There are many elaborate ways to customize how a variable looks. These are defined in
the W3C MathML Specification (Section 4.3.1.6.4.3: W3C MathML Specification) . We will
list off a few common ways to customize.

• Subscripts like \(x_i\) can be entered by typing "x_i " or using "msub "

\(^{100}\)http://www.unicode.org/charts/PDF/U2200.pdf

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
• A variable with both subscripts and superscripts can be entered using "\texttt{msubsup}".
• Backets like \((−∞, 0]\) can be added using "\texttt{mfenced}" and then changing the symbol used for the open and close bracket (by editing the source).
• Unlike subscripts which place content above/below and to the right, "\texttt{munderover}" places math directly above or below.
• A table can be added using "\texttt{mtable}".

**Customize Toolbar Operations**

Many operations that operate on a range have several ways of specifying the range they work on. For example, the following are equivalent:

\[
\sum_{i=1}^{n} i^2
\]

\[
\sum_{i\in S} i^2 \quad S = \{i \mid (i > 0) \text{ and } (i \leq n)\}
\]

Changing the range these operations required switching to the MathML source and being familiar with the W3C MathML Specification (Section 4.3.1.6.4.3: W3C MathML Specification). To change the former to the latter, we start with a clean "\texttt{sum}" operation. Then, to decrease the amount of hand editing, we can type "\texttt{i in S}" to the right of the equal sign. Then, we switch to MathML Source and replace every occurrence of "\texttt{interval}" with "\texttt{condition}" and removing the special "\texttt{<block ...>}" element just above the "\texttt{</condition>}".

### 4.3.1.6.4.3 W3C MathML Specification

The W3C MathML Specification\(^{101}\) defines all math used in Connexions modules. It provides ways to represent formulas in a way that records the semantic meaning in the formula (Content MathML\(^{102}\)) as well as a way to lay out variables and formulas (Presentation MathML\(^{103}\)).

### 4.3.2 Files

#### 4.3.2.1 Module Files\(^{104}\)

It is possible to create a module (Section 4.1) with just textual information. In this case, the content (Section 4.3.1.1) is all included in the CNXML file of the module. But in some cases you may want to include associated files with your module content, like an image, video, PDF, PPT, or other file.

\(^{101}\)http://www.w3.org/TR/MathML2/

\(^{102}\)http://www.w3.org/TR/MathML2/chapter4.html

\(^{103}\)http://www.w3.org/TR/MathML2/chapter3.html

\(^{104}\)This content is available online at <http://cnx.org/content/m38190/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
You can upload associated files on the "Files" tab at the top of the module edit page. In this case, the module is really more like a "box", which can contain any number of files associated with the module in addition to the CNXML file itself. When you upload files, you create an instance of the file in Connexions that you can then link to (Section 4.3.2.1.2: Linking to files) from your module for readers to download, or even embed (Section 4.3.1.2) directly into your module.

This module will explain how to:

- Upload files (Section 4.3.2.1.1: Uploading Files) to your module
- Link to files (Section 4.3.2.1.2: Linking to files) in your module
- Manage files (Section 4.3.2.1.3: Managing Files) on the "Files" tab

4.3.2.1.1 Uploading Files

4.3.2.1.1.1 File Guidelines

Before you add an associated file to a module, please note the following guidelines for files and filenames:

- The size of a file must be smaller than 50 MB.
- The first character in a filename cannot be a digit. For example, a file named "1965mustangphoto.gif" would result in an error. Renaming (Section 4.3.2.1.3.2: Renaming Files) the file to "photo1965mustang.gif" prevents the error.
- A filename cannot contain an embedded blank space. For example, the file named "eiffel tower.png" would cause an error. Renaming (Section 4.3.2.1.3.2: Renaming Files) the file to "eiffel_tower.png" or "eiffeltower.png" prevents the error.
- Use only alphanumeric, "." (dot), "-" (hyphen), and "_" (underscore) characters in a filename.
- Multimedia files (Section 4.3.1.2) must have a recognized MIME (Multipurpose Internet Mail Extension) type.

4.3.2.1.1.2 Uploading a file to a module

To add a file to a module, use the following steps:

1. Click on the "Files" tab (Figure 4.68). This page shows a list of all included files in the module. The index.cnxml file exists in every module – this is the CNXML file that creates the module contents which you edit on the Edit tab.
2. Click the Add new image/included file button to add a file.
3. Click **Browse** to navigate to the file in your system that you want to upload.
4. Click the blue **Upload** button to upload the file into Connexions. A confirmation message will display, along with a preview if the file is an image.

5. Click the **Return to file listing for module** link at the top of the page to return to a list of all files uploaded to this module. Here you can click the **Add new image/included file** button and start the process over to upload another file.

### 4.3.2.1.1.3 Uploading multiple files at once

If you have a lot of files that you want to include in a module, uploading them one by one can take a while. You can use the ZIP importer (Section 4.2.1.1.1.4: ZIP Importer) to import multiple files at once.

![Image not finished](http://cnx.org/content/col10151/1.27)
4.3.2.1.2 Linking to files

Readers can download a file that you added in two ways.

1. By adding the file to the module’s Featured Links (Section 4.3.5.1). This is probably the easiest way, but although the links appear online, they will not be included in the PDFs that are automatically generated.
2. You can add a link to the file within the module text. If the file you added was called "myfile.doc", you would add the following snippet to your module. "Download this <link resource="myfile.doc">file.</link>" More instructions on adding links in modules can be found on the <link> eip-help page105 .

NOTE: You can also embed (Section 4.3.1.2) the file into the CNXML document using the CNXML<media> tags.

4.3.2.1.3 Managing Files

WARNING: Do not remove or rename the index.cnxml file. This file holds the text and CNXML-markup of your module.

4.3.2.1.3.1 Removing Files

To remove a file from a module, select the checkbox next to the title of the file you want to remove and click Remove. You can select multiple files in the module and remove them in one operation.

4.3.2.1.3.2 Renaming Files

To rename a file in a module, use the following steps:

1. Select the checkbox next to the title of the file you want to rename and click Rename. The "Rename items" screen displays.
2. Enter the new name for the file in the "New File Name" field.
3. Click Rename All. You can rename multiple files at one time by selecting more than one item by clicking on the work area contents screen.
4. Update any existing links in your module to point to the new file name.

The files displayed on the "Files" tab are updated to reflect the changes.

105http://cnx.org/eip-help/link

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.2.1.3.3 Cutting or Copying Files

To copy and paste files in one module to a different module, use the following steps:

1. Select the checkbox next to the title of the files you want to cut or copy and click Cut or Copy. Cut removes the original item at the end of this process. Copy leaves the original item in its present location.
2. Display the location where you wish to paste the file, such as another module or work area, and click Paste.

4.3.3 Metadata

4.3.3.1 Module Metadata

You already provided the basic metadata (p. 50) information about the module when you created it. If you need to change it before publishing, you can return to the "Metadata" tab to make updates. Upon returning, you might notice that there is a new, more advanced field:

- If you use Google Analytics, you can enter a code to track usage of the module.

4.3.3.1.1 Google Analytics

One powerful option for keeping track of how your content is used is through the use of Google Analytics\(^\text{107}\). This service tracks how many people visit your content, where they are located, and how much time they spend on each page, to name just a few metrics.

If you already have a Google Analytics account, you can use your tracking number to keep up with how your content is being viewed. To add your tracking number to a module or collection, go to the metadata tab and find the Google Analytics field:

\[\text{Figure 4.71: Google Analytics on the metadata tab.}\]

Copy the tracking code you want to use into the field and click “Save”. Note that for collections, the Google Analytics code is linked to your collection home page and will only update when the home page is visited. It will not update if someone visits a module.

\(^{106}\)This content is available online at <http://cnx.org/content/m38204/1.3/>.

\(^{107}\)http://www.google.com/analytics/index.html

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
within your collection. To keep track of this information, you will need to insert your tracking code into the metadata section for each individual module.

If you do not already have a Google Analytics account but would like to take advantage of this feature, you can sign up for an account at the Google Analytics site. For help on setting up your account, see the Google Analytics help page.

4.3.4 Roles

4.3.4.1 Roles

As creator of a module or collection, by default you hold all three required roles: author, maintainer, and copyright holder. Note that of the following roles, only someone assigned the maintainer role may (re-)publish the content.

- **Authors** receive credit for writing the material. The author is the person who is responsible for the creation of the content, but may or may not be the person who actually enters the work into the Connexions. Any module or collection must have at least one person with an author role before it can be published in Connexions.
- **Maintainers** are able to edit and publish successive versions of a module or collection, even if they were not responsible for the creation of the content itself. Most authors will serve as maintainers for their materials, but some may choose not to. Authors may appoint additional maintainers to assist them in updating the materials.
- **Copyright Holders** have the legal right to license the materials. In most cases, the copyright holder is the same person as the author, but this may not be the case in institutions where faculty and employees are required to sign over copyright to the institution. In this case the institution serves as copyright holder and the individual retains the author and/or maintainer roles.

Additionally, there are two other roles that Connexions offers for special content types. These roles are not granted permission to publish content.

- **Editors** receive credit for editing a module or collection. Editing duties may include checking the material for accuracy, grammar, spelling, organization, and so on. Unlike authors, editors do not take credit for the ideas behind the content, but rather for its organization and presentation. Editors do not have rights to publish materials in Connexions, so editors who are also responsible for keeping modules or collections up-to-date must be assigned the maintainer role in order to publish new versions of the content. Editors are listed under the "More about this content" link that appears at the end of the module.

---

108 http://www.google.com/analytics/index.html
109 http://www.google.com/analytics/support.html
110 This content is available online at <http://cnx.org/content/m38177/1.3/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
• Translators have taken existing Connexions content and translated it from one language into another. Translators are listed next to the authors’ names in the content headers. As with editors, translators do not have permission to publish or update content, so translators with publication responsibilities should also be assigned as maintainers for the content.

You can grant one or more roles to other Connexions account holders who you want to collaborate with you on the module or collection. The "Roles" tab has a search feature for locating these people. The current collaborators on the content are listed by role under the heading "Current Roles" on the "Roles" (Figure 4.72) tab. You can change the roles assigned to these people by selecting or deselecting the check boxes in the "Authors", "Maintainers", "Copyright Holders", "Editors" and "Translators" columns under the heading "Change Roles".

![Figure 4.72: Example "Roles" tab for a module.](image)

When you grant or update roles for a collaborator on a module or collection in a work area, a role request is sent to the collaborator whose roles have been modified; the collaborator must either accept or reject the role request. If there are any pending roles (roles not yet accepted or rejected) for a module or collection, Connexions will not let you publish that content. Once the collaborator accepts the role or updated role, you must (re-)publish the content for the updated roles to appear on the published version of the content. The following paragraphs describe how to grant or update roles.

4.3.4.1.1 Adding and Removing Role Assignments

**Granting Roles to Collaborators**

To grant one or more roles in a module or collection to another Connexions account holder, use the following steps:

1. Click the "Roles" tab to display it.
2. If the collaborator is a member of your current workgroup, click the "Select from members of your current workgroup" link. A table that lists all the members of the current workgroup appears under the link.
3. If not, type the first or last name of the person you want to grant a role to in the text box next to Search.
4. Click Search. The Connexions accounts are searched and any Connexions accounts that match the search string are displayed.

5. Select the check boxes for the appropriate roles next to the name of the person you want to add and click Add People. The name of the new collaborator appears with the names of authors, maintainers, and copyright holders under the "Current Roles" heading and in the table under the "Change Roles" heading. In both places, the new collaborator is noted as a role change that is pending acceptance.

6. Look at the table under the "Change Roles" heading and verify that the check boxes in the "Author", "Maintainer", "Copyright Holder", "Editor", and "Translator" columns are checked for the roles you want to grant to the new collaborator. If necessary, you can click the check boxes to change which roles will be granted.

7. Click Submit Changes. The next time the new collaborator logs into Connexions, he or she will be notified of a pending role request (Section 4.3.4.1.2: Viewing and Processing Pending Role Requests). Connexions will not apply the roles to the new collaborator until he or she accepts the role request.

**Changing Role Assignments**

To change any roles that were previously granted to a user, return to the "Roles" tab, check or uncheck the appropriate check boxes to reassign roles, and click Submit Changes. Each time to change the role assignments, a notification of a pending role change request (Section 4.3.4.1.2: Viewing and Processing Pending Role Requests) will display to the collaborator upon his or her next login to Connexions. Connexions will not apply the roles to the new collaborator until he or she accepts the role request.

**Changing the Order of Author Names**

To have an author’s name appear first on the module or collection, his or her name must be at the top of the "Change Roles" table. The second name from the top of the table will appear as the second name in the list of authors, and so on. Click the up or down arrows in the "Order" column to move an author up or down in the order of appearance on the published content. The single arrows move the name up or down one place. The double arrows move the name to the top or bottom of the list. Click Submit Changes to save the order changes.

**Removing a Collaborator**

To remove a collaborator from a module or collection, use the following steps:

1. Click the "Roles" tab to display it.
2. Locate the name of the person or persons in the "Change Roles" table of which you want to remove.
3. Click on the check box in the "Remove User" column for the collaborator you want to remove.
4. Click Submit Changes. The "Change Roles" table is updated with the removed collaborator not appearing on the list of users with roles.
A notification of a pending role request (Section 4.3.4.1.2: Viewing and Processing Pending Role Requests) will display to the collaborator upon his or her next login to Connexions. Connexions will not remove the collaborator until he or she accepts the role removal request.

### 4.3.4.1.2 Viewing and Processing Pending Role Requests

Connexions makes it easy to collaborate on educational materials. One way to do this is to share authorship or maintenance of a module or collection. If another author has granted you a role in their module or collection so that you can collaborate with them, the "Pending" (Figure 4.73) sidebar will appear in the upper right corner of your Connexions pages.

![Image not finished](Image not finished)

**Figure 4.73:** "Pending" sidebar with role requests.

Click the **Role Request** link to display the "Collaboration Requests" (Figure 4.74) screen. From this screen you can view and act upon:

- Requests awaiting your approval (Section 4.3.4.1.2.1: Requests Awaiting Your Approval)
- Requests you have made awaiting others' acceptance (Section 4.3.4.1.2.2: Requests You Have Made)
- Completed requests (Section 4.3.4.1.2.3: Completed Requests)

![Image not finished](Image not finished)

**Figure 4.74:** "Collaboration Requests" screen showing one request awaiting your approval.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.3.4.1.2.1 Requests Awaiting Your Approval

If another Connexions author grants you a role on their module or collection as an author, maintainer, copyright holder, editor, and/or translator, an entry appears under the ‘Requests Awaiting Your Approval’ section of the “Collaboration Requests” screen. To accept the request select the check box in the table for the request you wish to accept and click Accept.

To reject the request, select the check box in the table for the request you wish to reject, and click Reject.

NOTE: If you have been requested to be added to a module or collection as a copyright holder, you must select the check box indicating that you agree to the license which has been selected for that module before you can accept the request. A link to the full license is provided in the "License" column of the table.

4.3.4.1.2.2 Requests You Have Made

If you have requested that someone be added to one of your modules or collections as an author, maintainer, and/or copyright holder, an entry appears under the "Requests You Have Made" section of the "Collaboration Requests" screen until the other author acts on it. You can cancel pending requests from here.

4.3.4.1.2.3 Completed Requests

Any requests you have made that were accepted or rejected are listed in the "Requests You Have Made – Completed" section of the "Collaboration Requests" screen. When you no longer need to keep the record, you can delete it.

