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:

Chuck Bearden
Connexions
Sarah Coppin
Adan Galvan
Elizabeth Gregory
Brent Hendricks
Mark Husband
Ricardo Radaelli-Sanchez
Charlet Reedstrom
Philip Schatz
Natalie Weber

Online:
< http://cnx.org/content/col10151/1.21/ >

C O N N E X I O N S

Rice University, Houston, Texas

This selection and arrangement of content as a collection is copyrighted by Connexions, Adan Galvan, Brent Hendricks, Mark Husband. It is licensed under the Creative Commons Attribution 1.0 license (http://creativecommons.org/licenses/by/1.0).
Collection structure revised: August 13, 2009
PDF generated: October 23, 2009
For copyright and attribution information for the modules contained in this collection, see p. 190.

Table of Contents

1 Connexions Overview
- **1.1** Introduction to Connexions ... 1
- **1.2** Reporting a Connexions Bug ... 8
- **1.3** Viewing Connexions Content ... 9
- Solutions ... ??

2 Working with Content
- **2.1** MyCNX and the Work Areas ... 13
- **2.2** How to Use the Collection Composer ... 28
- **2.3** Editing Modules ... 38
- **2.4** Adding Multimedia to Your Connexions Content ... 66
- **2.5** Derived Copy ... 85
- **2.6** Suggest Edits ... 87
- Solutions ... ??

3 XML and CNXML
- **3.1** Editing CNXML with Edit-In-Place ... 91
- **3.2** Editing XML with the Full Source Editor ... 116
- **3.3** Editing CNXML with the Full Source Editor ... 120
- Solutions ... ??

4 MathML Editor
- **4.1** Content MathML ... 145
- **4.2** Introduction to the MathML Editor ... 151
- **4.3** MathML Editor: The Basics ... 160
- **4.4** MathML Editor: Tutorials ... 170
- Solutions ... ??

5 Appendix
- **5.1** Grilling a Good Steak ... 175
- **5.2** Grilling a Better Steak ... 176
- **5.3** Grilling the Best Steak ... 180
- **5.4** Example module for use of cnxml/mathml tags ... 182
- Solutions ... 185

Glossary ... 186
Bibliography ... 187
Index ... 188
Attributions ... 190

Chapter 1

Connexions Overview

1.1 Introduction to Connexions[1]

1.1.1 Overview

Connexions is a web-based document creation and management system for education and research materials. There are two parts to Connexions: a Content Commons that contains these materials and the software tools necessary to create, manage, and access these materials.

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/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[2] .

There are three 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 courses, which are documents created by linking related modules together in a specific order.

Before you begin using the Connexions system, you should verify that you have all of the necessary software. Visit the Connexions Quick Start Guide[3] for a listing of the software that you will need and the links to help you download it.

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)

[1] This content is available online at <http://cnx.org/content/m10884/2.26/>.
[2] http://creativecommons.org/licenses/by/3.0/
[3] http://cnx.org/help

- Create Content (Section 1.1.2.7: Create Content)
- Spotlight Sidebar (Section 1.1.2.8: Spotlight Sidebar)
- Connexions News Sidebar
- Create a Connexions Account (Section 1.1.3: Account Requests)

These items also appear on the other Connexions web pages.

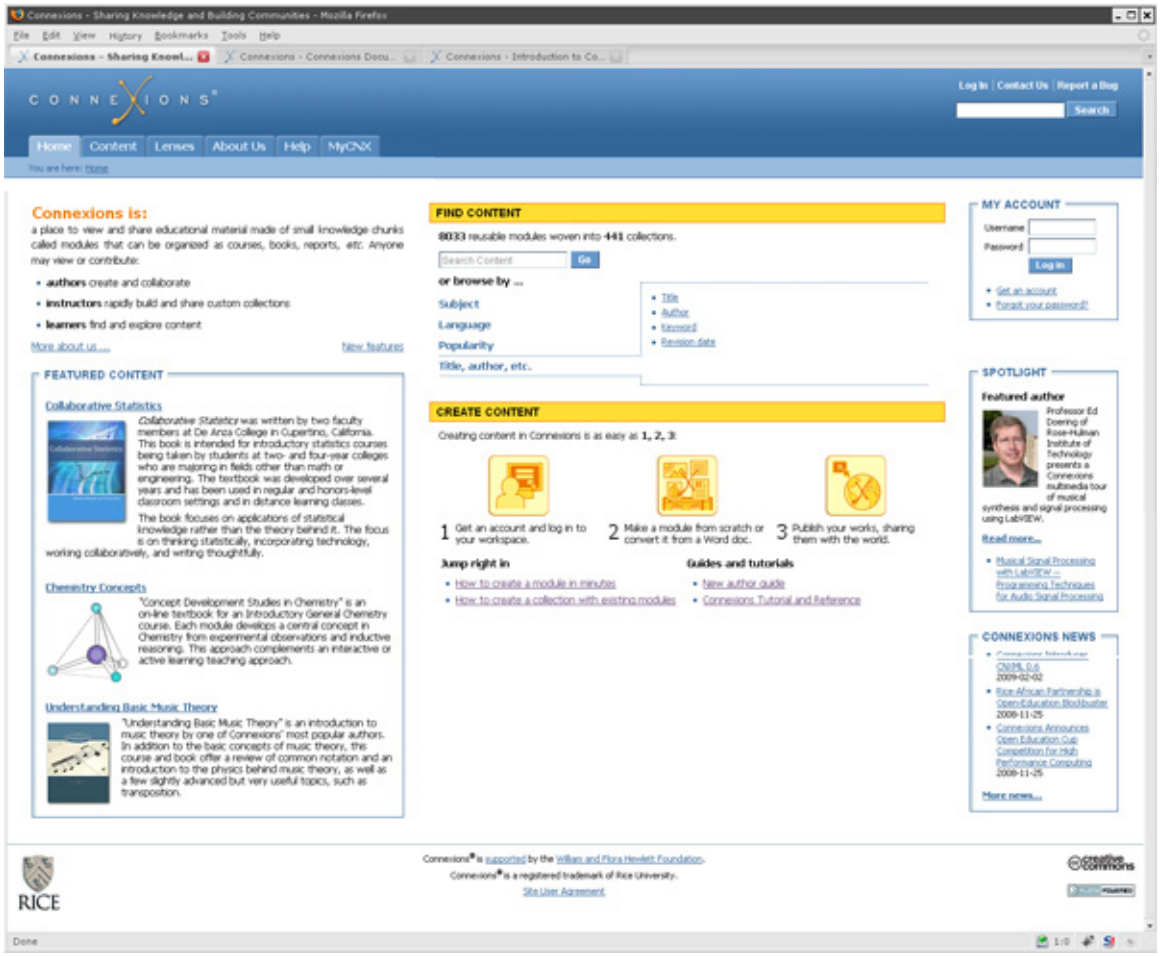


Figure 1.1: Connexions home page

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", "About Us", "Help", and "MyCNX". Clicking on these tabs display the following Connexions pages:

- Home[4] returns to the Connexions home page.
- Content[5] takes you to the Content landing page, where you can search for modules and collections of interest or browse the repository by subject, author, popularity and more.

[4] http://cnx.org/
[5] http://cnx.org/content/

- Lenses[6] is a portal to all user and organization-created lenses[7] .
- About[8] displays the "About" page that contains a general description of Connexions and links to more detailed information about Connexions, how it works, the people behind it, and a list of contacts.
- Help[9] displays the "Help" page that contains links to installation, reference, and other information to answer your questions about Connexions.
- MyCNX[10] displays your author homepage if you are signed in. This page includes links to your personal workspace, your workgroups, 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 "Home", "About", "Content", "Software", and "Help" 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. This feature is very useful for returning to the workgroup level page after working with the items within the workgroup.

Figure 1.2: Breadcrumb navigation bar

1.1.2.3 Search and Quick Links

You can search the Content Commons for a specific module or a course using the `Search` button and text box in the upper right of the Connexions home page. Type in a name, keyword, course 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 courses that match your entry. To view a module or course, click on its title.

Above the search box are three quick links for your convenience. These links vary, depending on whether you are logged into the system or not.

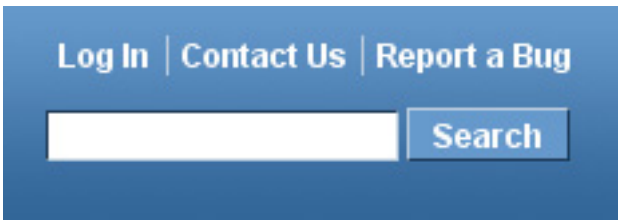


Figure 1.3: Personal toolbar before login

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):

[6] http://cnx.org/lenses/
[7] http://cnx.org/help/lens_what
[8] http://cnx.org/aboutus/
[9] http://cnx.org/help/
[10] http://cnx.org/author_home

- Contact Connexions
- Report a bug
- Log out of the system

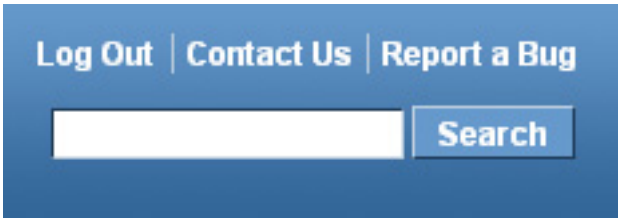
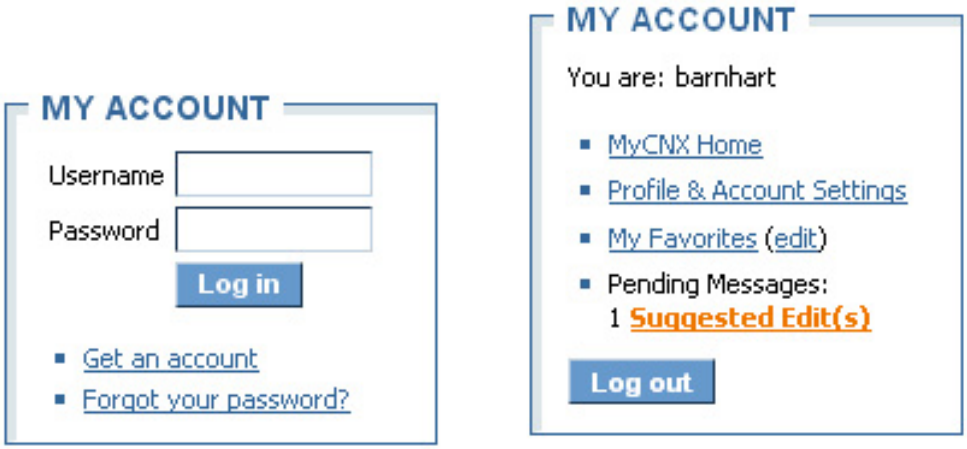


Figure 1.4: Personal toolbar after login

1.1.2.4 My Account and Login Sidebar

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



(a) (b)

Figure 1.5: My Accounts sidebar

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 reset your password (if you forgot it), and a link to the page for requesting a new Connexions account.