4.3.5 Links

4.3.5.1 Module Links

To add or edit links that are associated with a module, click on the "Links" tab on the editing interface. The "Edit Featured Links" (Figure 4.75) screen displays.

Figure 4.75: The "Links" tab.

---

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 4. AUTHORING CONTENT

From the "Links" tab you can:

- Add new featured links (Section 4.3.5.1.1: Adding New Featured Links)
- Edit existing featured links (Section 4.3.5.1.2: Editing Existing Featured Links)
- Delete a featured link (Section 4.3.5.1.3: Deleting a Featured Link)

These links are viewable in the "Links" box that displays with each module (online view only). The featured links appear under their specified category type, for example, the supplemental links are listed under the "Supplemental" heading.

![Image not finished](http://cnx.org/content/col10151/1.27)

**Figure 4.76:** Example of a Links box displayed on a module.

### 4.3.5.1.1 Adding New Featured Links

To add a new link to a module, use the following steps:

1. Type the label for the link in the "Title" field (required).
2. Click on the down arrow next to the "Type" field and select the type of link from the drop-down list (prerequisite, supplemental, or example).
3. Click on the down arrow next to the "Strength" field and select a number from the drop-down list for the strength of the link. 5 is the strongest link. Stronger links will display with more bars next to the link title in the module’s "Links" box.
4. Identify the target of the link by typing in a module ID in the "Module ID" field, or typing a URL. You can optionally specify a version if you entered a module ID. You can provide the name of any files (Section 4.3.2.1) that you added to the module and then readers can download that file using the featured link. Be sure and include just the file name and not the full URL to it, if the file is part of your module. For example: "myword.doc".
5. Click **Add** to add the link.

### 4.3.5.1.2 Editing Existing Featured Links

If the module already contains links, they will appear in the top half of the screen. To edit an existing link, use the following steps:

1. Update the contents of the "Title" field to change the label of the link.
2. Click on the down arrow next to the "Type" field and select another type of link from the drop-down list.
3. Click on the down arrow next to the "Strength" field and select another number from
the drop-down list for the strength of the link. 5 is the strongest link.
4. Update the contents of the "URL" field to change the URL of the link.
5. Click Update Links to make your changes.

4.3.5.1.3 Deleting a Featured Link

To delete an existing link, select the checkbox in the "Delete" column in the list of existing
links. Then click Update Links to delete the link.

4.3.6 Publish

4.3.6.1 Preview Module

After editing the module, you should preview the module to verify that it appears as you
want it to appear. You can view modules from the "Preview" tab in the following ways:

- Preview the module on-line (Section 4.3.6.1.1: Previewing Modules (Online))
- Preview the module in a print version (PDF file) (Section 4.3.6.1.2: Previewing Mod-
  ules (PDF))
- View the change summary (Section 4.3.6.1.3: Viewing Change Summary)

Note that these three options are also available in the Module Status (Figure 4.77) sidebar.
The sidebar also has a few extra options that are not available on the "Preview" tab (Source
and Discard) which we discuss below (Section 4.3.6.1.4: Previewing the code source).

**Image not finished**

Figure 4.77: The Module Status portlet.

4.3.6.1.1 Previewing Modules (Online)

To view a module online, use the following steps:

1. Click on the "Preview" tab. The "Preview" screen displays.
2. Click the **Online Preview** link. This displays an on-line version of the module that is similar to the on-line display of the published module. The difference between the preview and the published versions is that the preview cannot display certain porlets (like the Recently Viewed (Section 3.1.4: Recently Viewed) portlet). Review the module content and verify that it appears as you intended it to display.

3. Use your browser’s **Back** function to return to the "Preview" screen.

### 4.3.6.1.2 Previewing Modules (PDF)

The print version of a module is formatted differently than the online version. This difference is necessary because of the differences in the online and printed media. To view a print version of the module, use the following steps:

1. Click on the "Preview" tab. The "Preview" screen displays.
2. Click the **Print Preview** link. An "Opening index.pdf" dialog box appears.
3. Verify that the option to open the file with a PDF viewing application is selected and click **OK**. The print version of the module displays.
4. Review the module display and verify that text and figures display as you intended them to display.
5. You can print this PDF file using your PDF viewer’s file/print option.
6. Close the PDF viewing application.

### 4.3.6.1.3 Viewing Change Summary

When editing an existing module, you can view a summary of the differences between the last published version of the module and the version you are currently editing. Click the **Changes** link to view the change summary (Figure 4.78). This option is useful when you checked out a module some time ago and do not remember what changes you made to it.

---

**Figure 4.78:** The "Changes to module" summary.

---

The "Changes to module" summary lists changes to the module, including role assignments and the names of any files that were changed, files that were added, and files that were removed. Click on a file name to display a brief description of the changes.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Clicking on the index.cnxml link will display the CNXML markup (Figure 4.79) of the module. Lines of markup which have been removed or the "before" version of altered lines are shown with a minus sign in the left margin. Lines which have been added or the "after" version of altered lines are shown with a plus sign in the left margin.

Figure 4.79: An example of the "diff" log, showing the changes in index.cnxml.

4.3.6.1.4 Previewing the code source

Occasionally it may be convenient to view the markup source for a module. To view the source of the module, click the Source link under the "View" heading in the "Module Status" sidebar. The most recent markup source that has been saved displays with line numbers.

NOTE: This feature is useful when you are trying to find a markup error.

4.3.6.1.5 Discarding Your Changes

The situation may arise in which you wish to return a module that you are editing back to its original state: the state in which it was before you began editing it. To do this, click the Discard link under the "Actions" heading in the "Module Status" sidebar.

Image not finished

Figure 4.80

Discard: The module will revert back to the state it was in when you checked it out. If the module is new and has never been published, you will lose all the content you entered and the module will be blank.

Delete: This button removes the module from your current work area but does not remove it from the repository. Once published, a module remains in the Connexions repository.

Cancel: This button will return you to the screen you just left.
4.3.6.2 Publish

To make any unpublished changes to an existing module or collection viewable by anyone on the Internet, you must publish the content in Connexions. To publish content, use the following steps:

1. Click on the "Publish" tab. The "Publish Module" (Figure 4.81) (or "Publish Collection") screen displays. This screen contains a list of reminders for you to check before you publish the content, as well as a reminder of the terms of the license agreement.
2. Type a brief description of your new changes to the content in the "Description of Changes" text box.
3. Click Publish. Your changes to the content are entered into the Connexions Content Repository as a new version of the existing content (version 1.1 if the content is being published for the first time). Once in the Content Repository, anyone with Internet access can view the new version module or collection.

A new module or collection would not necessarily have editing changes, so in the "Description of Changes" text box you might enter a comment like "Created a new module" or "Created a new collection".

NOTE: Once published, a module or collection may not be "unpublished" or removed from the Content Repository. Therefore, use the Preview (Section 4.3.6.1) feature to preview what the module or collection will look like once published; only publish when you are sure you have a draft that you don’t mind making public.

This content is available online at <http://cnx.org/content/m13054/1.3/>.
4.4 Build Collection

4.4.1 Create new collection

4.4.1.1 What is a Collection?

A collection builds an ordered sequence of modules (Section 4.1). If you think of modules as individual building blocks, then collections are the things you can build out of those blocks - in this case textbooks, courses, lab manuals, journal issues, and so on. When viewed online, each module will be displayed as a separate web page with links allowing you to move between pages sequentially or jump to a specific point using the table of contents. In PDF format, modules in a collection are combined to form a single document that can be read as a book.

As a collection author, you are free to use any combination of published modules from the content repository, including content created by others. The collection composer allows you to arrange those modules in any order, create "chapters" or other hierarchical structures, and rename modules in order to give you complete control over the organization of the content. Once published, readers can access your collection as a free online ebook (available through the website), download a free PDF copy of your collection for printing or sharing, or even order a low-cost printed version of the text through our print-on-demand partner.

4.4.1.2 Creating Collections

Let’s imagine you want to make a collection with the following structure of its contents:

![Image not finished]

**Figure 4.82:** This collection contains an introduction, and two parts, each containing two modules.

Creating a new collection is similar to creating a new module (Section 4.1). You can create a new link from the "Create New Item" dropdown in one of your work areas (Section 3.3.1), or you can check the "Create a new collection" link on the MyCNX home page:

---

114 This content is available online at <http://cnx.org/content/m38767/1.2/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
From there you must agree to the license for this collection, select a work area in which to edit the content, and then enter the collection’s initial metadata.

After accepting the license, you will be asked to enter some basic information, or metadata, about your collection before getting started. (If you haven’t already selected a work area in which to work on the collection, you will be asked to do so.) The basic metadata include:

- A title for the collection
- The primary language used in the collection
- A subtype that reflects the intended use of the collection, such as Course or Report
- The subject categories that apply to the collection
- Terms that describe the most important topics in the collection
- A brief description of the collection. You may markup the summary section with inline CNXML tags.

Click the Next button. The metadata entries are saved and the "Contents" tab displays with the title of the collection. This tab is where you do the actual assembly of the collection.

4.4.1.3 Editing Collections

Once you have opened a new or adapted collection within one of your Work Areas, you will need to edit the collection to add modules and other collection-wide parameters. The recommended sequence to perform these operations in is described in this module and appears in the following list:

1. Arrange (Section 4.4.3) the collection contents. Edit how the title and links of the modules will appear when viewed within the context of the collection.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2. Edit the metadata (Section 4.4.2) to enter the appropriate search keywords and the collection title.
3. Edit the collection roles (Section 4.3.4.1). This sets the roles for the collection itself, but does not change the roles of the modules included in the collection.
4. Set the collection parameters (Section 4.4.4).
5. Preview (Section 4.4.5) the collection online.
6. Publish (Section 4.3.6.2) the collection.

4.4.2 Collection Metadata\textsuperscript{115}

You already provided the basic metadata information about the collection when you created (Section 4.4.1) it. If you need to change it before publishing, you can return to the "Metadata" tab to make updates. Upon returning, you might notice that there are some new, more advanced fields:

- If you use Google Analytics, you can enter a code to track usage of the collection home page. Modules within the collection must be individually tracked.
- The URL for an external web site related to the collection.
- A university or other institution affiliated with the collection.
- A formal designation for the collection if it’s of a course subtype.
- Names of the instructors if the collection subtype is a course.

\textbf{Image not finished}

\textit{Figure 4.85}: Advanced metadata fields for a collection.

4.4.2.1 Google Analytics

One powerful option for keeping track of how your content is used is through the use of Google Analytics\textsuperscript{116}. This service tracks how many people visit your content, where they are located, and how much time they spend on each page, to name just a few metrics.

If you already have a Google Analytics account, you can use your tracking number to keep up with how your content is being viewed. To add your tracking number to a module or collection, go to the metadata tab and find the Google Analytics field:

\begin{footnotesize}
\textsuperscript{115}This content is available online at <http://cnx.org/content/m38768/1.1/>.
\textsuperscript{116}http://www.google.com/analytics/index.html
\end{footnotesize}
Copy the tracking code you want to use into the field and click “Save”. Note that for collections, the Google Analytics code is linked to your collection home page and will only update when the home page is visited. It will not update if someone visits a module within your collection. To keep track of this information, you will need to insert your tracking code into the metadata section for each individual module.

If you do not already have a Google Analytics account but would like to take advantage of this feature, you can sign up for an account at the Google Analytics site. For help on setting up your account, see the Google Analytics help page.

4.4.3 Collection Composer

4.4.3.1 The Connexions Collection Composer

The Collection Composer (formerly known as the Course Composer) allows you to create new collections or edit existing collections. Any number of modules within Connexions’ Content Repository can be used to create a collection. You can perform the following tasks with the Collection Composer:

- Add and arrange elements (Section 4.4.3.2: Adding and Arranging Elements)
  - Add a module (Section 4.4.3.3: Add a Module)
  - Add a new subcollection (section) (Section 4.4.3.4: Add a New Subcollection (Section))
  - Rearrange contents (Section 4.4.3.5: Rearrange Contents)
  - Remove contents (Section 4.4.3.6: Remove Contents from the Collection)
  - Advanced editing of the collection’s elements (Section 4.4.3.7: Advanced Editing of the Collection’s Elements)
    - Provide an alternate module title (Section 4.4.3.7.1: Provide an Alternate Module Title)
    - Select the module version (Section 4.4.3.7.2: Select the Module Version)
    - Edit or add links on a module (Section 4.4.3.7.3: Edit or Add Links on a Module)
- Edit an existing collection (Section 4.4.3.8: Edit an Existing Collection)

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.4.3.2 Adding and Arranging Elements

At this point, you should have landed on the "Contents" tab of the collection composer. This is where you will add and arrange the contents of your collection. Basic instructions are provided at the top, but we will go through each editing feature in detail.

**Image not finished**

*Figure 4.87:* A collection in the making, before any content has been added.

At first, all that is shown is the collection’s title (in this case "Tutorial") and the three actions you can take at the collection’s top level:

- Add subcollections
- Add modules
- Edit title

Let’s start by adding the introductory module to the collection structure.

4.4.3.3 Add a Module

You can add already published modules to the collection from the Content Repository. You cannot add unpublished modules from your workspaces. When adding a module, Connexions does not copy the module into your collection. **The module remains in the Content Repository and Connexions inserts a reference or a pointer to the module in your collection.** A module cannot be inserted into a collection more than once.

4.4.3.3.1 Instructions for adding a module

In our example, we first want to add a module entitled *Connexions Workshop: Introduction* to the top of the collection. To do this, we start by hovering the mouse over the collection’s title. To the right, there is a link reading "Add modules":

**Image not finished**

*Figure 4.88:* Hover over the collection title to see the option to "Add modules" to the collection.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Click the "Add modules" link. A pop-up appears with a search box. Enter a word, name, or text string related to the module in the search text box. Click the "Search for modules" button:

![Image not finished]

**Figure 4.89:** A pop-up appears in which you can use to search for modules to add.

Results will appear in the same pop-up. If you don’t find the module you were searching for on first attempt, you can try using a different search term, sorting the results by another parameter, adjusting the number of searches that display per page, or going to the next page of results. Once you find the module or modules you want to add, select the checkbox next to its title(s) and click "Add selected content".

![Image not finished]

**Figure 4.90:** Select the checkbox of each module you want to add to the collection.

The module you’ve selected to add to the collection will now appear below the collection’s title in the composer. Remember that this only a pointer to the published module, not a copy of the module itself.

![Image not finished]

**Figure 4.91:** The added module appears in the composer.

Let’s now continue on to the next step of adding our two subcollections.
4.4.3.4 Add a New Subcollection (Section)

A subcollection is a grouping of related modules within a collection. Depending on the nature of your collection, you might think of a subcollection as a section, chapter, part, folder, division, etc. Each subcollection has a heading that identifies its contents to the viewers. You may place subcollections within other subcollections, as deeply nested as desired.

4.4.3.4.1 Instructions for adding a subcollection

In our example, our collection is structured by two "Parts" that follow the introductory module. To add these, we start by hovering the mouse over the collection’s title. To the right, there is a link reading "Add subcollections":

![Image not finished](image)

**Figure 4.92:** Hover over the collection title to see the option to "Add subcollections" to the collection.

Click the "Add subcollections" link. A pop-up appears with a text box. Enter the titles of the subcollections you want in your collection, one per line. Click the "Add new subcollections" button:

![Image not finished](image)

**Figure 4.93:** A pop-up appears in which you can add the titles of your collection’s subcollections.

Your subcollections will now appear in the composer below any other modules or subcollections you’ve already added at that level. From there, you can hover your mouse over each subcollection and find a list of actions to take at the subcollection level, such as adding modules, adding nested subcollections, or editing the subcollection’s title.
Let’s add some more modules to each of these subcollections so that your collection structure now looks like this:

Before things are done, we have some changes we want to make. Let’s continue.