The image on the right is displayed if you are logged into Connexions. It features links to your **MyCNX** authoring page, 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 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.

1.1.2.6 Find Content

You can start browsing Connexions' modules and collections from the Home Page by using the Find Content portal. The Search Content box is used to search for keywords related to the content you would like to find.

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 being browsing that particular selection of Connexions content.

1.1.2.7 Create Content

It's easy to get started creating content on Connexions. The Create Content portal contains links to help you being authoring your own modules and collections.

1.1.2.8 Spotlight Sidebar

This sidebar features authors and their content, feedback from Connexions users, and events related to Connexions.

Figure 1.6: The Spotlight sidebar

1.1.2.9 News Sidebar

This sidebar contains links to news items[11] about the latest developments with Connexions. This includes updates to the site and outside publications that feature Connexions.

1.1.3 Account Requests

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

1. Click on the **Get An Account** link in the **My Account** sidebar. The Request a Connexions Account (Figure 1.7) page displays.

[11] http://cnx.org/news/

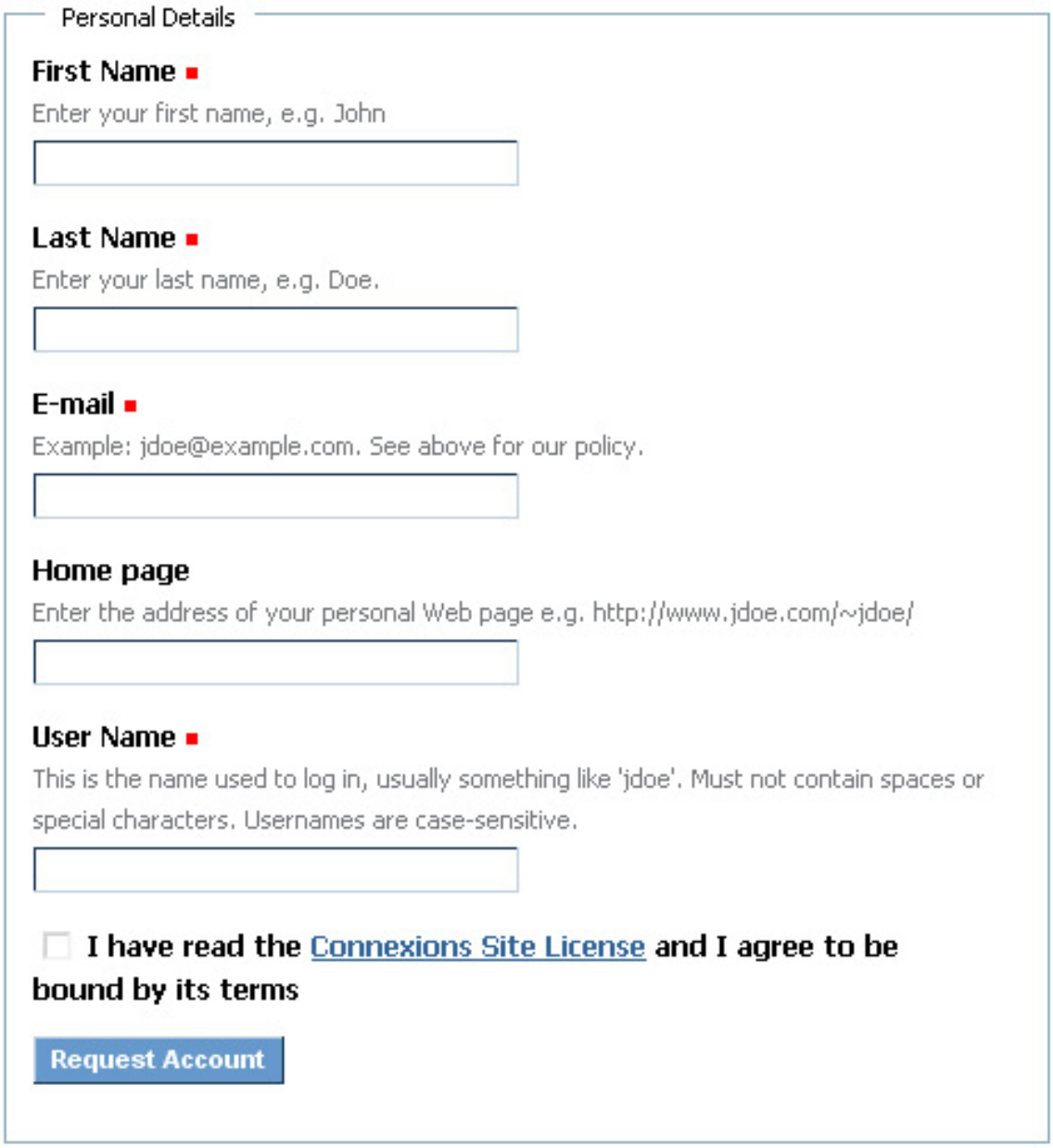


Figure 1.7: "Request a Connexions Account" page

2. 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.
3. Type in a user name.
4. 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.
5. 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.

1.1.4 Logging In and Out

To log into Connexions, enter your user name and password in the **My Account** sidebar and click `Log in`.

To log out, click the **Log out** link located above the Search Box (Figure 1.4), in the upper right side of the Connexions home page.

1.2 Reporting a Connexions Bug[12]

1.2.1 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[13] 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** link. The Bug Report Submission Form[14] 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 email address we can't get back to you with an answer.
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. A content object's ID appears in its URL or on the page behind the "More About This Content" link.
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. If possible, include step-by-step instructions for reproducing the bug.
9. Click `Submit` to enter the bug into the Connexions tracking system.

1.2.2 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[15] or to a Help[16] page, use the Back function on your browser to redisplay the error message.

[12] This content is available online at <http://cnx.org/content/m11836/1.7/>.
[13] http://cnx.org/
[14] http://cnx.org/bug_submit_form
[15] http://cnx.org/bug_submit_form
[16] http://cnx.org/help/

- 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, include the workgroup and/or module with which you were working. If you were using the Collection Composer, mention which course 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.

1.3 Viewing Connexions Content[17]

See the Introduction to Connexions (Section 1.1) module for an overview of Connexions.

1.3.1 Searching the Content Commons

You can search the **Content Commons** for a specific module or a course using the `Search` button and text box in the upper right of the Connexions web pages. Type in a name, keyword, course 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. To view a module or collection, click its title.

1.3.2 Opening a Collection

Select the "Content" tab (above the `You are here` Breadcrumb bar) to display the Content Commons (Figure 1.8) screen. This tab allows you to search for modules or collections by title, author name, or subject area. You can also display a listing of all content by subject, title, author name, keyword, popularity, and more by selecting the appropriate item under the "Browse All Content" heading on the tab.

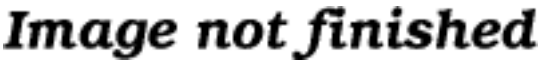


Figure 1.8: Connexions Courses

1.3.3 Navigating Within a Collection or Course

When you first open a course, the course title page displays. This page contains two panels. One panel displays the `Start Course` link and course information, such as the institution, instructor, contributing authors, and a brief description of the course. Click `Start Course` to display the entire course, beginning with the first module. The other panel is labeled "Course Contents" and it contains a table of contents for the course that lists the names of the sections and modules within that course. You can click a module name to display that module.

[17] This content is available online at <http://cnx.org/content/m11837/1.14/>.

Once you display a module from within a course, you can move to the previous module or to the next module in the course by clicking the ≪`Previous` or `Next`≫ links that appear in the upper right corner.

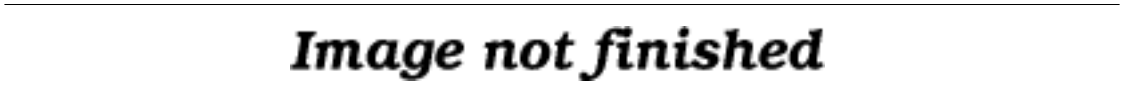


Figure 1.9: Links to the previous and next modules in a course

NOTE: The ≪`Previous` and `Next`≫ links only display when you access a module from within a course. If you access a module directly from a browser, these links to do not appear.

1.3.4 Viewing Related Material

You can view content related to the module you are displaying using the links in the "Related Material" panel that appears to the left of the module content. This panel includes:

- The names of other modules that contain content similar to the current module. Click a module name to display that module.
- Any other courses that contain the current module. Click a course name to display that course.

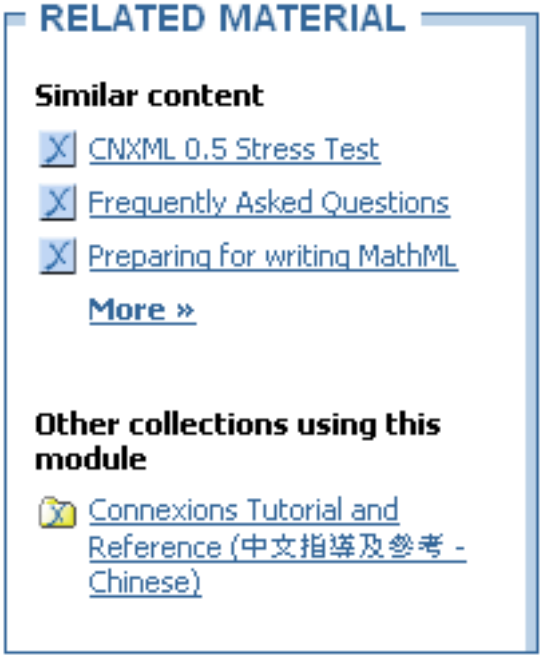


Figure 1.10: Links to related material

In the panel 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 or to a website outside of Connexions.

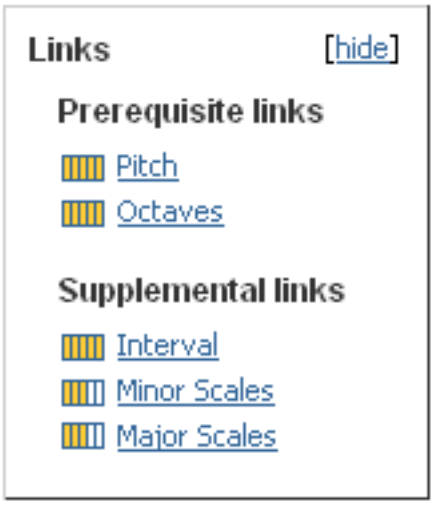


Figure 1.11: Links to pre-requisite materials, supplemental materials, and examples (not shown in this example)

1.3.5 Displaying Information about the Module

To display the **metadata** for the module, click the `Metadata` link at the end of the module. 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.

1.3.6 Viewing Module Revision History

To view the revision history of a module, click the `Version History` link at the end of the module. When a new module is published Connexions assigns a revision number to that version of the module. If if becomes necessary to update the module, the author checks it out, 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.

1.3.7 Printing a Module

To generate a PDF file of a module, which you can print, click the `Download PDF` link that appears in the upper right corner of the module in the "Content Actions" panel. Sometimes you may want or need a printed copy of the material for studying purposes. You can generate a PDF file for any module. These PDF files are formatted for printing with page numbers, headers, footers, and numbered headings. Since the printing process uses a PDF file, you must have a software package that can print PDF files installed on your PC.

Chapter 2

Working with Content

2.1 MyCNX and the Work Areas[1]

You can begin managing and creating new content from the authoring side of Connexions, known as "MyCNX." You can access it by clicking on the "MyCNX" tab located at the top of every Connexions page.

The MyCNX Home Page (Section 2.1.1: MyCNX Home Page) contains links to the Work Areas (Section 2.1.2: Connexions Work Areas): My Workspace (Section 2.1.2.2: My Workspace) and My Workgroups (Section 2.1.2.3: Workgroups). You can read more about the different parts of MyCNX below.

- MyCNX Home Page (Section 2.1.1: MyCNX Home Page)
  - Navigation sidebar (Section 2.1.1.1: Navigation Bar)
  - Create and edit content (Section 2.1.1.2: Create and Edit Content)
  - Access lenses (Section 2.1.1.3: Access lenses)
  - Guides and tutorials (Section 2.1.1.4: Guides and Tutorials)
  - Recently Viewed (Section 2.1.1.5: Recently Viewed)
  - Show/Hide Sidebars Icon (Section 2.1.1.6: Show/Hide Toolbars)
- Work Areas (Section 2.1.2: Connexions Work Areas)
  - My Workspace (Section 2.1.2.2: My Workspace)
  - Workgroups (Section 2.1.2.3: Workgroups)
  - Creating a New Workgroup (Section 2.1.2.4: Creating a New Workgroup)
  - Managing Work Area Contents (Section 2.1.3: Managing Work Area Contents)
  - Creating New Items (Section 2.1.3.1: Creating New Items)
  - Adding Existing Items from the Content Commons (Section 2.1.3.2: Adding Existing Items from the Content Commons)
  - Viewing the Work Area Contents (Section 2.1.3.3: Viewing the Work Area Contents)
  - Removing Items from a Work Area (Section 2.1.3.4: Removing Items from a Work Area)
  - Renaming Items in a Work Area (Section 2.1.3.5: Renaming Items in a Work Area)
  - Cutting or Copying Items in a Work Area (Section 2.1.3.6: Cutting or Copying Items in a Work Area)
  - Work Area Properties (Section 2.1.4: Work Area Properties)
    - Editing My Workspace Title and Description (Section 2.1.4.1: Editing My Workspace Title and Description)
    - Editing a Workgroup Title, E-mail Address, and Description (Section 2.1.4.2: Editing a Workgroup Title, E-mail Address, and Description)
  - Managing Workgroup Members (Section 2.1.5: Managing Workgroup Members)

[1] This content is available online at <http://cnx.org/content/m10885/2.22/>.

* Adding Members to a Workgroup (Section 2.1.5.1: Adding Members to a Workgroup)
* Removing Members from a Workgroup (Section 2.1.5.2: Removing Members from a Workgroup)
* E-mailing the Entire Workgroup (Section 2.1.5.3: E-mailing the Entire Workgroup)
* E-mailing an Individual Member (Section 2.1.5.4: E-mailing an Individual Member)
* Joining a Workgroup (Section 2.1.5.5: Joining a Workgroup)
* Leaving a Workgroup (Section 2.1.5.6: Leaving a Workgroup)
* Deleting a Workgroup (Section 2.1.5.7: Deleting a Workgroup)

2.1.1 MyCNX Home Page

The MyCNX Home page is a gateway to all of Connexions' authoring areas, such as your Personal Workspace (Section 2.1.2.2: My Workspace) and your Workgroups (Section 2.1.2.3: Workgroups). It is also a quick way to manage and edit your Lenses (Section 2.1.1.3: Access lenses) and your Favorites. You can access the MyCNX home page by clicking on the tab at the top of any Connexions page. You must obtain a Connexions account (Section 1.1.3: Account Requests) and log in to the site (Section 1.1.4: Logging In and Out) in order to use MyCNX.

2.1.1.1 Navigation Bar

The navigation sidebar (Figure 2.1) is located on the left side of all MyCNX authoring pages. You can always use the link at the top of the sidebar to return to the MyCNX Home page. The "By Type" section allow you to find all of the Modules, Collections, and Lenses you've created by clicking on the respective link. The "By Location" section displays your Personal Workspace and all of the workgroups you belong to; this section is collapsed on all pages except for the MyCNX Home page. You can collapse and expand the "By Location" section by clicking the arrow located next to the heading.

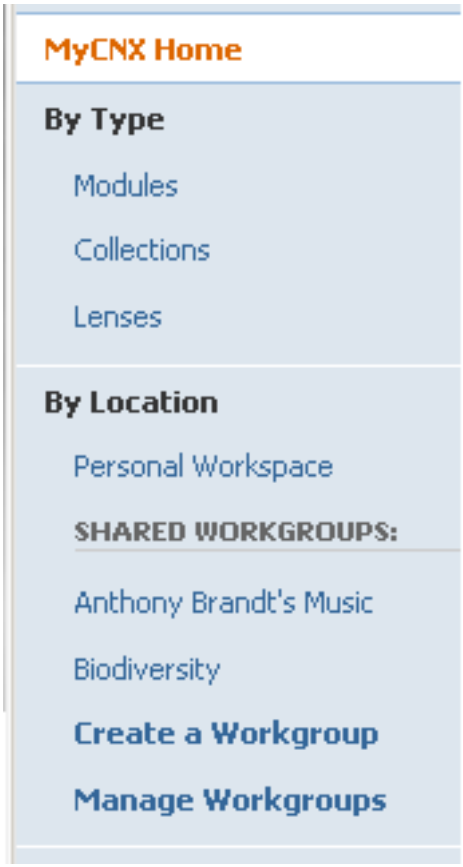


Figure 2.1: "Workspace" sidebar.

2.1.1.2 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 to bypass the work areas and immediately begin creating a new module (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.

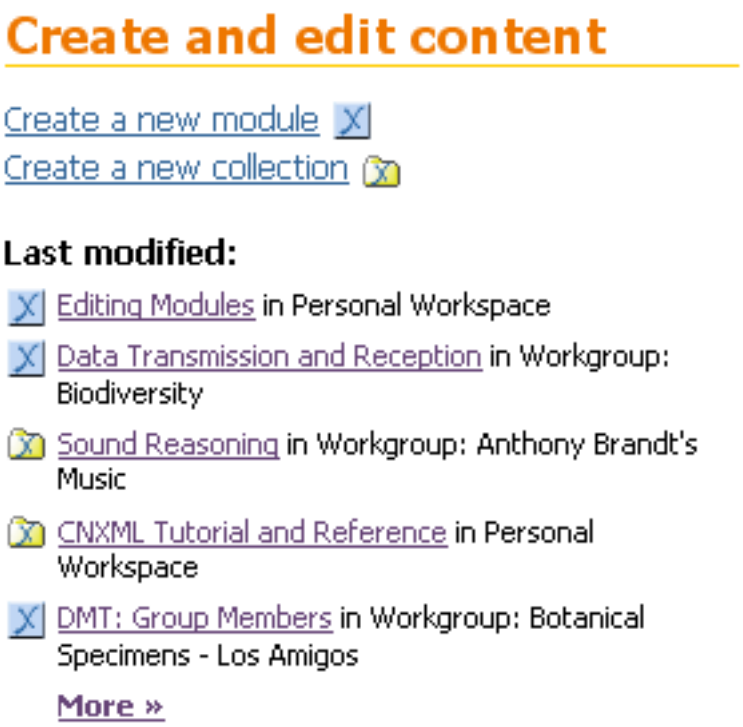


Figure 2.2: Create and Edit Content

2.1.1.3 Access lenses

To start creating and editing lenses[2] from the MyCNX home page you can follow the links under the "Access lenses" section. This area provides users with a list of their lenses (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.