4.4.3.5 Rearrange Contents

Since the collection composer adds modules and subcollections to the bottom of the collection or subcollection where you added them, you will sometimes need to rearrange the modules and subcollections to get them in the desired ordered. To do this, simply click down on the title of the item you want to move, drag it over to the desired location, and click up to drop it in that location. A dotted line appears where there is a valid location to drop the item.
Figure 4.96: (a) Click down on the title of a module or subcollection to move it. (b) Drag the module or subcollection over to its new location. A dotted line appears where there is a valid location to drop the item.

NOTE: Rearrangements don’t have to be strictly vertical movements of modules, as shown in the example. You can move both modules and subcollections, and you can move them up, down, into subcollections, or out of subcollections. The new structure is automatically saved when the item is dropped in place.

4.4.3.5.1 Collapsing/expanding subcollections for easier editing

If your collection contains a lot of modules and subcollections, it can become cumbersome to edit (especially for dragging and dropping). If this is the case in your collection, you can collapse the subcollection by clicking on the arrow to the left of the subcollection title.

Figure 4.97: Click the arrow icon next to a subcollection’s title to toggle it open or closed.

Subcollections without any titles do not have the expand/collapse toggles next to their title.

NOTE: When first checking out a collection, or when returning to the collection composer after having gone elsewhere, all subcollections appear closed by default. To see and/or edit their contents again, simply click the arrow toggle next to each collection title.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.4.3.6 Remove Contents from the Collection

To remove a module or subcollection from the collection, hover your mouse over the item. At the right, there will be a link reading "Remove from collection". In the example, let’s say you’ve decided not to include the module titled Module from Word Document:

![Image not finished](https://example.com/image.jpg)

Figure 4.98

A pop-up appears confirming that you want to remove the module. Behind the pop-up, the module marked for deletion is highlighted in pink with its title crossed out:

![Image not finished](https://example.com/image.jpg)  
(a)  
![Image not finished](https://example.com/image.jpg)  
(b)

Figure 4.99: (a) Confirming removal of a module. (b) Example showing the removal of an entire subcollection. Note that the you are shown an extra warning about any modules or subcollections inside the subcollection you are about to remove.

NOTE: Removing modules from a collection does not remove them from the Content Repository. It only removes the pointer in your collection to that published module.

4.4.3.7 Advanced Editing of the Collection’s Elements

4.4.3.7.1 Provide an Alternate Module Title

Modules that are published in the Content Repository display the titles given to them by their authors. If these titles are not appropriate for the module when it appears in the context of your collection (e.g., it may contain numbering), you can provide an alternate title for the module. The module displays your alternate title when it is accessed from

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
within your collection. The module displays its original title whenever it is accessed from outside of your collection.

In our example, let’s rename "Creating a Module in Minutes" to "Creating a Connexions Module" in order to match the similarly titled module in Part II of the collection. Hover over the module’s title and click on the "Edit title/version" link that appears to the right:

![Image not finished](image-url)

Figure 4.100

A pop-up appears. In the text field under "Alternate Title", type the title with which you want to override the module’s title. Click "Save".

![Image not finished](image-url)

Figure 4.101

If you want to undo the override, simply return to the pop-up, erase the text from the "Alternate Title" text field, and click "Save" again. The module’s title will revert to its original one.

4.4.3.7.2 Select the Module Version

When a module is updated or edited, Connexions saves the previous version of the module in history and uses the latest version of the module for viewing. Each new version of a module is given its own version number. You can either have your collection always use the latest version of the module, or you can pinpoint a specific version to be used.

In the example, let’s say that version 1.2 of one of the modules we include was a complete reformatting of that module’s content, and that the new format no longer makes sense in the context of your collection. Hover over the module’s title and click on the "Edit title/version" link that appears to the right:

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
A pop-up appears. In the "Set Version" area, select version 1.1 from the dropdown. Click "Save".

4.4.3.7.3 Edit or Add Links on a Module

Links allow the viewer to display related information or examples that are not part of the collection with the click of a mouse. Links may be defined by the module’s author, but they may also be added or redefined/overridden/removed for that module by the collection’s author. Any changes you make in the collection composer to a module’s links only display when viewing the module from within your collection.

In the example, let’s say that you felt strongly that the introductory module really needed a link to the Connexions home page. To add this link, hover your mouse over the module’s title. Click the link at right reading "Edit featured links":

A pop-up appears, and it includes a form where we can enter the new link’s information. (If this module’s author had already included links in the module, the form would have been hidden behind a link reading "Create a new link"). Enter some text for the link, select which type of link it is, the link’s strength (how strongly related the link is to the content) and either a URL or the ID of another collection.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Click "Add New Link". The link will then be added to a table. From there, you can edit the link, remove it, or add another one. When you are finished, click the "Done" button.

Now the links will be present on the module when viewed from inside your collection.

4.4.3.8 Edit an Existing Collection

You can edit an existing collection with the Collection Composer in the same way you create a new collection. You can edit only collections for which you have amaintainer role. To use the Collection Composer on an existing collection, first make sure you are logged in, then you can:

- Either derive a copy (Section 4.5) of a collection (if you wish to make an adaptation of the original collection),
- Or checkout (Section 3.3.2.2: Checkout Content from the Repository) a collection (yours or someone else’s) into one of your work area.

4.4.4 Collection Parameters

Modules in different collections may use different ways of presenting the same items, such as vectors and imaginary numbers. You can select the presentation of these items in your collection with the display parameters. You can also set some parameters related to the printing of your book, such as typeface selection. To set these parameters for your collection, use the following steps:

1. Click the "Parameters" tab in the Collection Composer.

---

120 This content is available online at <http://cnx.org/content/m38769/1.1/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2. Select the radio buttons for the desired notations. The defaults appear in the first column.
3. Click the "Update Properties" button.

Connexions applies the specified notation parameters to all the modules when viewed from inside your collection.

4.4.5 Preview Collection

After editing the collection, you should preview the collection to verify that it appears as you want it to appear. Click the "Preview" tab in the CollectionComposer (Section 4.4.3) to display the collection online, just as a student would view the finished collection. This preview allows you to verify that the collection displays as you intended:

Unlike when previewing modules (Section 4.3.6.1), you cannot preview the print version of a collection.

4.5 Creating Adaptations

4.5.1 What is an adaptation?

The Derive Copy function in Connexions allows you to create a derivative work ("adaptation") that is based on any published module or collection. An adaptation starts with a copy of existing content, to which you make changes and then publish as your own, with

---

121 This content is available online at <http://cnx.org/content/m38770/1.1/>.
122 This content is available online at <http://cnx.org/content/m11804/1.8/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
proper attribution to the original authors. The creation of adaptations is not a method of collaboration, though it is possible to add the original authors to your new content, should they wish to collaborate with you on your new adaptation.

WARNING: Any adaptations you create is a snapshot of the original at the time you created the work. If at a later time the original content is updated, your adaptation will not reflect those updates.

Adaptations are permitted under the Creative Commons Attribution License\textsuperscript{123}, which applies to all Connexions content. When you publish an adaptation, Connexions automatically adds an attribution statement that identifies the original work and its author, which is legally required by the Creative Commons Attribution License.

4.5.1.1 Examples

Some examples of adaptations are:

- You may wish to create a module or collection that is similar to an existing one, but with some changes. For example, you may find an existing module with excellent explanations that uses an example that won’t make sense for your students. You can create a derived copy with a new example. Or perhaps you want to you a module in your collection, but the module contains errors. You contact the original author and suggest corrections. The original author does not respond to your suggestions, so you create a derived copy of the module, insert your changes, and publish the new module.
- For example, say you are teaching two mathematics courses, but that each course uses a different calculator. You can write modules that show instructions for functions using one of the calculators, and then derive a copy of those modules to show the same functions using the other calculator.
- Derived copies provide a mechanism whereby content can be translated from one language into another.

4.5.2 Creating Adaptations

There are two ways to derive copies of existing content:

1. If there is a published collection or module that you would adapt, you can now quickly derive a copy directly from the content page itself. At the top of each page of published content, there is now a Reuse / Edit dropdown link (Section 4.5.2.1: Through the "Reuse/Edit" menu).

\textsuperscript{123}http://creativecommons.org/licenses/by/3.0/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2. Or you can checkout (Section 3.3.2) content to one of your work areas and derive a copy at a later time (Section 4.5.2.2: From a checked out copy) from the Edit Content page. (This method is more convenient especially if you wish to derive copies of multiple items. First do a batch checkout (Section 3.3.2.2: From the search/browse returns listing), and then derive a copy later.)

4.5.2.1 Through the "Reuse/Edit" menu

To check out an editable copy of a collection or module, Search the Content Repository[124] for the content you want to copy. Click on the title of the content you want to copy. Near the top right corner, select the menu reading Reuse / Edit and click the "Reuse or edit" link for the content you want to copy. (If you are viewing a module in the context of a collection, there are "Reuse or edit" links for both the collection and the module.)

![Figure 4.109: The ‘Reuse / Edit’ menu in the Content Actions bar.](http://cnx.org/content/search)

You will be prompted to log in if you have not already done so. Then,

- **If you have permission to edit the content**, it is assumed that you want to checkout (Section 3.3.2.2: Checkout Content from the Repository) a copy of the content. If you do actually wish to derive a copy and create an adaptation of your own content, you can do that from the Edit Content page after you checkout a new copy of the content (see below (p. 201)).

You will be prompted to select a work area to edit the content in.

---

[124]http://cnx.org/content/search

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Figure 4.110: ‘Choose a work area to check out content to’ popup.

If you leave the "Edit the content now" checkbox selected, you will be redirected to the work area in order to immediately start editing the content.

- **If you do NOT have permission to edit the content**, you will be prompted either to:
  - Derive a copy of the content. Deriving a copy (Section 4.5.1: What is an adaptation?) allows you to adapt this content and then publish your adaptation.
  - Or check it out anyway. You might choose to checkout (Section 3.3.2.2: Check-out Content from the Repository) the content, even though don’t have permission to publish changes. You may ask to be a co-author later, derive a copy later, or just want to see how the content is constructed.

![Image of Reuse/Edit dialog box]

Figure 4.111: Derive a copy or check it out anyway.

The next step in the pop-up will ask you to select a work area in which to edit the content.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Figure 4.112: Choose a work area and accept the license.

If you leave the "Edit the content now" checkbox selected, you will be redirected to the work area in order to immediately start editing the content. You will also need to agree to the Creative Commons Attribution License\(^\text{125}\) for your newly derived content. Agree to the license and click Yes, Create Copy to proceed.

At this point, an adaptation of the original content will be checked out in the work area you selected above. All the contents of the new module or collection are exactly the same as the contents of the original. The metadata is the same, except for the roles. The roles now show you as an author, maintainer, and copyright holder. You can edit the content of the module or collection, its files, metadata, roles, and links.

\(^{125}\)http://creativecommons.org/licenses/by/3.0/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
as necessary.

4.5.2.2 From a checked out copy

If you chose to checkout (Section 3.3.2.2: Checkout Content from the Repository) content for which you did not hold the maintainer role, then if you try to publish the checkout, you will be prompted to derive a copy.

![Image not finished](image1)

**Figure 4.113:** The "Publish" option is blocked.

You can derive a copy of any item in one of your Work Areas at any time. Click on the content to edit it. Then click the **Derive Copy** link in the Module Status portlet on the right side of the page.

![Image not finished](image2)

**Figure 4.114:** The Module Status portlet with a "Derive Copy" link.

4.6 Create Lens

4.6.1 Create a lens\(^{126}\)

Go to your MyCNX\(^{127}\) page and click on the **Create a lens** link underneath the "Access Lenses" header.

![Image not finished](image3)

**Figure 4.115:** Access Lenses section of MyCNX

---

\(^{126}\)This content is available online at <http://cnx.org/content/m43300/1.1/>.

\(^{127}\)http://cnx.org/content/m43300/latest/../mycnx

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
On the next page, choose a name for your lens and select the type of lens (Section 2.3) you want to create. If you want to create an Endorsement or Affiliation lens, you must be logged in to an organizational account (Section 4.6.3).

The next page allows you to enter metadata about your lens and set initial properties, including privacy settings and tag type. If you are using an organizational account (Section 4.6.3) you may choose a logo and a banner color to be associated with your lens.

![Image not finished](image1)

**Figure 4.116:** Uploading a logo

Your logo and branding bar will show up near the top of the page whenever someone views content in the context (Section 2.1.3: Lens Context) of your lens. That means that if they first click on your lens to view your lens page, and then click on any of the content you have highlighted in your lens, your logo and branding bar will remind them that this content is associated with your lens.

![Image not finished](image2)

**Figure 4.117:** An example of a branding bar

Most of these properties can be changed at any time. To do so, log in and go to your MyCNX\(^{128}\) page. Under "Access Lenses", find your lens and click the (edit) link beside it. From there you can click the green **Edit lens properties** tab to change your settings, or click the green **Edit lens content** tab to modify any content already in your lens.

![Image not finished](image3)

**Figure 4.118:** Edit lens properties tab

---

\(^{128}\)http://cnx.org/content/m43300/latest/../mycnx

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4.6.2 Populate a lens\footnote{This content is available online at \url{http://cnx.org/content/m43304/1.2/}.}

To add a module or a collection to your lens, you must first navigate to that module or collection (Section 2.4).

At the top of the module in the Content Action bar (Section 2.1.1.1: Content Action Bar) is the option \textit{Add to lens}. Hover your mouse over this option and choose "Add to my lens" from the dropdown menu.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{add_to_lens.png}
\caption{Add to lens dropdown}
\end{figure}

You may add your own tags and description associated with this content. The tags and description will only show up on your lens. Viewers may filter content in your lens by tags.

4.6.3 Create an organizational account\footnote{This content is available online at \url{http://cnx.org/content/m43299/1.1/}.}

In order to set up an endorsement or affiliation lens, you will first need to set up an organizational account.

1. Create a personal account using the sign-up form\footnote{http://cnx.org/content/m43299/latest/../join_form}. You will need a valid email address in order to complete the registration process.
2. After logging into the site, create an organizational account\footnote{http://cnx.org/content/m43299/latest/../org_account_form}.
3. Log out of your personal account and log into the newly created organizational account.
4. Click on the MyCNX tab at the top of the screen and click on the Create a lens link.
5. Provide a name for the lens, select a lens type, and click on the Continue button.
6. Customize your lens by providing additional information such as a description, logo, website, etc. (Note: This information should provide readers with a description of your organization as well as the criteria used to endorse content (in the case of an endorsement lens). Also, you may wish to leave your lens in a "Private" state until you have had a chance to add content to the lens - this option can be changed at any time.)
7. Save your lens.

Once created, you will be able to add content to your lens by using the Add to a lens feature in the Content Actions (Section 2.1.1.1: Content Action Bar) bar at the top of the content. If you wish to modify or remove items from your lens, or to make your lens public or private, you can do so by clicking on the MyCNX tab, clicking on the View all my lenses link, and making the necessary changes to your lens.
Chapter 5

CNXML Tutorial

5.1 XML Basics

5.1.1 What is XML?

The eXtensible Markup Language (XML) is a meta-markup language defined by the World Wide Web Consortium (W3C). It is not strictly a markup language itself, but rather a set of rules for creating markup languages. For our purposes a markup language is any language (HTML, for example) that uses tags surrounding text to convey information such as content or format. CNXML, the markup language used by the Connexions Project is an example of a language written in XML. There are many other examples at the W3C site. Here is an example of some markup in CNXML.

Example 5.1

<para>
This is a paragraph in <term>CNXML</term>. Notice that the markup contains tags that express the meaning of the text.
</para>

<para> and </para> are the tags that enclose the text. In XML, tags are always marked by angle brackets (also known as < and >). Tags generally come in pairs. An opening tag will look like <tagname>. A closing tag will look like </tagname>, with a / preceding the tag name.

---

1This content is available online at <http://cnx.org/content/m9002/2.24/>.
2http://www.w3.org
3http://cnx.rice.edu/cnxml
4http://cnx.rice.edu

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
XML allows the separation of presentation from content. For example, HTML has tags such as `<u>` and `<i>`, which underline and italicize text respectively. This does not express content information, only formatting. XML allows you to define your own language of tags to represent content. You could create a tag called `<book>` to represent book titles, and create a stylesheet (a separate formatting document), that says that every `<book>` tag should be italicized or underlined. Then when you want to change the presentation of that type of content, you just change one small part of the stylesheet. Also, if you make tags that convey the content of the document, you can enable better searching. For example, you might look for the author of a document by looking at the author tag.

5.1.2 Well-formed XML

XML has a few rules that apply to all of its languages, including CNXML. If a document satisfies these rules, then it is well-formed. XML documents are required to be well-formed.

- Every tag that is opened must be closed. An opening tag looks like `<module>` and a closing tag looks like `</module>`. There is a shortcut. If your tag contains no other tags (referred to as an empty tag), then you can can type a `/` before the end of the opening tag and delete the closing tag. For example, `<media> </media>` can be abbreviated `<media/>`.
- Tags must be nested within each other. So, `<b>red <i>and</i> blue</b>` is fine, but `<b>red <i>and</i> blue</b>` is incorrect because the `<b>` and `<i>` tags have overlapping content.
- You must put either single or double quotes around an attribute value. An attribute is some sort of information that is associated with a tag and is listed inside of the tag itself. For example, `<module id="m0001"/>` and `<module id='m0001'/>` are fine, but `<module id=m0001/>` is incorrect.
- You can also choose to start every document with an XML declaration. If you do use the XML declaration, then it has to be the very first thing in the file. It cannot even be preceded by whitespace. It is not considered to be a tag. The XML declaration is as follows. `<?xml version="1.0"?>` You can also include other information such as the encoding of the document or whether the document depends on other files or not.
- There must be one tag that contains all of the other tags. For example in xhtml `<html>` and `</html>` must surround all of the other tags. There are some things that are included at the top of the document that are not tags and that are not included with the tags. The XML declaration is an example of this.

5.1.3 Valid XML

It is possible to define a set of rules that apply to all of the tags in a particular XML language. These rules can be defined in a couple of different ways. The most common
way is to use a **DTD** (Document Type Definition). Any document which follows all of the rules for that language is called **valid**. A document is not required to be valid in order to be XML. However, it is generally a good idea.

### 5.1.4 Entity References

**NOTE**: Entity references are no longer supported by CNXML 0.6. Instead, we suggest that you use character references as described below (Section 5.1.5: Character References) to add special characters to your module.

XML uses several characters in special ways as part of its markup, in particular the less-than symbol (`<`), the greater-than symbol (`>`), the double quotation mark (`"`), the apostrophe (`'`), and the ampersand (`&`). You’ve already seen examples of markup using the first four of those previously in this module. But what if you need to use these characters in your content, and you don’t want them to be treated as part of the markup by XML processors? You can use XML entity references for this purpose. The XML Specification defines the following five entity references for use in any well-formed XML document:

- `&amp;` refers to an ampersand (`&`)
- `&lt;` refers to a less-than symbol (`<`)
- `&gt;` refers to a greater-than symbol (`>`)  
- `&quot;` refers to a double-quote mark (`"`)  
- `&apos;` refers to an apostrophe (`'`)  

**Example 5.2**
Suppose you have a document with the following:

```xml
<para id="p1">The firm was known as Scrooge and Marley.</para>
```

you could replace ‘and’ with the entity reference `&amp;`:

```xml
<para id="p1">The firm was known as Scrooge &amp; Marley.</para>
```

All entity references outside the above five must be defined in a document type declaration, and they may only be used in documents that conform to that DTD. Note that an entity reference always begins with `&` and ends with `;`.  

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.1.5 Character References

You can also use any character defined in Unicode in an XML document by means of character references. Unicode is a project to define a unique code for every character in any human language. Unicode is very useful any time that you need to use a symbol that is not a part of ASCII.

Character references in XML either begin with &#, or they begin with &#x, and they end with a semicolon ;. A character reference contains a representation of a Unicode code point: if it begins with &#, then it contains a decimal representation of a Unicode code point; if it begins with &#x, then it contains a hexadecimal representation of a Unicode code point.

Example 5.3
The hexadecimal representation of the Unicode code point for the small 'o' with a stroke is 00F8, and the decimal representation for the same is 248. Therefore, the character references for the small 'o' with a stroke are &#00F8; and &#248; So you could write

\[ \text{<emphasis>The majestik møøse</emphasis>} \]

or

\[ \text{<emphasis>The majestik m&#248;se</emphasis>} \]

or even

\[ \text{<emphasis>The majestik m&#x00F8;e</emphasis>} \]

to get

The majestik møøse

5.2 Combining XML Languages\(^5\)

XML\(^6\) allows you to create documents in custom markup languages. But what if you want to combine markup from multiple languages in the same document? What if there are one

\(^5\)This content is available online at <http://cnx.org/content/m10159/2.16/>.

\(^6\)http://www.w3.org/XML

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
or more tags that exist in both languages, but with different meanings? You could, for example, have a `<table>` tag in HTML and one in a language describing office furniture as well. How do you use these tags unambiguously, without losing functionality?

The solution is to use an extension to XML called **namespaces** (See the W3C’s recommendation, Namespaces in XML\(^7\)). A namespace associates a unique global identifier (usually a URI) with a particular set of tags and their usage rules. To declare a namespace for a particular tag, set the `xmlns` attribute to the value of the unique identifier.

You can also define a **namespace prefix** for use in your document. To do this, use a modified version of the `xmlns` attribute. For example, you would use the attribute `xmlns:foo="http://somewhere.org/foo"` to associate the prefix `foo` with the namespace identifier `http://somewhere.org/foo`. You can then indicate which tags come from that namespace by adding the appropriate prefix to each tag. Thus, the `bar` tag in foo’s namespace would be written as `<foo:bar>` and `</foo:bar>`.

When you use the default namespace any children of that tag lacking an explicit prefix will be assumed to have come from the same namespace. This allows you to define a default namespace for all of the children of a tag. This is especially useful when used on the **root node**, which is the outermost tag in a document.

**Example 5.4**

For CNXML 0.6 there is only one schema. The document tag will contain the namespace for all available languages and will look like this:

```xml
<document xmlns="http://cnx.rice.edu/cnxml"
    xmlns:md="http://cnx.rice.edu/mdml/0.4"
    xmlns:bib="http://bibtextml.sf.net/"
    xmlns:m="http://www.w3.org/1998/Math/MathML"
    xmlns:q="http://cnx.rice.edu/qml/1.0"
    id="new"
    cnxml-version="0.6"
    module-id="new"/>
```

\(^7\)http://www.w3.org/TR/REC-xml-names/
5.3 The Basic CNXML

5.3.1 Starting with CNXML

CNXML is a lightweight XML markup language for marking up educational content. The goal of CNXML is to convey the content of the material and not a particular presentation. Connexions uses the Connexions Markup Language (CNXML) as its primary language for storing documents. Now let’s get started!

5.3.2 CNXML Tags

5.3.2.1 Document

All CNXML documents have as their root the document tag. Everything about the document including its metadata and content are contained within the document tag. It is important that you understand the basic structure for a CNXML document. The structure is as follows:

Document (root tag)

- Title
- Metadata Section
- Content Section

The document tag has one required attribute:

- id - a unique ID given to the document.

This is automatically assigned.

5.3.2.1.1 ID Requirements

One major difference between CNXML and other markup languages is the id attribute requirement. Certain tags require that you include the id attribute, but all can possess an id. The tags requiring an id are listed below:

- document
- para
- equation
- list
- rule
- definition
- exercise

---

8 This content is available online at <http://cnx.org/content/m9000/2.36/>.
9 http://cnx.org

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
So, if you are going to use any of the above tags, be sure to add the id attribute and give it a unique ‘id’. Be aware that in CNXML 0.6 ids will be generated automatically, but you are still permitted to specify your own ids if you wish.

**Example 5.5**

Here is an example of a couple of paras containing a user generated ids.

```<para id='uniqueid1'>
  This is an example to illustrate the use of the `<code>id</code>` attribute.
</para>
<para id='uniqueid2'>
  This paragraph has a different id than the last.
</para>```

**NOTE:** Any tag can contain an id attribute. This is useful if you want to link to the information contained in a particular tag.
5.3.2 Namespaces

The document tag should also contain any namespace declarations. Namespaces (Section 5.2) allow us to easily use other mark-up languages within CNXML without having to worry about whether tag name collision will occur. For simple documents using only CNXML, you need to include the CNXML namespace attribute. Any additional languages need to be declared as well and should be given their own prefixes. For example, to associate the MathML namespace with the prefix "m", include the following attribute:

```
xmlns:m='http://www.w3.org/1998/Math/MathML'
```

This states that any tag with a prepended "m" will be interpreted as a MathML tag while any tag without a prefix will be interpreted as CNXML. The document tag should also contain the metadata namespace

```
xmlns:md='http://cnx.rice.edu/mdml/0.4'
```

the bibtex namespace

```
xmlns:bib='http://bibtexml.sf.net/'
```

and the question markup language

```
xmlns:q='http://cnx.rice.edu/qml/1.0'
```

Example 5.6

This what the document tag should look like.

```xml
<document xmlns="http://cnx.rice.edu/cnxml"
xmlns:m='http://www.w3.org/1998/Math/MathML'
xmlns:md='http://cnx.rice.edu/mdml/0.4'
xmlns:bib='http://bibtexml.sf.net/'
xmlns:q='http://cnx.rice.edu/qml/1.0" id="m9000" module-id="" cnxml-version="0.6">

```

NOTE: Be aware that the you document id can not be the same as this example. Each module will have its own unique id.

5.3.2.2 Title

The title tag can be used with many CNXML tags to hold the name of its parent. This tag can only contain information in ASCII text or MathML. I mention it here to allow you to put in the name of the module (since I mentioned that it was the first required tag in the document tag).

Example 5.7

```xml
<title>Grilling a Good Steak</title>
```

NOTE: Please see the CNXML tag list in Edit-In-Place\(^\text{10}\) to see if a tag can be named.

\(^{10}\text{http://cnx.org/eip-help/tags}\)

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.3.2.3 Content

Now that you have the document tag set up with an id and namespace info, the next thing to do with your document is add content. By ‘content’ I mean the text that will make up the bulk of your document.

NOTE: Strictly speaking the metadata should precede information about content, but we will leave this until later.

All of this content is conveniently placed in the content tag. Every CNXML document will have one content tag. The body of the document will be here inside the content tag.

Structural tags are the tags which are used inside of the content tag to give structure to the document. These tags are discussed below.

5.3.2.3.1 Structure Tags

Some of the structure tags are section (Section 5.3.2.3.1.2: Section), para (Section 5.3.2.3.1.1: Para), document (p. 210), and content (Section 5.3.2.3: Content).

We have already discussed the document (p. 210) and content (Section 5.3.2.3: Content) tags, so we will proceed with a short description and examples of the other other tags listed.

5.3.2.3.1.1 Para

Text can be inserted into documents by using the para tag. Each para has a required id which must be unique within the document.

Example 5.8

<para id='intro'>
I have eaten many steaks in my life and none have been more satisfying than the backyard-grill cooked steak. Maybe this is because of the relaxing nature of drinking a beer, being outside, and lounging that accompanies the grilling procedure. Maybe it is because of the aroma of the grill and the beef perfectly seasoned to your taste. Either way, this document shows how a good steak can be prepared.
</para>
5.3.2.3.1.2 Section

As often is the case in textbooks, chapters are divided into smaller sections. Because it is often necessary to segment text for better understanding and coherence, CNXML has included a section tag.

The section tag has one required attribute, id, and a optional first child tag, title (Section 5.3.2.2: Title).

Example 5.9