[2] http://cnx.org/help/lens_what

Figure 2.3: Access lenses

2.1.1.4 Guides and Tutorials

The MyCNX Home page provides several links to tutorials that can help you begin authoring content in Connexions.

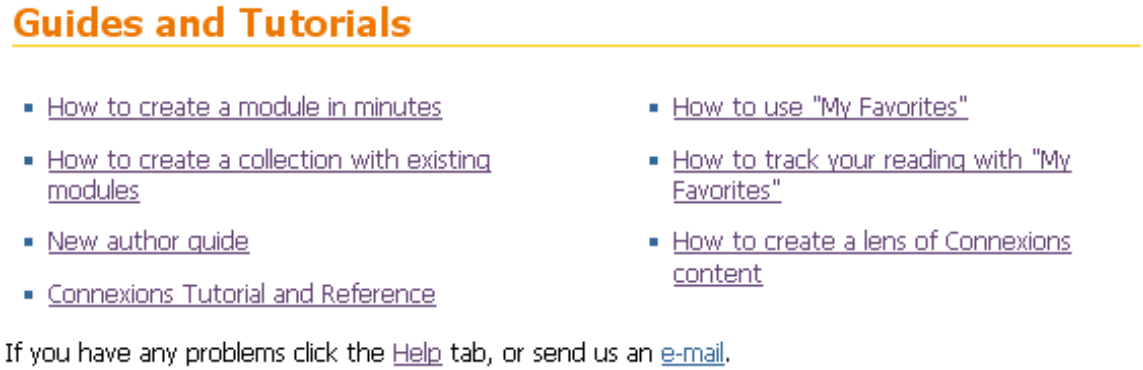


Figure 2.4: Links to guides and tutorials

2.1.1.5 Recently Viewed

Finally, the "Recently Viewed" portal located on the MyCNX Home page lists the collections, modules, and lenses that you've most recently visited.

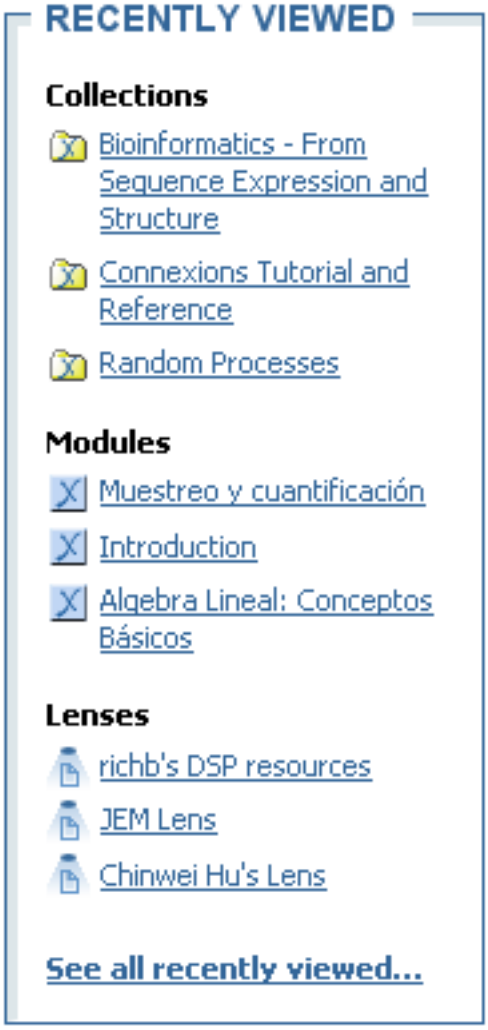


Figure 2.5: Links to recently viewed modules

2.1.1.6 Show/Hide Toolbars

Each page in the authoring area, including the MyCNX Home Page (Section 2.1.1: MyCNX Home Page) and module editing areas, features a useful "Show/Hide Sidebars" icon. 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.

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

Using the Show/Hide Sidebars icon

(a)

(b)

Figure 2.6: 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.

2.1.2 Connexions Work Areas

Connexions provides two types of areas in which to work on content: "My Workspace" and "Workgroups". You can perform the same content creating, editing, and managing tasks 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. Every person with a Connexions account has their own "My Workspace". It was created automatically when their Connexions account was set up. "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, having a different one for each project. Both of these types of work areas are described in this module.

2.1.2.1 Viewing Work Areas

Once you log into Connexions, the "Log in" sidebar in the lower left of the Connexions home page[3] is replaced by the "Workspaces" sidebar. This sidebar remains on the home page and the other Connexions web pages as long as you are logged in, and allows you to switch among your work areas easily and quickly.

[3] http://cnx.org/

2.1.2.2 My Workspace

"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 2.1.2.3: Workgroups) where a group of collaborators can make comments or edit the material.

You can display your personal work area from a Connexions web page by clicking on the **My Workspace** link in the "Workspaces" sidebar or by clicking on the **My Workspace** link in the Personal toolbar.

2.1.2.3 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 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.

Each workgroup can have modules or files that are associated with it. The members of the workgroup can add existing modules from the Content Commons to the workgroup. Once a module is created or checked out, any member of the workgroup who has editing permission on that module may modify it. To have editing permission, a person must have the maintainer role for that module. For more on this see Editing Module Roles (Section 2.3.7: Editing Module Roles).

The names of the workgroups of which you are a member are listed in the "Workspaces" sidebar. You can display the contents of a workgroup by clicking on the name of the workgroup.

2.1.2.4 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 2.1.3: Managing Work Area Contents) for a description of how to add content to the new workgroup. See Managing Workgroup Members (Section 2.1.5: Managing Workgroup Members) for a description of how to add members to the new workgroup.

2.1.3 Managing Work Area Contents

To display "My Workspace", click on the **My Workspace** link in the "Workspaces" sidebar. To display a workgroup, click on its name in the "Workspaces" sidebar. If this is the first time you have displayed "My Workspace" or if you have just created the workgroup, there will be no content in the work area.

The "My Workspace" display has a "Contents" and a "Properties" tab. The workgroup display has "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 shows the "Contents" tab for a "My Workspace" that contains a collection and three modules.

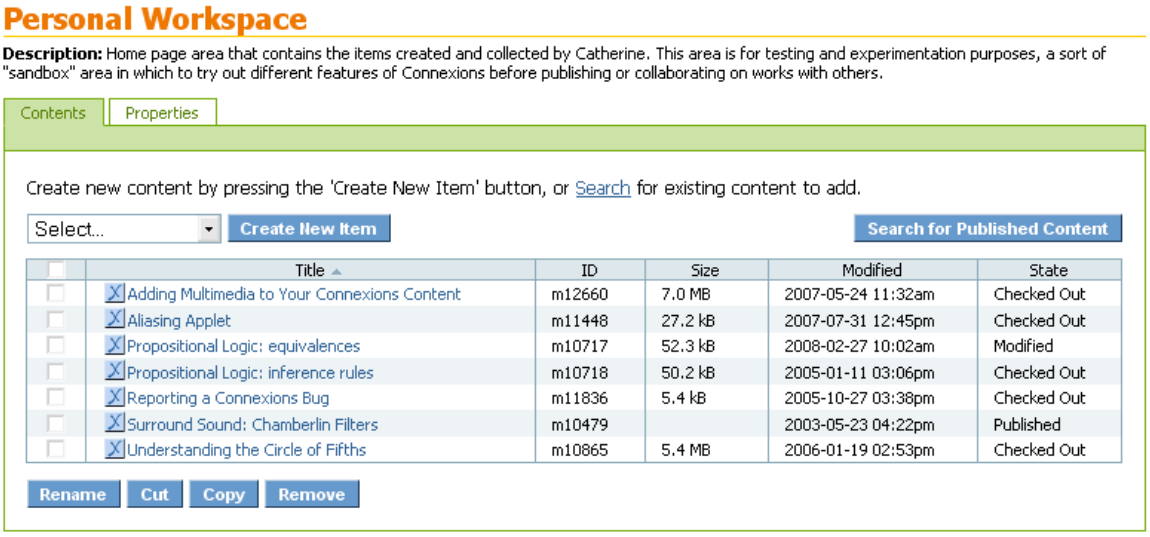


Figure 2.7: "Contents" tab for "My Workspace".

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

- Creating New Items (Section 2.1.3.1: Creating New Items)
- Adding Existing Items from the Content Commons (Section 2.1.3.2: Adding Existing Items from the Content Commons)
- Viewing the Work Area Contents (Section 2.1.3.3: Viewing the Work Area Contents)
- Removing Items from a Work Area (Section 2.1.3.4: Removing Items from a Work Area)
- Renaming Items in a Work Area (Section 2.1.3.5: Renaming Items in a Work Area)
- Cutting or Copying Items in a Work Area (Section 2.1.3.6: Cutting or Copying Items in a Work Area)

2.1.3.1 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.
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 2.2) 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 2.3) 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 "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.