```xml
<section id='ingredsec'>
  <title>Ingredients</title>
  <list> ... </list>
</section>
<section id='marinadesec'>
  <title>Marinade</title>
  <para id='marinate'> ... </para>
  <list id='marinade'> ... </list>
  <para id='tobecontinued'> ... </para>
</section>
<section id='grillingsec'>
  <title>Grilling</title>
  <para id='prepgrill'> ... </para>
  <para id='grilling'> ... </para>
</section>
```

Obviously ellipses would be replaced by appropriate text.

5.3.2.3.2 Inline Tags

Inline tags are used to embed content and functionality inside of the structural tags. Some of the more commonly used tags are discussed below.

5.3.2.3.2.1 Emphasis

The emphasis tag is used to emphasize text in a CNXML document where emphasis in text would be needed or desired. It is important to note that this refers to semantic emphasis and not a typeface, although many stylesheets may choose to render it visually with a different typeface.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
I have eaten many steaks in my life and none have been more satisfying than the backyard-grill cooked steak. Maybe this is because of the relaxing nature of drinking a beer, being outside, and lounging that accompanies the grilling procedure. Maybe it is because of the aroma of the grill and the beef perfectly seasoned to your taste. Either way, this document shows how a good steak can be prepared.

5.3.2.3.2.2 Term

The term tag is used to mark words or phrases which are being defined. However, its use is confined to either a para (Section 5.3.2.3.1.1: Para) or definition tag. The term tag has several optional attributes:

- url - a URL specifying the source or definition of the term.
- window - contains the possible values "replace" which results in the associated url opening in the present window, and "new" which result in the associated url opening in a new window or tab.
- document - the id of another Connexions module or collection.
- target-id - the id of a specific element (such as a para or section) in the current or another Connexions document.
- resource - This reference points to a file that is associated with the term in question. The resource could be a pdf, text file, or any other supplementary resource.
- version - The version of a Connexions module or collection. This attribute is used in conjunction with the document attribute.
- id - A unique identifier, whose value must begin, with a letter and contain only letters, numbers, hyphens, underscores, colons, and/or periods (no spaces).

To ensure the best flavor possible, it is necessary to marinate the
beef. A steak \(<\text{term url='http://marinade.com'}>\text{marinates}</\text{term}>\) when left to sit in a prepared sauce, or \(<\text{term}>\text{marinade}</\text{term}>\), where it will absorb the flavors of the ingredients. Marinating may take as little as 15 minutes or as long as 6 hours and should \(<\text{emphasis}>\text{always}</\text{emphasis}>\) be done in the refrigerator and \(<\text{emphasis}>\text{not}</\text{emphasis}>\) at room temperature.
</para>

5.3.2.3.2.3 Note
The note tag creates a note to the reader, which could be a warning, tip, etc. There are five allowed types of note: note; aside; warning; tip; important. The type of note is specified by an optional type attribute.

Example 5.12

<para id='intro'>I have eaten many steaks in my life and none have been more satisfying than the backyard-grill cooked steak. Maybe this is because of the relaxing nature of drinking a beer, being outside, and lounging that accompanies the grilling procedure. \(<\text{note type='warning'}>\text{Excessive drinking or fun may result in overcooked or burned steak.}</\text{note}>\) Maybe it is because of the aroma of the grill and the beef \(<\text{emphasis}>\text{perfectly}</\text{emphasis}>\) seasoned to your taste. Either way, this document shows how a good steak can be prepared.
</para>

The above markup will display as:

I have eaten many steaks in my life and none have been more satisfying than the backyard-grill cooked steak. Maybe this is because of the relaxing nature of drinking a beer, being outside, and lounging that accompanies the grilling procedure.

WARNING: Excessive drinking or fun may result in overcooked or burned steak.

Maybe it is because of the aroma of the grill and the beef perfectly seasoned to your taste. Either way, this document shows how a good steak can be prepared.
5.3.2.3.2.4 Link

The link tag is used to provide a quick link to other Connexions modules, collections or external websites. The link tag can contain the following attributes.

- **strength** - The Strength attribute can contain the values 1, 2, or 3 (with 3 being the strongest) specifying the relevance of the link.
- **url** - a URL specifying the source or definition of the term.
- **window** - contains the possible values "replace" which results in the associated url opening in the present window, and "new" which result in the associated url opening in a new window or tab.
- **document** - the id of another Connexions module or collection.
- **target-id** - the id of a specific element (such as a para or section) in the current or another Connexions document.
- **resource** - This reference points to a file that is associated with the term in question. The resource could be a pdf, text file, or any other supplementary resource.
- **version** - The version of a Connexions module or collection. This attribute is used in conjunction with the document attribute.
- **id** - A unique identifier, whose value must begin with a letter and contain only letters, numbers, hyphens, underscores, colons, and/or periods (no spaces).

The target and document attributes can be used together or alone. If both are used then you will link to a particular tag in another document. If only document is used, you will link to another document. If only target is used, you will link to a particular tag within the current document.

5.3.2.3.2.5 Cite

The cite tag is used to refer to non-electronic materials within a document, primarily containing the title of a work. Cite has several optional attributes:

- **url** - a URL specifying the source or definition of the term.
- **window** - contains the possible values "replace" which results in the associated url opening in the present window, and "new" which result in the associated url opening in a new window or tab.
- **document** - the id of another Connexions module or collection.
- **target-id** - the id of a specific element (such as a para or section) in the current or another Connexions document.
- **resource** - This reference points to a file that is associated with the term in question. The resource could be a pdf, text file, or any other supplementary resource.
- **version** - The version of a Connexions module or collection. This attribute is used in conjunction with the document attribute.
- **id** - A unique identifier, whose value must begin with a letter and contain only letters, numbers, hyphens, underscores, colons, and/or periods (no spaces).
5.3.2.3.2.6 Quote

The quote tag is used to denote that some text is a direct quote from some other source. The quote tag has a display attribute which denotes whether the quote is inline or block. Quote can also contain all of the attributes associated with cite (p. 217).

Example 5.13

```xml
<para id='steakquote'>
Everyone has an opinion on how a steak should be cooked. <quote display='inline'>"A good steak should be pink in the middle and black on the outside."</quote> Although this may sound reasonable many remember the words of George Washington: <quote type='block'>"In any free country a man should have the ability to purchase a nice rare steak."</quote>
</para>

Everyone has an opinion on how a steak should be cooked. "A good steak should be pink in the middle and black on the outside." Although this may sound reasonable many remember the words of George Washington:

"In any free country a man should have the ability to purchase a nice rare steak."
```

5.3.2.3.2.7 Foreign

The foreign tag is used to denote that a foreign word or phrase is being used. Foreign can also contain all of the attributes associated with cite (p. 217).

Example 5.14

```xml
<para id='steakquote2'>
In many latin american countries steak is called <foreign>carne asada</foreign>.
</para>

In many latin american countries steak is called carne asada.
```
5.3.3 Document Example Code

Below is an example of what your document could look like if you included all the tags above to make a document about making a steak.

```
<document id='meat'>
  <title>Grilling a Good Steak</title>
  <content>
    <section id='intro'>
      <para id='intro'>
        I have eaten many steaks in my life and none have been more satisfying than the backyard-grill cooked steak. Maybe this is because of the relaxing nature of drinking a beer, being outside, and lounging that accompanies the grilling procedure.
        <note type='warning'>Excessive drinking or fun may result in overcooked or burned steak.</note> Maybe it is because of the aroma of the grill and the beef <emphasis>perfectly</emphasis> seasoned to your taste. Either way, this document shows how a good steak can be prepared.
      </para>
    </section>
    <section id='marinate_section'>
      <para id='marinate'>
        To ensure the best flavor possible, it is necessary to marinate the beef. A steak <term>marinates</term> when left to sit in <term>marinade</term>, or prepared sauce, where it will absorb the flavor of the ingredients. Marinating may take as little as 15 minutes or as long as 6 hours and should <emphasis>always</emphasis> be done in the refrigerator and <emphasis>not</emphasis> at room temperature.
      </para>
    </section>
    <section id='tobecontinued_section'>
      <para id='tobecontinued'>
        I'll be adding to this document in <link document='m9006'>The Intermediate CNXML</link> which focuses on more advanced CNXML tags. For more marinades see the <link url='http://www.2eatcab.com'>Angus Beef website</link>.
      </para>
    </section>
  </content>
</document>
```
Finally, a good resource is the <cite>Steak Lover’s Cookbook -- William Rice</cite>.

</para>
</section>

</content>
</document>

See how Connexions\textsuperscript{11} would render this example (Section 5.7.1).

5.3.4 Other Required Stuff

The first line in any XML file should be the XML declaration. (Strictly speaking, this is optional, but it’s a good practice to follow). The XML declaration looks like this: \texttt{<xml version="1.0" encoding="utf-8"/>}, and must not be preceded by any blank lines or whitespace. CNXML 0.6 only uses one schema, so there is no need to specify specific DTDs. Below is an example of a correct CNXML 0.6 document tag containing the proper namespaces.

\textbf{Example 5.15}

\begin{verbatim}
<document xmlns="http://cnx.rice.edu/cnxml"
         xmlns:m="http://www.w3.org/1998/Math/MathML"
         xmlns:md="http://cnx.rice.edu/mdml/0.4"
         xmlns:bib="http://bibtexml.sf.net/"
         xmlns:q="http://cnx.rice.edu/qml/1.0"
         id="m9000" module-id="" cnxml-version="0.6">

5.3.5 Conclusions

Remember that when composing documents it is always best to consult the CNXML Tag List\textsuperscript{12} for any questions regarding the exact usage of CNXML tags. For more advanced topics see The Intermediate CNXML (Section 5.4) or The Advanced CNXML (Section 5.5), which concludes the cooking lesson.

\textsuperscript{11}http://cnx.org
\textsuperscript{12}http://cnx.org/eip-help/tags

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.4 The Intermediate CNXML\textsuperscript{13}

5.4.1 Example

As is often the case in textbooks, authors will include examples in the middle of a chapter or section. For this reason CNXML provides a tag that allows an author to include examples in a document. The example tag has a unique \texttt{id} attribute and can contain most tags as children, the first being an optional title (Section 5.3.2.2: Title). For specifics you should always consult the CNXML Spec\textsuperscript{14}.

Example 5.16

\begin{verbatim}
<example id='tboneexam'>
  <figure id='tbonefig'>
    <title>T-Bone Steak</title>
    <media id="image-example" display="block" alt="A T-bone Steak.">
      <image type='image/jpg' src='tbone.jpg'/>
    </media>
  </figure>
</example>
\end{verbatim}

5.4.2 Figure

The \texttt{figure} tag provides the structure for creating a figure within a document. They can contain either two or more subfigure (Section 5.4.3: Subfigure) tags, or a single media (p. 221), table (p. 232), or code (Section 5.5.1: Code) tag.

The \texttt{figure} tag has two attributes:

- \texttt{id} - a unique ID, required
- \texttt{orient} - defines how multiple subfigure (Section 5.4.3: Subfigure)s are to be displayed. It takes two values, \texttt{vertical} or \texttt{horizontal}, and will default to \texttt{horizontal}.

The optional first tag of the \texttt{figure} tag is title (Section 5.3.2.2: Title) which is used to title a figure.

\textsuperscript{13}This content is available online at \texttt{<http://cnx.org/content/m9006/2.22/>}.
\textsuperscript{14}http://cnx.rice.edu/technology/cnxml/0.5/spec

Available for free at Connexions \texttt{<http://cnx.org/content/col10151/1.27>
The title tag is followed by any of the tags listed above; however, the most commonly used tag is media, which is used to include any sort of media such as images, video, music, or java applets. The media object tags have two required attributes:

- **src** - the location of the displayed media
- **mime-type** - defines the type of media being displayed, which can be any valid MIME\textsuperscript{15} type.
  
  - audio - audio/mp3, audio/wav, etc.
  - video - video/qt, video/mov, etc.
  - image - image/png, image/gif, etc.
  - application - application/PostScript, application/x-java-applet, etc.

The final tag is the optional caption which is used to add a small caption to the figure.

**Example 5.17**

```xml
<figure id='tbone'>
  <title>T-Bone Steak</title>
  <media id="image-example" display="block" alt="A T-bone Steak.">
    <image mime-type='image/jpeg' src='tbone.jpg'/>
  </media>
  <caption>
    Upon successful completion of these documents, you should be able to grill a steak that looks just as good!
  </caption>
</figure>
```

### 5.4.3 Subfigure

The subfigure tag is used when you want to include more than one media (p. 221), code (Section 5.5.1: Code) or table (p. 232) within the same figure (Section 5.4.2: Figure).

The usage of the subfigure tag is similar to that of figure (Section 5.4.2: Figure). It has an optional id attribute, an optional first child title (Section 5.3.2.2: Title) tag, a single media (p. 221), code (Section 5.5.1: Code) or table (p. 232), followed by an optional caption (p. 222).

\textsuperscript{15}http://www.ietf.org/rfc/rfc1341.txt?number=1341

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Now the orient attribute for figure (Section 5.4.2: Figure) becomes very important. orient lets you specify whether the subfigures should be displayed side-to-side or one on top of the other.

**Example 5.18**

```xml
<figure orient='horizontal' id='horfig'>
  <title>Steaks</title>
  <subfigure id='subfigTBone1'>
    <title>T-Bone</title>
    <media id="image-example" display="block" alt="A T-bone Steak.">
      <image type='image/jpeg' src='tbone.jpg'/>
    </media>
  </subfigure>
  <subfigure id='subfigNYStrip1'>
    <title>New York Strip</title>
    <media id="image-example" display="block" alt="A NY Strip.">
      <image mime-type='image/jpeg' src='ny_strip.gif'/>
    </media>
  </subfigure>
  <caption>Upon successful completion of these documents, you should be able to grill a steak that looks just as good!</caption>
</figure>

Or

```xml
<figure orient='vertical' id='verfig'>
  <title>Steaks</title>
  <subfigure id='subfigTBone2'>
    <title>T-Bone</title>
    <media id="image-example" display="block" alt="A T-bone Steak.">
      <image type='image/jpeg' src='tbone.jpg'/>
    </media>
  </subfigure>
  <subfigure id='subfig2'>
    <title>New York Strip</title>
    <media id="image-example" display="block" alt="A NY Strip.">
      <image mime-type='image/jpeg' src='ny_strip.gif'/>
    </media>
  </subfigure>
</figure>
```

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Upon successful completion of these documents, you should be able to grill a steak that looks just as good!

5.4.4 List

The list tag is used to make lists. It has two attributes:

- **id** - a unique ID, required
- **list-type** - defines the formatting of the list. list-type takes the values bulleted (default), enumerated, named-item or inline

The list tag has two children: title (p. 221), which is optional, and item, which is where the list information is stored.

**Example 5.19: Example List**

```xml
<list id='marinade' list-type='enumerated'>
  <title>Beer Marinade</title>
  <item>pour beer into large bowl</item>
  <item>add chili powder to taste</item>
  <item>squeeze half lime into beer marinade</item>
  <item>place steak in beer, let soak for 30 minutes</item>
</list>
```

The resulting list will look like:

**Beer Marinade**

1. pour beer into large bowl
2. add chili powder to taste
3. squeeze half lime into beer marinade
4. place steak in beer, let soak for 30 minutes

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Example 5.20: New List Types Example
CNXML 0.6 gives you much more control over the list environment. Now you will be able to choose from eight preset bullet styles as well as an option that allows you to choose your own literal text to serve as the bullet style. The enumerated list type now offers several styles, including Arabic numerals, upper and lower case alphabet characters, and also upper and lower case Roman numerals. In addition to these changes, you can now also select to have your lists follow a stepwise progression. In CNXML 0.6 the named-item list has been slightly altered, and is now called a labeled-item list. As you may have guessed, the change is quite intuitive. Instead of using `<name>` to specify the item’s label, you use `<label>`.

Here is an example of a stepwise enumerate list:

```xml
<list id="eip-165" list-type="enumerated" number-style="arabic" class="stepwise">
  <title>Beer Marinade</title>
  <item>pour beer into large bowl</item>
  <item>add chili powder to taste</item>
  <item>squeeze half lime into beer marinade</item>
  <item>place steak in beer, let soak for 30 minutes</item>
</list>
```

The resulting list will look like:

**Beer Marinade**

**Step 1:** pour beer into large bowl  
**Step 2:** add chili powder to taste  
**Step 3:** squeeze half lime into beer marinade  
**Step 4:** place steak in beer, let soak for 30 minutes

5.4.5 Equation

The `equation` tag is used to set off and number equations in CNXML documents by using ASCII text, MathML and embedded media (p. 221) to display math.

**NOTE:** It is strongly encouraged, however, to use `equation` with MathML (Section 4.3.1.6.1) tags when displaying math.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.4.5.1 ASCII Text and Images

The first child of equation is an optional title (Section 5.3.2.2: Title) followed by any number of media (p. 221) tags.

**Example 5.21**

```
<equation id="eqn14">
  <title>Euler's Relation</title>
  <media id="equation-example" display="block" alt="Euler’s Relation.">
    <image mime-type='image/gif' src='euler.gif' />
  </media>
</equation>
```

```
<equation id='eqn15'>
  <title>Simple Arithmetic</title>
  11+27=38
</equation>
```

This equation will display as:

**Simple Arithmetic**

```
11+27=38
```

(5.1)

You could also write this equation using MathML:

```
<equation id="eqn22">
  <m:math>
    <m:mn>11</m:mn><m:mo>+</m:mo><m:mn>27</m:mn><m:mo>=</m:mo><m:mn>38</m:mn>
  </m:math>
</equation>
```

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.4.6 Definition

The definition tag is used to define a word in a CNXML document. It has a required id attribute and three children: term (Section 5.3.2.3.2.2: Term), meaning and example (Section 5.4.1: Example). How to use definition is a little confusing, so don’t forget to check out Example 5.22.

The first child tag should be term (Section 5.3.2.3.2.2: Term) which contains the word/phrase being defined. It is then followed by a meaning tag which is followed by any number of example (Section 5.4.1: Example)s. This process repeats for all meanings.