2.1.3.2 Adding Existing Items from the Content Commons

You can add modules or collections that already exist in the Content Commons to your work area. In this way you can re-use information that appears elsewhere in Connexions. To add existing items, use the following steps:

1. Click `Search for Published Content`. The "Search the Connexions repository" screen displays.
2. Enter an object ID, title, keyword, author name, or text string in the text box next to the `Search` button.
3. If desired, deselect the "modules" or "collections" checkboxes and click `Search`. A list of the modules and collections that match your search criteria displays on the screen.
4. Select the checkboxes next to the items you want to add to your work area, or use the checkbox in the title bar to select all the results.
5. Click the down arrow in the "Add selected content to" text box. A drop-down list of all the work areas to which you belong displays. Select a work area from the list.
6. Click `Add`. The selected items are added to the work area you selected and the "Contents" tab for that work area displays.

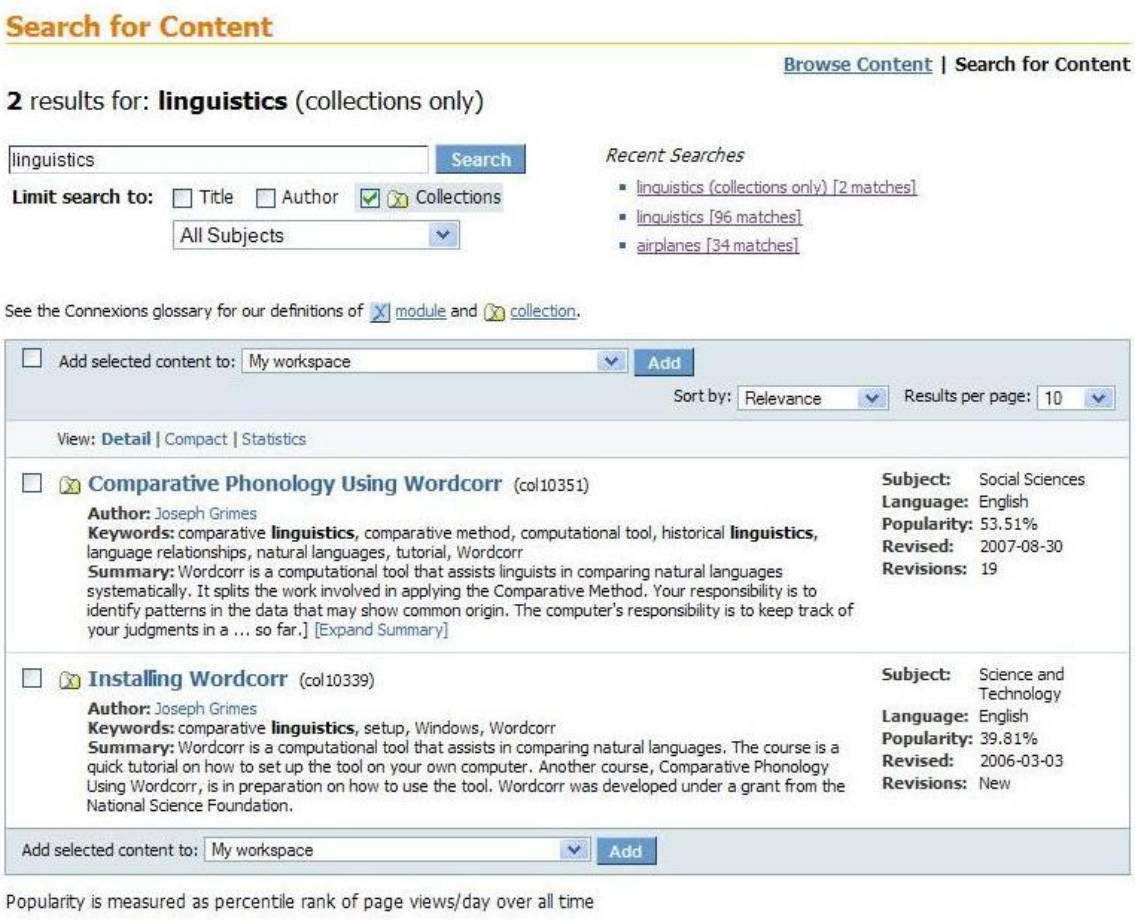


Figure 2.8: Results of a search for existing content.

2.1.3.3 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 Commons and is now accessible by the public.
- Checked Out - The item was copied from the Content Commons 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 Commons.
- 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".

2.1.3.4 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.

2.1.3.5 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.

2.1.3.6 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.

2.1.4 Work Area Properties

You can edit certain properties of a work area, such as the name of a workgroup or the description of your personal work area, "My Workspace". These changes are accessed on the work area's "Properties" tab.

2.1.4.1 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`.

2.1.4.2 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 "Workspaces" sidebar to display the workgroup screen.
- Click on the "Properties" tab. The "Properties of" 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`.

2.1.5 Managing Workgroup Members

Once you have created a workgroup or you have become a member of an existing workgroup, you can add other members to the workgroup. These members can work together on the content in the workgroup.

NOTE: Only a person who has a Connexions account can be a member of a workgroup.

The names and e-mail addresses of the members of the workgroup are listed on the "Members" tab. 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.

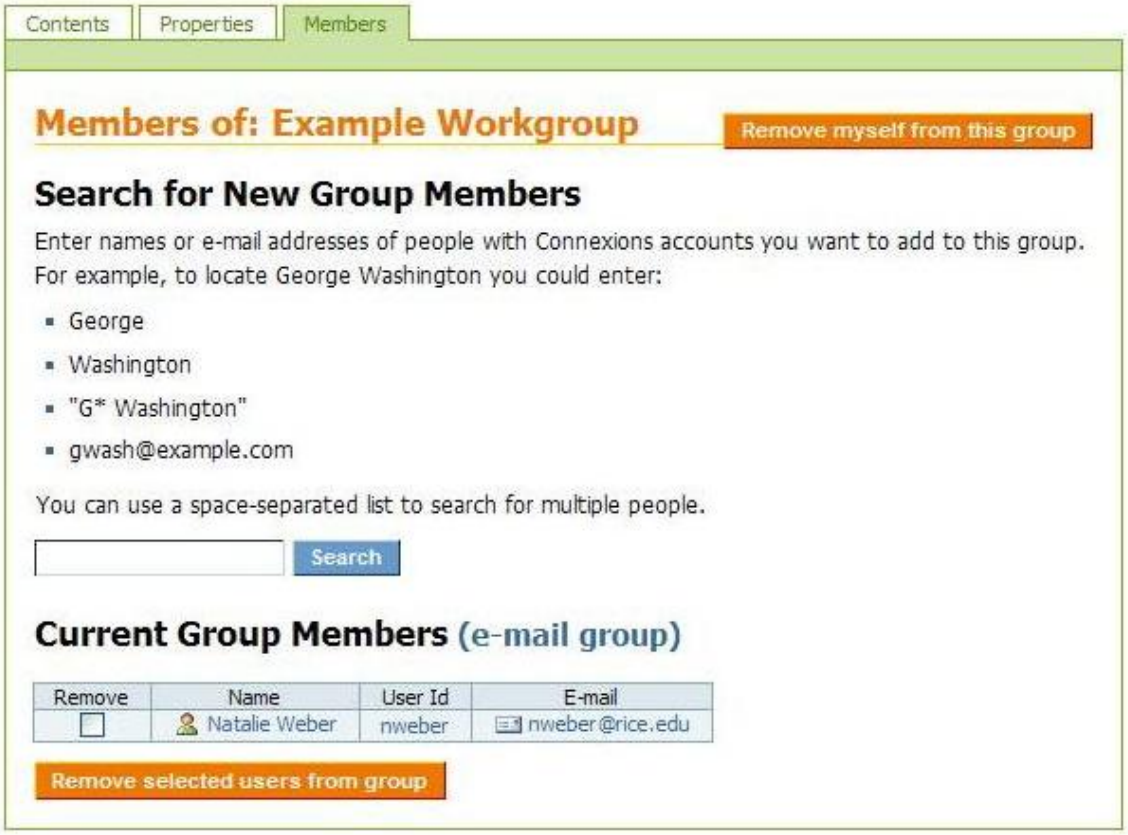


Figure 2.9: Workgroup "Members" tab.

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

- Adding Members to a Workgroup (Section 2.1.5.1: Adding Members to a Workgroup)
- Removing Members from a Workgroup (Section 2.1.5.2: Removing Members from a Workgroup)
- E-mailing the Entire Workgroup (Section 2.1.5.3: E-mailing the Entire Workgroup)
- E-mailing an Individual Member (Section 2.1.5.4: E-mailing an Individual Member)
- Leaving a Workgroup (Section 2.1.5.6: Leaving a Workgroup)
- Deleting a Workgroup (Section 2.1.5.7: Deleting a Workgroup)

Joining a workgroup requires an invitation. See Joining a Workgroup (Section 2.1.5.5: Joining a Workgroup) for more information.

2.1.5.1 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.

2.1.5.2 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.

2.1.5.3 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.

2.1.5.4 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.

2.1.5.5 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.

2.1.5.6 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.

2.1.5.7 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. Display the "Members" tab for the workgroup you wish to delete.
2. Select the checkbox in the "Remove" column next every members' name.
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 "Workspaces" sidebar of all members of the workgroup.