Example 5.22

<definition id='tbonedef'>
	<term>T-Bone</term>
	<meaning>
	"The T-bone steak is cut between 1 and 3 inches thick and comes from the center section of the short loin. This steak is characterized by its T-shape bone, has a fine-grained shell and a small tenderloin eye,"
	<cite>http://www.chophousecalgary.com/steak.html</cite>.
	</meaning>
	<example id='tboneexam'>
		<figure id='tbonefig'>
			<title>T-Bone Steak</title>
			<media id="image-example" display="block" alt="A T-bone Steak.">
				<image mime-type='image/jpeg' src='tbone.jpg'/>
			</media>
		</figure>
	</example>
</definition>

Definition 5.1: T-Bone
"The T-bone steak is cut between 1 and 3 inches thick and comes from the center section of the short loin. This steak is characterized by its T-shape bone, has a fine-grained shell and a small tenderloin eye," http://www.chophousecalgary.com/steak.html.

Example

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.4.7 Rule

The rule tag is used to insert a rule, such as a theorem, axiom, or rule of thumb, into a cnxml document. It has two attributes:

- \texttt{id} - required, unique ID
- \texttt{type} - required, specifies the type of rule (e.g. theorem, axiom, rule of thumb, etc.)

It may also have an optional title (Section 5.3.2.2: Title) and it must have one or more statement (Section 5.4.7.1: Statement) tags and zero or more proof (Section 5.4.7.2: Proof) or example (Section 5.4.1: Example) tags.
5.4.7.1 Statement

The statement tag is used inside a rule tag and defines the statement of the rule. It has an optional id attribute, which, like all IDs, must be unique. It also has two children, para (Section 5.3.2.3.1.1: Para) and equation (Section 5.4.5: Equation).

5.4.7.2 Proof

The proof tag is used inside the rule tag and marks the proof of the rule. It has an optional id attribute and may contain another rule, para (Section 5.3.2.3.1.1: Para), equation (Section 5.4.5: Equation), figure (Section 5.4.2: Figure), or list tag.

5.4.7.3 Using rule

Example 5.23

```
<rule id='murph' type='law'>
  <title>Murphy's Law</title>
  <statement>
    <para id='murphp1'>
      If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.
    </para>
  </statement>
  <proof>
    <para id='murphp2'>
      Edward A. Murphy, Jr. was one of the engineers on the rocket-sled experiments that were done by the U.S. Air Force in 1949 to test human acceleration tolerances (USAF project MX981). One experiment involved a set of 16 accelerometers mounted to different parts of the subject's body. There were two ways each sensor could be glued to its mount, and somebody methodically installed all 16 the wrong way around. Murphy then made the original form of his pronouncement, which the test subject (Major John Paul Stapp) quoted at a news conference a few days later
      <cite>http://www.lylemariam.com/murphy.htm</cite>.
    </para>
  </proof>
</rule>
```
Law 5.1: Murphy’s Law
If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.

Proof:
Edward A. Murphy, Jr. was one of the engineers on the rocket-sled experiments that were done by the U.S. Air Force in 1949 to test human acceleration tolerances (USAF project MX981). One experiment involved a set of 16 accelerometers mounted to different parts of the subject’s body. There were two ways each sensor could be glued to its mount, and somebody methodically installed all 16 the wrong way around. Murphy then made the original form of his pronouncement, which the test subject (Major John Paul Stapp) quoted at a news conference a few days later http://www.lylemarium.com/murphy.htm.

5.4.8 Finishing Remarks
Thanks for making it through another tutorial. I’m sure that you still want to know more so here’s a link to The Advanced CNXML (Section 5.5).

5.5 The Advanced CNXML

5.5.1 Code
The code tag is used to insert example computer output/input as either inline text within a paragraph or as a block of text. To see which tags it may contain or be inside, consult the CNXML Spec. The code tag has a display attribute with two possible values.

- inline (default) - used to specify code that is inline.
- block - used to specify code that should be in a separate block of text.

Example 5.24

```xml
<para id='copy'>
In a unix terminal the command to copy a file is <code display='inline'>cp original copy</code>.
</para>
```

This content is available online at <http://cnx.org/content/m9007/2.24/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
In a unix terminal the command to copy a file is `cp original copy`

### 5.5.2 Exercise

The `exercise` tag provides a tag for authors to add practice problems into their documents. The `exercise` tag has a required `id` attribute and has two child tags, `problem` and `solution`.

To create more complex answers, such as multiple-choice, multiple-response, ordered-response, and text-response questions, QML (Questions Markup Language) may be used in place of the `problem` and `solution` tags. For more information, please see the information about QML.\(^\text{18}\)

**Example 5.25**

```xml
<exercise id='grilltest'>
  <problem>
    <para id='grilltestp1'>
      For food safety, a steak should be cooked to a minimum temperature of what?
    </para>
  </problem>
  <solution>
    <para id='sol1p1'>
      160° F or until the juices run clear and the meat is no longer pink.
    </para>
  </solution>
</exercise>
```

**Problem**

For food safety, a steak should be cooked to a minimum temperature of what?

**Solution**

160° F or until the juices run clear and the meat is no longer pink

\(^\text{18}\)http://cnx.rice.edu/qml/intro/qml.xml

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.5.3 CALS Table

CNXML uses the industry standard CALS Table Model\(^{19}\) for including tables into CNXML documents. Provided below is a brief description of the CALS tags, their attributes, and children (along with a helpful example (Table 5.1: Steak Cooking Temperatures)). For a more complete description of the CALS Table consult the CALS Table Spec\(^ {20}\).

5.5.3.1 table

The `table` tag marks the beginning of a table. It has an optional first child of title (Section 5.3.2.2: Title) and must contain one or more `tgroup` (p. 232) tags. The `table` tag also has many attributes, to find out more information consult the CALS Table Spec\(^ {21}\).

5.5.3.2 tgroup

The `tgroup` tag marks the beginning of a new portion of a table (p. 232). It has a required attribute `cols` which is the number of columns in the `tgroup`. Its children tags are zero, one, or more `colspec` (p. 232) or `spanspec` (p. 232), zero or one `thead` (p. 232) or `tfoot` (p. 233), and one `tbody` (p. 233) tag.

5.5.3.3 colspec

The `colspec` tag is an empty tag that specifies the column of a table (p. 232) or `entrytbl` (p. 233). The names and numbers specified as attributes are used for referencing by other tags.

5.5.3.4 spanspec

The `spanspec` tag is an empty tag that identifies a horizontal span of columns and associated attributes that can subsequently be referenced by its spanname for repeated use in `entry` (p. 233) or `entrytbl` (p. 233) in different rows (p. 233).

5.5.3.5 thead

The `thead` tag identifies the heading row (p. 233) of a `tgroup` (p. 232) or `entrytbl` (p. 233). The `thead` tag can have zero, one, or more `colspec` (p. 232) tags and one or more `row` (p. 233).

\(^{19}\)http://www.oasis-open.org/specs/a502.htm
\(^{20}\)http://www.oasis-open.org/specs/a502.htm
\(^{21}\)http://www.oasis-open.org/specs/a502.htm

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.5.3.6 tfoot

The tfoot tag identifies the rows (p. 233) of footer information that are displayed after the tbody (p. 233). The tfoot tag can have zero, one, or more colspec (p. 232) tags and one or more row (p. 233).

5.5.3.7 tbody

The tbody tag identifies the body of a tgroup (p. 232) or entrytbl (p. 233). The tbody tag must have one or more row (p. 233) tags.

5.5.3.8 row

The row tag identifies the row of information in a thead (p. 232), tbody (p. 233), or tfoot (p. 233). The row tag must have one or more entry (p. 233) or entrytbl (p. 233).

5.5.3.9 entrytbl

The entrytbl tag takes the place of an entry (p. 233), but fits into a single row (p. 233) of tbody (p. 233) in a tgroup (p. 232). The content model is the same as that of a tgroup (p. 232) except that tfoot (p. 233) is omitted and entrytbl is self-excluding. Its children tags are zero, one, or more colspec (p. 232) or spanspec (p. 232), zero or one thead (p. 232) or tfoot (p. 233), and one tbody (p. 233) tag.

5.5.3.10 entry

The entry tag identifies an entry in a row (p. 233). The entry tag contains ASCII text and zero, one, or many cite (Section 5.3.2.3.2.5: Cite), term (Section 5.3.2.3.2.2: Term), cnxn, link, code, emphasis (Section 5.3.2.3.2.1: Emphasis), or media (p. 221).

5.5.3.11 Using CALS Tables

It might sound a little confusing but I think that the best way to understand a table is to look at Example 5.26. For more information, consult the CALS Table Spec\textsuperscript{22} or the CNXML Spec\textsuperscript{23}.

Example 5.26

\begin{verbatim}
<table id='grilltemp' frame='all'>
<title>Steak Cooking Temperatures</title>
\end{verbatim}

\textsuperscript{22}http://www.oasis-open.org/specs/a502.htm
\textsuperscript{23}http://cnx.rice.edu/technology/cnxml/0.5/spec

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
<table>
<thead>
<tr>
<th>Temperature (°F)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>140</td>
<td>Rare</td>
</tr>
<tr>
<td>150</td>
<td>Medium Rare</td>
</tr>
<tr>
<td>160</td>
<td>Medium</td>
</tr>
<tr>
<td>165</td>
<td>Medium Well</td>
</tr>
<tr>
<td>170</td>
<td>Well</td>
</tr>
</tbody>
</table>
Steak Cooking Temperatures

<table>
<thead>
<tr>
<th>Temperature (°F)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>140</td>
<td>Rare</td>
</tr>
<tr>
<td>150</td>
<td>Medium Rare</td>
</tr>
<tr>
<td>160</td>
<td>Medium</td>
</tr>
<tr>
<td>165</td>
<td>Medium Well</td>
</tr>
<tr>
<td>170</td>
<td>Well</td>
</tr>
</tbody>
</table>

Table 5.1

5.5.4 Conclusions

This concludes the CNXML tutorial.

5.6 CNXML Reference Extensions

5.6.1 Introduction

As an author/editor, you will often times need a way to include additional information in a document that does not actually appear in the flow of text. This information may include a glossary, and bibliographic references. There are many ways to include this type of information, but for our purposes, we have chosen to create a new CNXML tag named glossary, and have chosen to use an xml language called bibteXML for references. The two are described below. I have also included the glossary and bibteXML file examples in the source of this document. Scroll to the bottom of the page to see how these examples would be rendered.

5.6.1.1 BibteXML

"BibteXML is a bibliography DTD for XML that expresses the content model of BibTeX, the bibliographic system for use with LaTeX. It provides conversion tools for tagging your bibliographic data in XML, or export it to HTML or native BibTeX syntax, saving typing time." (http://freshmeat.net/projects/bibtexml/?topic_id=87)

24This content is available online at <http://cnx.org/content/m11215/1.8/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
In plain language, this means that bibtexml is an XML version of the popular and widely accepted latex extension bibtex. One can markup references in their document using semantic tags such as author and editor. More info will be provided below.

5.6.1.2 The ‘Glossary’ Tag

Often in textbooks there will be a list of definitions included at the end of the book. In the same way, the glossary tag will contain a list of definitions that will be included at the end of a module. One can link to these definitions using the term tag (see Example 5.28 (Linking to Definitions in a Glossary)).

5.6.2 Including a Glossary

It is very easy to include a glossary in your CNXML document. In the Basic CNXML Tutorial (Section 5.3) it is stated that the structure usually resembles the following:

Document

• name
• metadata(optional)
• content

When one wishes to add a glossary the structure will change to match the following:

Document

• name
• metadata(optional)
• content
• glossary

Inside of the glossary tag one can add as many definitions as one wishes. For more information on the definition tag, see the CNXML 0.5 specification25.

Example 5.27: Glossary Example

Following is an example of the code necessary to add a glossary with one definition.

\[
\begin{align*}
<\text{glossary}> \\
<\text{definition id='quardef'>} \\
<\text{term}>quarter</text{term}> \\
<\text{meaning}> <text{name}>Meaning Name</text{name}> One fourth of something.</text{meaning}> \\
<\text{example id='def'}}> \\
<\text{para id='par'}}> \\
\end{align*}
\]

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
"He cut the pie into quarters and gave all four people a piece."
</para>
</example>
<meaning>25 cents, a quarter of a dollar.</meaning>
<example id='def2'>
<para id='par2'>
"The drink cost a quarter."
</para>
</example>
<example id='def3'>
<para id='par3'>
"She picked up a roll of quarters so that she could do laundry."
</para>
</example>
</definition>
</glossary>

Example 5.28: Linking to Definitions in a Glossary
Often, one will need to refer to a definition in the glossary. To do this, one can use the term tag. By putting the target-id attribute in the term tag, one can link to a definition. Simply set the value of the target-id attribute to the id of the definition in the glossary, and that term will automatically become a link to the definition in the glossary. Shown below is an example of the term tag being used to link to the definition in the definition example (Example 5.27: Glossary Example):

<term target-id='quardef'>quarter</term>

5.6.3 Including BibteXML
It is very easy to include a bibteXML reference section in your CNXML document. In the Basic CNXML Tutorial\(^2\) it is stated that the structure usually resembles the following:

**Document**

- name

\(^2\)"Themes" <http://cnx.org/content/m0000/latest/>
• metadata (optional)
• content

When one wishes to add a bibteXML reference section the structure will change to match the following:

**Document**

• name
• metadata (optional)
• content
• glossary
• file

**NOTE**: It is possible to include either a glossary or a bibteXML file or both. The only restriction is that if you include both the glossary must precede the bibteXML file.

The file tag is the root tag of the bibteXML language. Inside of the file tag one will add other tags that correspond to the different type of bibliographical references. An example of a bibliography is given below:

**Example 5.29: BibteXML Example**

```xml
<bib:file>
  <bib:entry id="esbensen">
    <bib:book>
      <bib:author>Kim Esbensen; Tonje Midtgaard; Suzanne Schonkopf</bib:author>
      <bib:title>Multivariate Analysis in Practice</bib:title>
      <bib:publisher>Camo AS</bib:publisher>
      <bib:year>1994</bib:year>
      <bib:address>Trondheim</bib:address>
    </bib:book>
  </bib:entry>

  <bib:entry id="martens.nes">
    <bib:book>
      <bib:author>Harald Martens; Tormod Nas</bib:author>
      <bib:title>Multivariate Calibration</bib:title>
      <bib:publisher>John Wiley &amp; Sons Ltd.</bib:publisher>
      <bib:year>1989</bib:year>
      <bib:address>Chichester</bib:address>
    </bib:book>
  </bib:entry>
</bib:file>
```

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
NOTE: Each tag in the example code begins with a namespace prefix. In the case of bibteXML, the prefix is bib:.

Example 5.30: Linking to Bibliography
You will want to refer to an entry in the bibliography. To do this, one can use the cite tag. By putting the target-id attribute in the cite tag, one can link to a bibliographic entry. Simply set the value of the target-id attribute to the id of the bib:entry, and that reference will automatically become a link to the bibliographic entry. Shown below is an example of the cite tag being used to link to the bibliography in Example 5.29 (BibteXML Example):

```xml
<cite target-id='esbensen'>Multivariate Analysis in Practice</cite>
```

5.6.3.1 BibteXML Tags
By looking at Example 5.29 (BibteXML Example) one can see the types of tags that are available for use in bibteXML. Below I will attempt to give a brief explanation on the use of some of the bibteXML tags. For more information and the complete specification please see BibteXML Homepage27.