2.2 How to Use the Collection Composer[4]

2.2.1 The Collection Composer

The Connexions 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 Commons can be used to create a collection. You can perform the following tasks with the Collection Composer:

- Create a New Collection (Section 2.2.2: Create a New Collection)
- Enter Identifying Metadata for a New Collection (Section 2.2.3: Enter Identifying Metadata for a New Collection)
- Add a New Section (Section 2.2.4: Add a New Section)
- Add a Subsection (Section 2.2.5: Add a Subsection)
- Add a Module (Section 2.2.6: Add a Module)
- Provide an Alternate Module Title (Section 2.2.7: Provide an Alternate Module Title)
- Select the Module Version (Section 2.2.8: Select the Module Version)
- Edit or Add Links (Section 2.2.9: Edit or Add Links)
- Rearrange Contents (Section 2.2.10: Rearrange Contents)
- Remove Collection Contents (Section 2.2.11: Remove Collection Contents)
- Assign Roles to Collaborating Authors (Section 2.2.12: Assign Roles to Collaborating Authors)
- Set Notation Parameters (Section 2.2.13: Set Notation Parameters)
- Preview the Collection (Section 2.2.14: Preview the Collection)
- Publish the Collection
- Edit an Existing Collection (Section 2.2.16: Edit an Existing Collection)

2.2.2 Create a New Collection

Creating a new collection is similar to creating a new module (Section 2.1.3.1: Creating New Items). You access the Collection Composer from your workspace or workgroup[5] .

1. Select Collection from the drop-down list next to the `Create New Item` button and click `Create New Item`. The "New Content: Licensing" screen displays.
2. Select the check box at the bottom of the screen to signify that you agree to place the collection under the Creative Commons Attribution license[6] .
3. Click `Next`. The "Metadata" tab displays with the message that a collection has been created. The "Collection Status" sidebar appears on the right side of the screen and lists the current state of the collection and the last action taken, which was creation of the collection along with the name of the person who created the collection and when they created it.

[4] This content is available online at <http://cnx.org/content/m10288/2.24/>.
[5] http://cnx.rice.edu/help/reference/workgroups
[6] http://creativecommons.org/licenses/by/1.0/

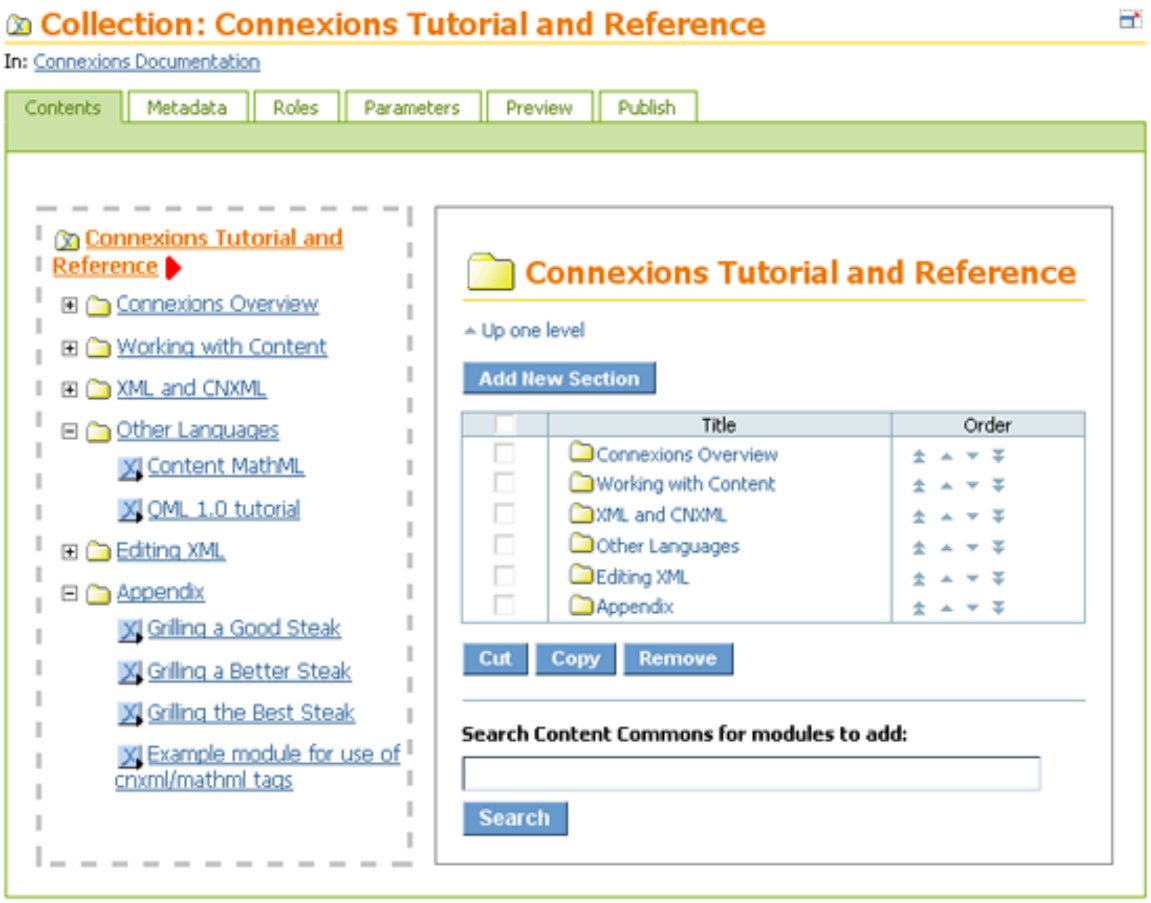


Figure 2.10: Collection Composer editing screen

2.2.3 Enter Identifying Metadata for a New Collection

1. Select the "Metadata" tab in the Collection Composer.
2. Enter the appropriate information in the metadata fields, including:
   - Title (required) - An applicable title for the collection
   - Language (required) - The primary language used in the collection
   - Collection Subtype - A subtype that reflects the intended use of the collection, such as Course or Report
   - Subject(s) - The subject categories that apply to the collection
   - Keywords - Terms that describe the most important topics in the collection
   - Summary - A brief description of the collection. You may markup the summary section with inline CNXML tags.
   - There are additional metadata fields for collections that will be used as courses at schools or universities. These fields are:
     - External Web Page

- Code or Number
- Institution
- Instructor

If you do not make an entry in a required field, you will be prompted for an entry when you click `Save` to record the metadata.

3. Click `Save`. The metadata entries are saved and the "Contents" tab displays with the title of the course. This tab is where you do the actual assembly of the course. It has two areas. The left side of the tab has a course outline display. The right side of the tab is where you add, arrange, or remove content objects in the course.

2.2.4 Add a New Section

A section is a grouping of related modules within a collection. Each section has a heading that identifies its contents to the viewers. You may place sections within sections, creating subsections.

1. Select the "Contents" tab in the Collection Composer.
2. Click on the title of the collection in the collection outline, if it is not already highlighted. It is the highest-level object in the collection outline.
3. Display the "Contents" tab, if it is not already displaying.
4. Click `Add New Section` in the "Contents" tab. A new section is added to the collection outline as the last object and the "Edit" tab displays with a blank field for the name of the new section.
5. Enter the name of the new section in the "Title" field.
6. Click `Save`.

The title you entered appears as the new section in the collection outline. This section is highlighted, enabling you to add content to it in the form of a subsection or a module.

2.2.5 Add a Subsection

A subsection is a grouping of related modules within a section. Each subsection has a heading that identifies its contents to the viewers.

1. Select the "Contents" tab in the Collection Composer.
2. Click on the section title in the collection outline, if it is not already highlighted.
3. Display the "Contents" tab, if it is not already displaying.
4. Click `Add New Section` in the "Contents" tab. A new subsection is added to the collection outline under the section that was highlighted. The "Edit" tab displays with a blank "Title" field for the name of the subsection.
5. Enter the name of the subsection in the "Title" field.
6. Click `Save`.

The title you entered appears as the new subsection in the collection outline. This subsection is highlighted, enabling you to add content to it.

2.2.6 Add a Module

You can add a published module to the collection from the Content Commons or from the collection's own workspace or workgroup. When adding a module, Connexions does not copy the module into your collection. **The module remains in the Content Commons and Connexions inserts a reference or a pointer to the module in your collection.** Any new or existing modules can be added to a collection with the following considerations:

- A module cannot be inserted into a collection more than once.
- A module must be published in Connexions before you can insert it into a collection.
- Searches in the Content Commons for modules use titles, names, authors, keywords, or text strings.

If no matches are found for your search string, Connexions displays some suggestions about search entries.

2.2.6.1 Add a Module from the Content Commons

To search the Content Commons for a module and add it as content in your collection, use the following steps:

1. Select the "Contents" tab in the Collection Composer, if it is not already displaying.
2. Click on the item title in the collection outline where you want to insert the module, if it is not already highlighted. This item could be the collection title, a section title, or a subsection title.
3. Enter a word or text string related to the module in the search text box.
4. Deselect the checkbox for "Collections". Leave the checkbox for "Modules" selected.
5. Click `Search`. A list of all the modules that match the search input displays. If more matches occurred than will fit on one page, then additional pages of matches are listed.
6. Select the checkbox for any module or modules you want to add as content.
7. Click `Add selected content`.

The titles of the selected modules appear in the collection outline under the section you selected.

2.2.6.2 Add a Module from Your Workspace or Workgroup

To add a module from your workspace or workgroup as content in your collection, use the following steps:

1. Display the "Contents" tab for your workspace or workgroup that contains the collection and the module.
2. Select the checkbox for the desired module. You can add only modules that are in the "published" state.
3. Click `Copy`. A message displays that the item was copied.
4. Click on the collection title to access the Collection Composer. The `Paste` button now appears on the "Content" tab along with the `Cut`, `Copy`, and `Remove` buttons.
5. Click on the section title where you want to insert the module.
6. Click `Paste` on the "Content" tab for the collection.

The title of the module appears in the collection outline under the section you selected.