5.6.3.1.1 File
The file tag is the root tag of the bibteXML language. It denotes the beginning of the bibliography.

Children
The file tag must contain one or more entry (Section 5.6.3.1.2: Entry) tags.

5.6.3.1.2 Entry
The entry tag denotes the beginning of an individual bibliographical reference.

Children
The entry must contain one of the following containers:
• article
• book
• booklet
• manual
• techreport
• mastersthesis
• phdthesis
• inbook
• incollection
• proceedings
• inproceedings
• conference
• unpublished
• misc

5.6.3.1.3 Containers

Each of the possible children of entry (article, book, booklet, etc.) are containers for metadata on that specific type of entry.

Children

Every container tag must contain a different combination of bibteXML metadata tags (Section 5.6.3.1.4: BibteXML Metadata Tags). For more information on exactly which children a specific container may possess, please see the BibteXMLHomepage\(^{28}\).

5.6.3.1.4 BibteXML Metadata Tags

Every child listed as a possible child of entry (Section 5.6.3.1.2: Entry) can contain metadata tags. These metadata tags are now listed.

Metadata Tag List

• address
• author
• booktitle
• chapter
• edition
• editor
• howpublished
• institution
• journal

\(^{28}\)http://bibtexml.sourceforge.net/
5.7 Examples

5.7.1 Grilling a Good Steak

I have eaten many steaks in my life and none have been more satisfying than the backyard-grill cooked steak. Maybe this is because of the relaxing nature of drinking a beer, being outside, and lounging that accompanies the grilling procedure.

**WARNING:** Excessive drinking or fun may result in overcooked or burned steak.

Maybe it is because of the aroma of the grill and the beef perfectly seasoned to your taste. Either way, this module shows how a good steak can be prepared.

Before we begin to cook I have compiled a list of ingredients.

**Ingredients**

- Salt
- Fresh ground pepper
- Lime
- Beer
- Chili powder
- T-Bone steak

To ensure the best flavor possible, it is necessary to marinate the beef first. A steak marinates when left to sit in marinade, or prepared sauce, where it will absorb the flavor of the ingredients. Marinating may take as little as 15 minutes or as long as 6 hours and should always be done in the refrigerator and not at room temperature.

**Instructions**

---

29This content is available online at <http://cnx.org/content/m10275/2.6/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
1. pour beer into large bowl
2. add chili powder to taste
3. squeeze half lime into beer marinade
4. place steak in beer, let soak for 30 minutes

I’ll be adding to this module in The Intermediate CNXML (Section 5.4) which focuses on more advanced CNXML tags. For more marinades see the Angus Beef website\(^\text{30}\). Finally, a good resource is the *Steak Lover’s Cookbook – William Rice*.

### 5.7.2 Grilling a Better Steak\(^\text{31}\)

I have eaten many steaks in my life and none have been more satisfying than the backyard-grill cooked steak. Maybe this is because of the relaxing nature of drinking a beer, being outside, and lounging that accompanies the grilling procedure. Maybe it is because of the aroma of the grill and the beef perfectly seasoned to your taste. Either way, this module shows how a good steak can be prepared.

#### 5.7.2.1 Ingredients

Before we begin to cook I have compiled a list of ingredients.

**Ingredients**

- Salt
- Fresh ground pepper
- Lime
- Beer
- Chili powder
- T-Bone steak

#### 5.7.2.2 Marinade

To ensure the best flavor possible, it is necessary to marinate the beef. A steak **marinates** when left to sit in **marinade**, or prepared sauce, where it will absorb the flavor of the ingredients. Marinating may take as little as 15 minutes or as long as 6 hours and should **always** be done in the refrigerator and **not** at room temperature.

**Marinade**

1. pour beer into large bowl
2. add chili powder to taste
3. squeeze half lime into beer marinade

\(^{30}\)http://www.2eatcab.com

\(^{31}\)This content is available online at <http://cnx.org/content/m10278/2.9/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
4. place steak in beer, let soak for 30 minutes

Figure 5.2: Upon successful completion of these modules, you should be able to grill a steak that looks just as good!

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
CHAPTER 5. CNXML TUTORIAL

Steaks

(a) T-Bone

(b) New York Strip

Figure 5.3: Upon successful completion of these modules, you should be able to grill a steak that looks just as good!
Figure 5.4: Upon successful completion of these modules, you should be able to grill a steak that looks just as good!
How to grill the steak will be covered in The Advanced CNXML (Section 5.5). For more marinades see the Angus Beef website\(^{32}\). Finally, a good resource is the *Steak Lover’s Cookbook* – William Rice; Paperback.

### 5.7.3 Grilling the Best Steak\(^{33}\)

I have eaten many steaks in my life and none have been more satisfying than the backyard-grill cooked steak. Maybe this is because of the relaxing nature of drinking a beer, being outside, and lounging that accompanies the grilling procedure. Maybe it is because of the aroma of the grill and the beef perfectly seasoned to your taste. Either way, this module shows how a good steak can be prepared.

---

**Steaks**

![T-Bone and New York Strip Steaks](image)

(a) T-Bone  
(b) New York Strip

**Figure 5.5:** Upon successful completion of these modules, you should be able to grill a steak that looks just as good!

---

\(^{32}\)http://www.2eatcab.com  
\(^{33}\)This content is available online at <http://cnx.org/content/m10281/2.7/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
5.7.3.1 Ingredients

Before we begin to cook I have compiled a list of ingredients.

Ingredients

- Salt
- Fresh ground pepper
- Lime
- Beer
- Chili powder
- T-Bone

Definition 5.3: T-Bone

"The T-bone steak is cut between 1 and 3 inches thick and comes from the center section of the short loin. This steak is characterized by its T-shape bone, has a fine-grained shell and a small tenderloin eye," [http://www.chophousecalgary.com/steak.html](http://www.chophousecalgary.com/steak.html).

5.7.3.2 Marinade

To ensure the best flavor possible, it is necessary to marinate the beef. A steak marinates when left to sit in a prepared sauce, or marinade, where it will absorb the flavors of the ingredients. Marinating may take as little as 15 minutes or as long as 6 hours and should always be done in the refrigerator and not at room temperature.

Marinade

1. Pour beer into large bowl
2. Add chili powder to taste
3. Squeeze half lime into beer marinade
4. Place steak in beer, let soak for 30 minutes
5. Before grilling rub salt and pepper onto steak

5.7.3.3 Grilling

Grilling is pretty easy. After having heated the coals or igniting the grill, start cooking the meat. I would recommend periodically checking the meat and when you start to see it being cooked on top, flip it over. Then, wait until fully cooked. Below you will find a table of cooking temperatures. Please note the safety warning at the bottom.

Available for free at Connexions [http://cnx.org/content/col10151/1.27]
CHAPTER 5. CNXML TUTORIAL

<table>
<thead>
<tr>
<th>Temperature(F)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>140</td>
<td>Rare</td>
</tr>
<tr>
<td>150</td>
<td>Medium Rare</td>
</tr>
<tr>
<td>160</td>
<td>Medium</td>
</tr>
<tr>
<td>165</td>
<td>Medium Well</td>
</tr>
<tr>
<td>170</td>
<td>Well</td>
</tr>
</tbody>
</table>

Table 5.2

Remember that for safety’s sake, always cook your steak to 160 F or until meat is no longer pink.

5.7.3.4 Rounding Off the Experience

The experience of grilling a steak in your own back yard is part of what makes the home cooked steak so enjoyable. It is necessary to cook in the evening as it is getting cool and to enjoy your beverage of choice. Finally, one of the best ways to enjoy a steak is in the company of your friends.

To make sure that you were paying attention to my tutorial, I’ve included a one question exam:

**Exercise 5.7.3.1** *(Solution on p. 252.*)

For food safety, a steak should be cooked to a minimum temperature of what?

For more marinades see the Angus Beef website[^34] . Finally, a good resource is the *Steak Lover’s Cookbook – William Rice; Paperback.*

5.7.4 Example module for use of cnxml/mathml tags[^35]

A paragraph is the place for text. You can also include **vocabulary terms**.

**NOTE:** Do not use `<emphasis>` for vocabulary terms.

There are two ways to present definitions - using the glossary to define the term as above or using the `definition` tag within the text.

**Definition 5.4: problem**

1. a question raised for inquiry, consideration, or solution; a proposition in mathematics or physics stating something to be done

[^34]: http://www.2eatcab.com
[^35]: This content is available online at <http://cnx.org/content/m11216/1.3/>.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
2. an intricate unsettled question; a source of perplexity, distress, or vexation; difficulty in understanding or accepting
   See Also: enigma, puzzle

5.7.4.1 Section Name

A document can have sections, however they are not required.

5.7.4.1.1 Subsection Name

Sections can have subsections. You can include quotes in paragraphs.

_The seasons alter: hoary-headed frosts Fall in the fresh lap of the crimson rose, And on old Hiems thin and icy crown An odorous chaplet of sweet summer buds Is, as in mockery, set. The spring, the summer, The childing autumn, angry winter, change Their wonted liveries, and the mazed world, By their increase, now knows not which is which._ - _William Shakespeare; A Midsummer Night’s Dream_?

You can also denote words from another language, such as biological genus and species, *E.coli*, or the southern German greeting, *GrüßGott*.

Paragraphs can contain many other tags such as lists and figures. Figure 5.6 (Optional Figure Name) shows how a figure will display in our system.

**Optional Figure Name**

![Diagram](image)

**Figure 5.6:** The caption of the figure would be here.

There are four different options for list. One type of list is a named-item list.

**List name**

- **first name** - description of first item

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
See the CNXML-0.5 Spec\textsuperscript{36} for more information regarding list types.

**Example 5.31: Optional Name of Example**

Here is where you would put an example that relates to what the previous paragraphs were discussing. In an example, you can include any tags that are allowed in any other paragraph including tables (see Table 5.3: Temperatures in 5 cities on 11/16/2002).

**Temperatures in 5 cities on 11/16/2002**

<table>
<thead>
<tr>
<th>City</th>
<th>Degrees Fahrenheit</th>
<th>Degrees Centigrade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Houston</td>
<td>54</td>
<td>12.22</td>
</tr>
<tr>
<td>Chicago</td>
<td>37</td>
<td>2.78</td>
</tr>
<tr>
<td>Minneapolis</td>
<td>31</td>
<td>-0.56</td>
</tr>
<tr>
<td>Miami</td>
<td>78</td>
<td>25.56</td>
</tr>
<tr>
<td>Phoenix</td>
<td>70</td>
<td>21.11</td>
</tr>
<tr>
<td><strong>Mean</strong></td>
<td><strong>54.000</strong></td>
<td><strong>12.22</strong></td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td><strong>54.000</strong></td>
<td><strong>12.22</strong></td>
</tr>
<tr>
<td><strong>Variance</strong></td>
<td><strong>330.00</strong></td>
<td><strong>18.166</strong></td>
</tr>
<tr>
<td><strong>SD</strong></td>
<td><strong>101.852</strong></td>
<td><strong>10.092</strong></td>
</tr>
</tbody>
</table>

Table 5.3

In the connexions system, it is important to use `frame="all"` with the `table` tag and `colsep="1"` `rowsep="1"` with the `tbody` tag to properly display the table in the print system.

If you would like a block of material to display exactly as you type it, you can use `<code type='block'>` as you would when discussing computer programming.

```latex
\begin{verbatim}
\texttt{syms t}
\texttt{laplace(exp(t))}
\texttt{ans = 1/(s-1)}
\texttt{laplace(t*exp(-t))}
\texttt{ans = 1/(s+1)^2}
\end{verbatim}
```

\textsuperscript{36}http://cnx.rice.edu/cnxml/0.5/spec/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
You can also include math in your document. Math can be displayed in three ways in our system: inline, block, or numbered equations. Here are examples of an inline variable, $x$, and an inline equation, $0 < x < \pi$. Math that you want set apart from the text can be numbered when in an equation tag or simply set apart.

$$\lim_{n \to \infty} \| x_n - x \| = 0$$

optional equation name

$$P(f) = \int_{-\infty}^{\infty} p(t) e^{-i2\pi ft} \, dt = \int_{0}^{\Delta} e^{-i2\pi ft} \, dt = \frac{1}{-(i2\pi f)} \left( e^{-i2\pi f \Delta} - 1 \right) \quad (5.2)$$

See our discussion of Content MathML (Section 4.3.1.6.1) for a more basic discussion of math. See also our MathML Extensions page\textsuperscript{37} for more \texttt{csymbol} options.

**Theorem 5.1:** Pythagorean Theorem
For a right triangle with legs $a$ and $b$ and hypotenuse $c$,

$$a^2 + b^2 = c^2$$

**Proof:**
The proof would go here.

**Exercise 5.7.4.1**
What is a composite number?

\textsuperscript{37}http://bunker.ece.rice.edu:8080/mntb/wikis/mathml/CSymbol

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Solutions to Exercises in Chapter 5

Solution to Exercise 5.7.3.1 (p. 248)
160 F or until the juices run clear and the meat is no longer pink

Solution to Exercise 5.7.4.1 (p. 251)
A composite number $n$ is a positive integer, $n > 1$, which is not prime.
Glossary

P  problem
1. a question raised for inquiry, consideration, or solution; a proposition in mathematics or physics stating something to be done
2. an intricate unsettled question; a source of perplexity, distress, or vexation; difficulty in understanding or accepting

Q  quarter
1. Meaning Name One fourth of something.
   Example: "He cut the pie into quarters and gave all four people a piece."
2. 25 cents, a quarter of a dollar.
   Example: "The drink cost a quarter."
   Example: "She picked up a roll of quarters so that she could do laundry."

T  T-Bone
"The T-bone steak is cut between 1 and 3 inches thick and comes from the center section of the short loin. This steak is characterized by its T-shape bone, has a fine-grained shell and a small tenderloin eye,"
http://www.chophousecalgary.com/steak.html

T-Bone
"The T-bone steak is cut between 1 and 3 inches thick and comes from the center section of the short loin. This steak is characterized by its T-shape bone, has a fine-grained shell and a small tenderloin eye,"
http://www.chophousecalgary.com/steak.html

Example:

   T-Bone Steak

This is an unsupported media type. To view, please see http://cnx.org/content/m9006/latest/

Figure 5.1

V  vocabulary term
A word or phrase being defined in the text. A more complete definition with an example could be placed here in the glossary.

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Bibliography


Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. Ex. apples, § 1.1 (1) Terms are referenced by the page they appear on. Ex. apples, 1

A  account, § 1.2(6)
adaptation, 12
affiliation, § 2.3(24)
affiliation lenses, 25
Alt text, 80
alternate, § 4.3.1.2(78)
annotation, § 2.1(11)
apply, § 4.3.1.6.1(131)
attribute, § 2.2(20), 106, § 5.1(205), 206
audio, § 4.3.1.2(78)
authentic, 75
author, § 1.1(1), § 4.3.4.1(173)
authoring, § 3.2(35)
Authors, 173
cite, § 2.2(20)
cn, § 4.3.1.6.1(131), 132
cnxml, § 4.2.1(51), § 4.2.3(68), § 4.3.1.3(105), 106, § 4.3.1.5(126), § 5.3(210), § 5.4(221), § 5.7.2(242), § 5.7.3(246), § 5.7.4(248)
cnxml example, § 5.7.1(241)
codeblock, § 5.4(221)
codeline, § 5.4(221)
collection, § 2.4(26), § 3.2(35), § 4.2.1(51), § 4.3.6.2(181), § 4.4.1(183), § 4.4.2(185), § 4.4.3(186), § 4.4.5(196)
collection composer, § 4.4.3(186)
collection source export ZIP, 23
Collection XML, 22
collections, 1, 15, § 4.4.4(195)
collxml, § 4.2.1(51)
comments, § 2.3(24), 24
conexions, § 3.2(35), § 4.3.1.3(105), § 4.3.1.6.3(151), § 5.3(210)
Connexions Markup Language, 106
conexions project, § 1.1(1)
content, § 2.2(20), § 4.1(49), § 4.2.3(68), § 4.3.1.3(105), § 4.3.1.6.2(139), § 4.4.1(183), § 5.3(210)
content mathml, § 4.3.1.6.1(131)
content repository, 11, 38
Content Section, 210
contents, § 3.3.2(39), § 4.3.1.1(74)
context, 11, 18, 19
copy, § 4.5(196)

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
copyright holder, § 4.3.4.1(173)
Copyright Holders, 173
course, § 4.4.3(186)
course composer, § 4.4.3(186)
create, § 3.3.1(37)
csymbol, § 4.3.1.6.1(131), 132

dashboard, § 3.1(33), 33
definition, § 4.3.1.4(110), § 5.5(230), § 5.6(235)
delete, § 3.3.1(37)
derivative, § 4.5(196)
derive, § 4.5(196)
differentiation, § 4.3.1.6.1(131)
discard, § 4.3.6.1(179)
document, § 4.3.1.3(105), § 5.3(210)
documentation, § 3.1(33), § 4.3.1.6.3(151), § 4.6.1(201), § 4.6.2(202), § 4.6.3(203)
downloads, § 2.2(20)
DTD, 207, § 5.2(208)
dynamic reference, 85

edit, § 4.3.1.1(74), § 4.3.2.1(168), § 4.3.3.1(172), § 4.3.4.1(173), § 4.3.5.1(177), § 4.4.2(185), § 4.5(196)
edit in place, § 4.3.1.1(74)
editing, § 3.2(35), § 4.1(49), § 4.3.1.6.2(139), § 4.4.1(183)
editing interface, § 3.2(35)
editor, § 4.3.1.6.3(151), § 4.3.4.1(173)
editors, 173
elem, § 4.3.1.4(110), § 5.5(230)
embed tag, 100, 103
empty tag, 127, 206, 232
empty tags, 133
endorsement, § 2.3(24)
endorsement lenses, 25
enigma, 249
entity, § 5.1(205)
entity references, 207
equation, § 5.4(221)
error, § 1.3(8)

element, § 4.3.1.4(110), § 5.5(230)
examples, 14
exercise, § 4.3.1.4(110), § 5.5(230)
export, § 4.2.1(51)

F
featured links, § 2.4(26), § 4.3.5.1(177)
figure, § 5.4(221)
files, § 4.3.2.1(168)
flash, § 4.3.1.2(78)
full source editor, § 4.3.1.1(74)
function, § 4.3.1.6.1(131)

g
Gardenias, 107
general information, § 1.1(1)
glossary, § 5.6(235)
grilling, § 5.7.1(241), § 5.7.2(242), § 5.7.3(246)
group, § 4.3.1.4(110), § 4.4.3(186), § 5.5(230)
group members, § 3.3.4(44)

H
help, § 4.2.3(68)
home page, § 1.1(1)

I
I have read the Connexions Site
License and I agree to be bound by its
terms., 7
ID, 111
identifier, 132
IEEE, § 4.2.3(68)
iframe tag, 101
image, § 4.3.1.2(78)
images, § 4.2.3(68)
import, § 4.2.1(51), § 4.3.1.1(74)
Importer, § 4.2.3(68)
Importing, § 4.2.3(68)
inline, 83, 85
instructor, § 1.1(l)
integration, § 4.3.1.6.1(131)
issues, § 4.2.3(68)

J
java applet, § 4.3.1.2(78)

K
keywords, § 2.4(26)
INDEX

L
language, § 2.4(26)
l latex, § 4.2.1(51), § 4.2.3(68)
lenses, 1, 15, § 2.3(24), § 2.4(26), § 4.6.1(201), § 4.6.2(202), § 4.6.3(203)
links, § 4.3.5.1(177)
locate, § 2.1(11)
log in, § 1.1(1), § 1.2(6)
log out, § 1.2(6)
longdesc, 81

M
maintainer, § 4.3.4.1(173)
Maintainers, 173
marinade, 241, 242, 247
marinates, 241, 242, 247
markup, § 4.3.1.3(105), § 5.3(210)
markup language, 205
math, § 4.2.3(68), § 4.3.1.6.2(139)
mathematics, § 4.4.4(195)
mathml, § 4.3.1.6.1(131), § 4.3.1.6.3(151), § 5.7.4(248)
matrix, § 4.3.1.6.1(131)
mbox, § 4.2.3(68)
meaning, § 4.3.1.4(110), § 5.5(230)
media, § 5.4(221)
media subtype, 83
member, § 1.2(6)
member list, § 2.3(24)
member list lenses, 25
members, § 3.3.4(44)
meta-markup language, 205
metadata, § 2.2(20), 20, 20, § 4.1(49), § 4.3.3.1(172), § 4.4.1(183), § 4.4.2(185)
Metadata Section, 210
Microsoft, § 4.2.2(56)
module, § 4.2.1(51), § 4.3.1.1(74), § 4.3.2.1(168), § 4.3.3.1(172), § 4.3.4.1(173), § 4.3.5.1(177), § 4.3.6.1(179), § 4.3.6.2(181), § 4.4.3(186), § 4.5(196)
module source export ZIP, 22
Module XML, 22
modules, 1, 15, § 3.2(35), § 4.1(49), § 4.3.1.6.2(139)

movie, § 4.3.1.2(78)
mpeg, § 4.3.1.2(78)
multimedia, § 4.3.1.2(78)
multimedia object, § 4.3.1.2(78)
music, § 4.3.1.2(78)
my favorites, § 2.3(24), 25
MyCNX, 5, § 3.1(33), 33

N
n-ary operator, 134
namespace, § 4.3.1.3(105), § 5.2(208), § 5.3(210), 212
namespace prefix, 209
namespaces, 209
navigation, § 3.2(35)
nesting, § 5.1(205)
notation, § 4.4.4(195)
notes, § 2.1(11)

O
online, § 4.3.6.1(179), § 4.4.5(196)
openoffice, § 4.2.1(51)
operator, § 4.3.1.6.1(131)
organizational account, § 4.6.3(203)

P
paragraph, § 4.2.2(56)
param, 104
param tags, 103
parameters, § 4.4.4(195)
PDF, § 4.3.6.1(179)
personal workspace, § 3.3.1(37), § 3.3.2(39), § 3.3.3(44), § 3.3.4(44)
portlets, § 2.4(26)
prefix, § 4.3.1.6.1(131)
Prefix notation, 132
prerequisite material, 14
preview, § 4.3.6.1(179), § 4.4.5(196)
print, § 4.3.6.1(179), § 4.4.4(195)
problem, § 4.2.3(68), § 4.3.1.4(110), § 5.5(230), 248
problems, § 4.2.3(68)
proof, § 4.3.1.4(110), § 5.5(230)
properties, § 3.3.3(44)
publish, § 4.3.6.2(181)
puzzle, 249

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Attributions

Collection: Connexions Tutorial and Reference
Edited by: Connexions, Adan Galvan, Brent Hendricks, Mark Husband
URL: http://cnx.org/content/col10151/1.27/
License: http://creativecommons.org/licenses/by/3.0/

Module: "Introduction to Connexions"
Used here as: "Basic Site Layout"
By: Mark Husband
URL: http://cnx.org/content/m10884/2.28/
Pages: 1-6
Copyright: Mark Husband, Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "Create a Connexions Account"
Used here as: "Create an Account"
By: Connexions
URL: http://cnx.org/content/m37412/1.2/
Pages: 6-8
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "Reporting a Connexions Bug"
Used here as: "Report a Bug"
By: Mark Husband
URL: http://cnx.org/content/m11836/1.8/
Pages: 8-9
Copyright: Mark Husband, Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "Viewing Connexions Content"
Used here as: "Modules and Collections"
By: Connexions, Mark Husband
URL: http://cnx.org/content/m11837/1.15/
Pages: 11-20
Copyright: Connexions, Mark Husband
License: http://creativecommons.org/licenses/by/3.0/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
ATTRIBUTIONS

Module: "Viewing Metadata"
Used here as: "Metadata"
By: Connexions
URL: http://cnx.org/content/m37413/1.2/
Pages: 20-24
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "Using Lenses"
Used here as: "Lenses"
By: Connexions
URL: http://cnx.org/content/m37415/1.2/
Pages: 24-26
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "Finding Content"
By: Connexions
URL: http://cnx.org/content/m37430/1.2/
Pages: 26-31
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "MyCNX Dashboard"
Used here as: "MyCNX Tab"
By: Connexions, Mark Husband, Adan Galvan
URL: http://cnx.org/content/m10885/2.23/
Pages: 33-35
Copyright: Connexions, Mark Husband, Adan Galvan
License: http://creativecommons.org/licenses/by/3.0/

Module: "Navigating the Editing Interface"
By: Connexions
URL: http://cnx.org/content/m37463/1.2/
Pages: 35-36
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "Work Areas Overview"
Used here as: "Overview"
By: Connexions
URL: http://cnx.org/content/m37462/1.2/
Pages: 37-39
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Module: "Advanced CNXML using Edit-in-Place"
By: Elizabeth Gregory, Connexions
URL: http://cnx.org/content/m14395/1.5/
Pages: 110-126
Copyright: Elizabeth Gregory
License: http://creativecommons.org/licenses/by/3.0/
Based on: The Advanced CNXML
By: Ricardo Radaelli-Sanchez
URL: http://cnx.org/content/m9007/2.22/

Module: "CALS Table"
By: Elizabeth Gregory
URL: http://cnx.org/content/m14396/1.5/
Pages: 126-131
Copyright: Elizabeth Gregory
License: http://creativecommons.org/licenses/by/2.0/
Based on: The Advanced CNXML
By: Ricardo Radaelli-Sanchez
URL: http://cnx.org/content/m9007/2.22/

Module: "Content MathML"
By: Sarah Coppin, Brent Hendricks
URL: http://cnx.org/content/m9008/2.15/
Pages: 131-139
Copyright: Sarah Coppin, Brent Hendricks
License: http://creativecommons.org/licenses/by/1.0

Module: "MathML Editor: Manual"
Used here as: "Introduction to the MathML Editor"
By: Philip Schatz
URL: http://cnx.org/content/m24561/1.2/
Pages: 139-151
Copyright: Philip Schatz
License: http://creativecommons.org/licenses/by/3.0/

Module: "MathML Editor: The Basics"
By: Natalie Weber
URL: http://cnx.org/content/m26312/1.2/
Pages: 151-164
Copyright: Natalie Weber
License: http://creativecommons.org/licenses/by/3.0/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Module: "Previewing a Module"
Used here as: "Preview Module"
By: Connexions
URL: http://cnx.org/content/m38209/1.2/
Pages: 179-181
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "Publishing Content"
Used here as: "Publish"
By: Mark Husband, Connexions
URL: http://cnx.org/content/m13054/1.3/
Pages: 181-182
Copyright: Mark Husband
License: http://creativecommons.org/licenses/by/3.0/

Module: "Overview of Authoring Collections"
Used here as: "Create new collection"
By: Connexions
URL: http://cnx.org/content/m38767/1.2/
Pages: 183-185
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/
Based on: Overview of Authoring Modules
By: Connexions, Mark Husband, Adan Galvan
URL: http://cnx.org/content/m10887/2.31/

Module: "Editing Collection Metadata"
Used here as: "Collection Metadata"
By: Connexions
URL: http://cnx.org/content/m38768/1.1/
Pages: 185-186
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/
Based on: Editing Module Metadata
By: Connexions
URL: http://cnx.org/content/m38204/1.3/

Module: "How to Use the Collection Composer"
Used here as: "Collection Composer"
By: Connexions, Sarah Coppin, Mark Husband, Max Starkenburg
URL: http://cnx.org/content/m10288/2.26/
Pages: 186-195
Copyright: Connexions, Sarah Coppin, Mark Husband
License: http://creativecommons.org/licenses/by/3.0/

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Module: "Create an organizational account"
By: Connexions
URL: http://cnx.org/content/m43299/1.1/
Pages: 203-204
Copyright: Connexions
License: http://creativecommons.org/licenses/by/3.0/

Module: "XML Basics"
By: Connexions, Sarah Coppin, Brent Hendricks, Chuck Bearden
URL: http://cnx.org/content/m9002/2.24/
Pages: 205-208
Copyright: Connexions, Sarah Coppin, Brent Hendricks, Chuck Bearden
License: http://creativecommons.org/licenses/by/1.0

Module: "Combining XML Languages"
By: Sarah Coppin, Brent Hendricks, Connexions
URL: http://cnx.org/content/m10159/2.16/
Pages: 208-209
Copyright: Sarah Coppin, Brent Hendricks
License: http://creativecommons.org/licenses/by/1.0

Module: "The Basic CNXML"
By: Ricardo Radaelli-Sanchez, Brent Hendricks, Connexions
URL: http://cnx.org/content/m9000/2.36/
Pages: 210-220
Copyright: Ricardo Radaelli-Sanchez
License: http://creativecommons.org/licenses/by/1.0

Module: "The Intermediate CNXML"
By: Ricardo Radaelli-Sanchez, Connexions
URL: http://cnx.org/content/m9006/2.22/
Pages: 221-230
Copyright: Ricardo Radaelli-Sanchez
License: http://creativecommons.org/licenses/by/1.0

Module: "The Advanced CNXML"
By: Ricardo Radaelli-Sanchez, Connexions
URL: http://cnx.org/content/m9007/2.24/
Pages: 230-235
Copyright: Ricardo Radaelli-Sanchez
License: http://creativecommons.org/licenses/by/1.0

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
ATTRIBUTIONS

Module: "CNXML Reference Extensions"
By: Adan Galvan
URL: http://cnx.org/content/m11215/1.8/
Pages: 235-241
Copyright: Adan Galvan
License: http://creativecommons.org/licenses/by/1.0

Module: "Grilling a Good Steak"
By: Sarah Coppin, Ricardo Radaelli-Sanchez
URL: http://cnx.org/content/m10275/2.6/
Pages: 241-242
Copyright: Sarah Coppin, Ricardo Radaelli-Sanchez
License: http://creativecommons.org/licenses/by/1.0

Module: "Grilling a Better Steak"
By: Sarah Coppin, Ricardo Radaelli-Sanchez
URL: http://cnx.org/content/m10278/2.9/
Pages: 242-246
Copyright: Sarah Coppin, Ricardo Radaelli-Sanchez
License: http://creativecommons.org/licenses/by/1.0

Module: "Grilling the Best Steak"
By: Sarah Coppin
URL: http://cnx.org/content/m10281/2.7/
Pages: 246-248
Copyright: Sarah Coppin
License: http://creativecommons.org/licenses/by/1.0

Module: "Example module for use of cnxml/mathml tags"
By: Charlet Reedstrom
URL: http://cnx.org/content/m11216/1.3/
Pages: 248-251
Copyright: Charlet Reedstrom
License: http://creativecommons.org/licenses/by/1.0

Available for free at Connexions <http://cnx.org/content/col10151/1.27>
Connexions Tutorial and Reference
Tutorial and reference materials for getting started with Connexions

About OpenStax-CNX
Rhaptos is a web-based collaborative publishing system for educational material.