> NOTE: If you try to add a module that is not published, an error message displays. To verify that the module you want to add is published, look in the "State" column in the "Contents" tab of your work area.

2.2.7 Provide an Alternate Module Title

Modules that are published in the Content Commons display the titles given to them by their authors. If these titles are not appropriate for the module when it appears in your collection, you can provide an alternate title for the module. The module displays your alternate title when it is accessed from within your collection. The module displays its original title whenever it is accessed from outside of your collection.

1. Click the module title in the collection outline. The module level "Edit" tab appears on the right side of the screen.
2. Type the new title for the module in the "Alternate Title" field.
3. Click `Save`.

The module title in the collection outline changes to the alternate title.

2.2.8 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 select which version of each module is used in your collection.

1. Click the module title in the collection outline. The module level "Edit" tab appears on the right side of the screen.
2. Select the `Version` radio button for the version of the module you want to use in the collection. The default is "Always use newest version".
3. Click `Save`.

2.2.9 Edit or Add Links

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 at the collection level or at the individual module level. Any changes you make to the links in an existing module only display with the instance of the module in your collection.

2.2.9.1 Edit Existing Links

1. Click the module title in the collection outline. The module "Edit" tab appears on the right side of the screen.
2. Click on the "Links" tab.
3. Make the desired changes in the "Name", "Link Type", and "Strength" fields in the "Edit Links" panel of the tab.
4. Select the checkbox in the "Delete" column for any link you do not want to appear in the module.
5. Click `Update Links`.

The changes are made to the original author's list of links and will appear in the "Links" sidebar when the module is viewed.

2.2.9.2 Add Links to a Module

1. Click the module title in the collection outline. The module "Edit" tab appears on the right side of the screen.
2. Click on the "Links" tab.
3. To enter a new link, type the link name in the "Enter a Name" field in the "Add a New Link" panel of the tab.
4. Select the category of link from the drop-down list next to the "Link Type" field.
5. Select the importance of the link from the drop-down list next to the "Link Strength" field.
6. Enter the target for the link in the "Module ID" or "Link URL" field.
7. Click `Add`.

The link will appear in the "Links" sidebar when the module is viewed, along with the original author's links, if any.

2.2.10 Rearrange Contents

The Collection Composer has two methods of rearranging the content items within a collection. The first method uses up and down arrows to move items around within a collection or within a section. The second method uses the `Cut` and `Paste` buttons to move items from collection to section or from section to section.

2.2.10.1 Rearrange with the Up and Down Arrows

To move items around within a collection or within a section with the up and down arrows, use the following steps:

1. Select the "Contents" tab in the Collection Composer.
2. Click on the collection title to rearrange the content items within the collection, or click on a section title to rearrange the content items within that section. A listing of the contents of the collection or selected section appears on the right side of the screen.
3. Click the up or down arrows in the "Order" column to relocate a content item to a different location within the collection or section. The outer two arrows move the content item to the first or last position. The inner two arrows move the content item up or down to the next position.

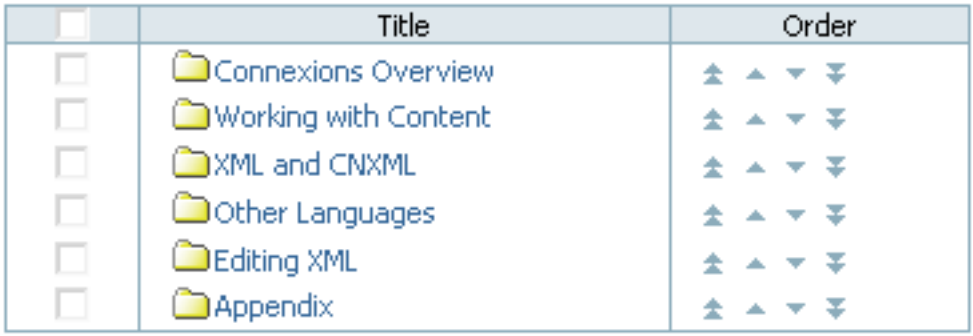


Figure 2.11: "Title" and "Order" columns

2.2.10.2 Rearrange with the Cut and Paste Buttons

To move items from collection to section or from section to section with the `Cut` and `Paste` buttons, use the following steps:

1. Select the "Contents" tab in the Collection Composer.
2. Click on the collection title in the collection outline to cut and paste content items within the collection, or click on a section title in the collection outline to cut and paste content items within that section. A listing of the contents of the collection or selected section appears on the right side of the screen.
3. Select the checkbox to the left of the content item or items you want to move.
4. Click `Cut`.
5. Click on the title of the section in the collection outline for the section in which you want to paste the content item.
6. Click `Paste`. The content item or items appear in the collection outline under the selected section title and disappear from under the originating collection or section title.

2.2.11 Remove Collection Contents

To remove a content item from a collection or from a section within a collection, use the following steps:

1. Select the "Contents" tab in the Collection Composer.

2. Click on the collection title to remove a content item from within the collection, or click on a section title to remove a content item from within that section. A listing of the contents of that section appears on the right side of the screen.
3. Select the checkbox for the object to be removed.
4. Click `Remove`. The item is removed from the collection or section.

2.2.12 Assign Roles to Collaborating Authors

As creator of a collection, by default you hold all three roles[7] : author, maintainer, and copyright holder.

- **Authors** receive credit for writing the material. The author is the person who is responsible for the creation of the work, but may or may not be the person who actually enters the work into the Connexions. Any module or collection work 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 collection. 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.

You can grant one or more roles to other Connexions account holders who you want to collaborate with you on the collection. The "Roles" tab has a search feature for locating these people. The current collaborators on the collection are listed by role under the heading "Current Roles" on the "Roles" (Figure 2.12) tab. You can change the roles assigned to these people by selecting or deselecting the check boxes in the "Authors", "Maintainers", or "Copyright Holders" columns under the heading "Change Roles". The following paragraphs describe how to grant or update roles.

[7] http://cnx.rice.edu/help/reference/roles/document_view

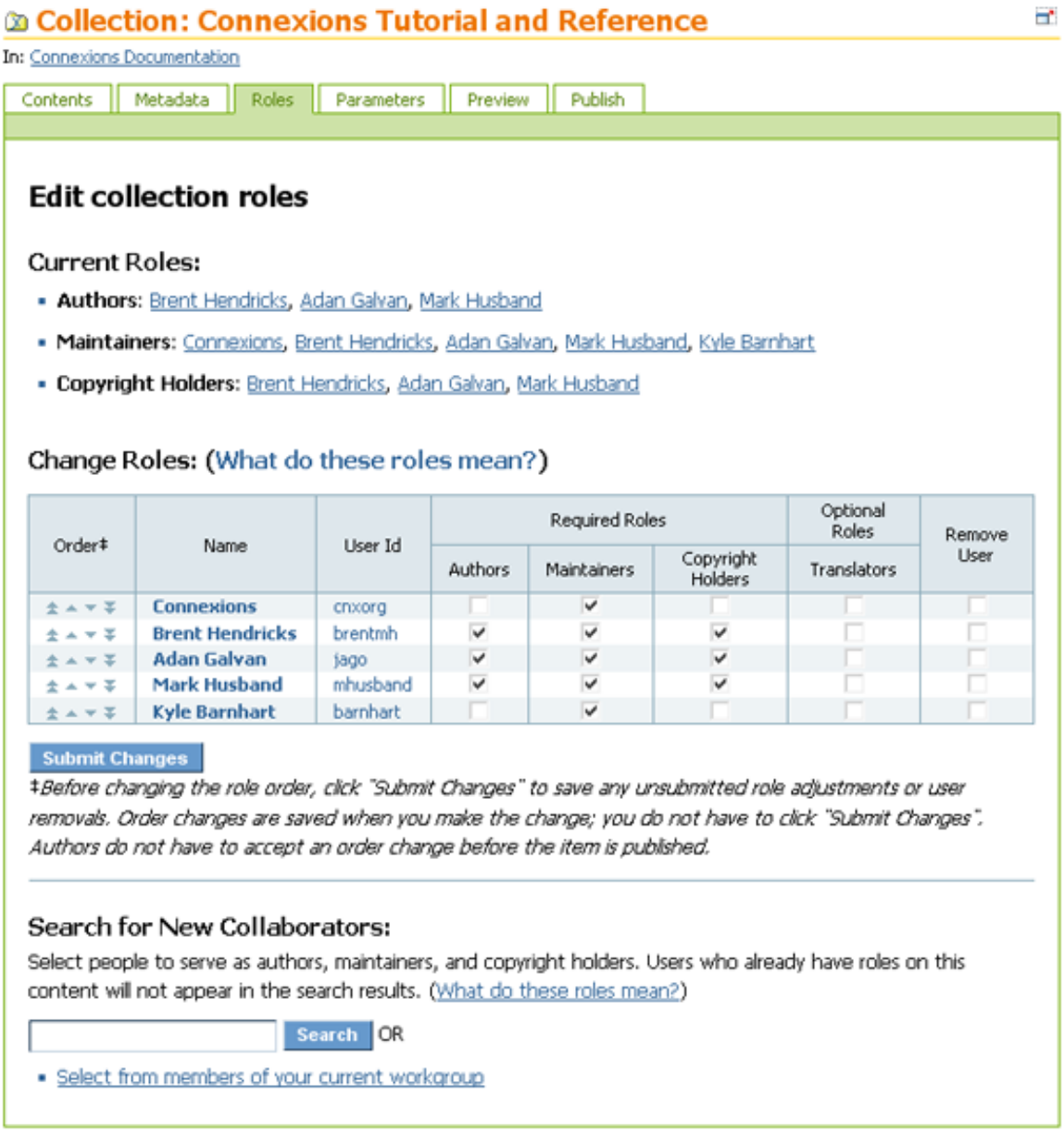


Figure 2.12: Example "Roles" tab.

When you grant or update roles for a collaborator in a collection in a workgroup, these changes do not take affect until that person accepts the role change and you publish the collection. The prospective collaborator must accept the role because Connexions provides its account holders with the ability to accept or reject a role in another person's collection. If there are any pending roles (roles not yet accepted or rejected) for a collection, Connexions will not let you publish that collection. Once the collaborator accepts the role or updated role, you must publish the collection for Connexions to recognize the collaborator. Connexions

looks to the information in a published collection to see who is an author, maintainer, or copyright holder for the collection.

> NOTE: All pending role additions or changes must be accepted or rejected by the prospective authors, maintainers, and copyright holders before you can publish a module or collection in Connexions. And you must publish a module or collection with the new roles before Connexions will recognize the new roles.

2.2.12.1 Granting Roles to Someone Outside of Your Workgroup

To grant one or more roles[8] in a collection to a Connexions account holder outside of the current workgroup, use the following steps:

1. Click the "Roles" tab to display it.
2. Type the first or last name of the person you want to grant a role to in the text box next to `Search`.
3. Click `Search`. The Connexions accounts are searched and any Connexions accounts that match the search string are displayed.
4. 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.
5. Look at the table under the "Change Roles" heading and verify that the check boxes in the "Author", "Maintainer", or "Copyright Holder" 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.
6. Click `Submit Changes`. The next time the new collaborator logs into Connexions, he or she will be notified of a pending role request. Connexions will not apply the roles to the new collaborator until he or she accepts the role request.

2.2.12.2 Granting Roles to a Member of Your Workgroup

To grant one or more roles[9] in a collection to a member of your current workgroup, use the following steps:

1. Click the "Roles" tab to display it.
2. 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. Locate the name of the person or persons in the table that you want to grant roles to and click the check boxes in the Authors, Maintainers, and Copyright Holders columns to select the roles for that person.
4. Click `Add People`. The name of the newly added 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.

A notification of a pending role request 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.

2.2.12.3 Changing the Order of Appearance for Author Names

To change the order in which the authors' names will appear on the collection, use the following steps:

[8] http://cnx.rice.edu/help/reference/roles/document_view
[9] http://cnx.rice.edu/help/reference/roles/document_view

1. Click the "Roles" tab to display it.
2. Verify that all users who should have an author role on the collection are listed in the Change Roles table. Select or deselect the check boxes in the Adjust Roles columns to make any necessary additions or changes to the collection roles.
3. Click `Submit Changes` to save the role changes. The next step will erase any role changes that you have not saved with the `Submit Changes` button.
4. Click the up or down arrows in the "Order" column to move an author up or down in the order of appearance in the module. 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.

To have an author's name appear first in the 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.

2.2.12.4 Changing Role Assignments

To change the roles[10] in a collection that were previously granted to a user, 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 change roles.
3. Click on a check box that contains a check mark to remove the check mark and to remove that role from the person. Click on an empty check box to make a check mark appear and to grant that role to the person.
4. Click `Submit Changes`. The "Change Roles" table is updated to show the role changes and to note the role changes that are pending acceptance.

A notification of a pending role change request 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.

2.2.12.5 Removing a Collaborator

To remove a collaborator from a 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 no longer appearing on the list of users with roles.

A notification of a pending role request 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.

2.2.13 Set Notation 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. To set these parameters for your collection, use the following steps:

1. Click the "Parameters" tab in the Collection Composer.
2. Select the radio buttons for the desired notations. The defaults appear in the first column.
3. Click `Update Properties`.

Connexions applies the specified notations to the parameter displays of all the modules in the collection.

[10] http://cnx.rice.edu/help/reference/roles/document_view

2.2.14 Preview the Collection

Click the "Preview" tab in the Collection Composer to display the collection on-line, just as a student would view the finished collection. This preview allows you to verify that the collection displays as you intended.

2.2.15 Publish the Collection

A published collection is available to anyone on the Internet who wants to view it. Once the collection is complete, you will publish it in Connexions.

1. Verify that the metadata entries, roles, and on-line version of the collection are as you want them.
2. Click the "Publish" tab in the Collection Composer.
3. Type a brief explanation of this revision of the collection in the "Description of Changes" field. This text displays on the revision history page for the collection.
4. Click `Publish`.

The collection is published in Connexions. You may check out the collection at any time and revise it with the Collection Composer.

> NOTE: Before Connexions will publish a collection, the collection must have a person named to the Author, Maintainer, and Copyright Holder roles. In addition, the collection can have no pending role requests. All role requests must be either accepted or rejected.

2.2.16 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 a maintainer role. To use the Collection Composer on an existing collection, begin in your workspace or workgroup and use the following steps:

1. Click `Search for Published Content`. The "Search the Connexions repository" screen displays.
2. Enter a word or text string related to the collection in the search text box.
3. Deselect the checkbox for "Modules". Leave the checkbox for "Collections" selected.
4. Click `Search`. A list of the objects that match the search string displays.
5. Select the checkbox for the desired collection
6. Select the workspace or workgroup where the collection will be placed from the "Add selected content to" drop-down list
7. Click `Add`. The collection is added to the specified workspace or workgroup.
8. Open the collection in the Collection Composer and edit as described under the Create a New Collection (Section 2.2.2: Create a New Collection) heading earlier in this section.

2.3 Editing Modules[11]

2.3.1 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

[11] This content is available online at <http://cnx.org/content/m10887/2.28/>.

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.

An example of the contents of a new module are shown in Figure 2.13, which displays the "Manage Files" screen located under the "Files" tab.

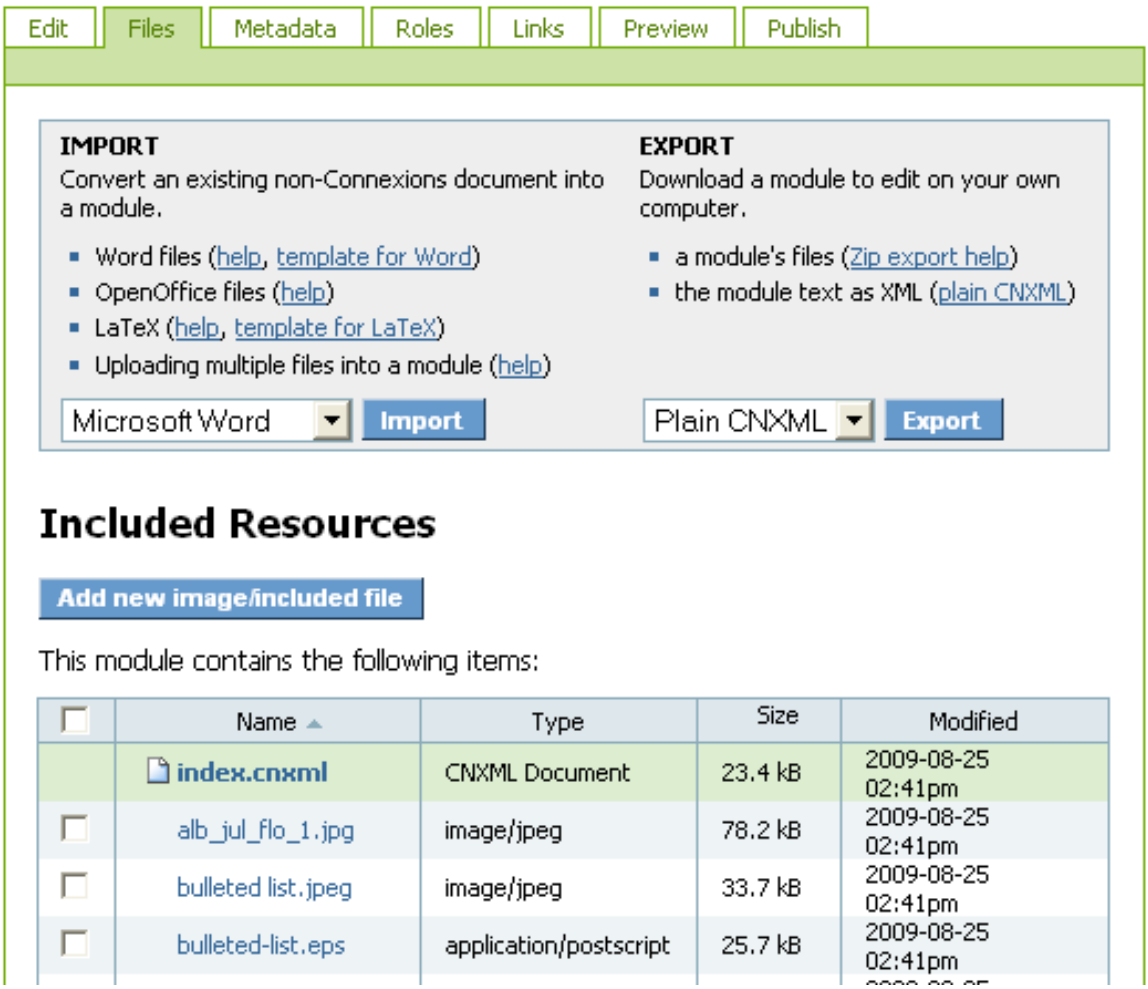


Figure 2.13: "Files" tab of a module.

2.3.2 The Module Status Sidebar

The "Module Status" (Figure 2.14) sidebar contains important status information about the module and several links to useful tasks for the module editor. This sidebar appears to the right of the module contents and contains the following items:

- State (p. 23) - the current status of the module
- Last action - The last action performed on the module, who performed it, and when

- Actions - This item displays links to the Connexions pages for publishing (Section 2.3.12: Publishing the Module) modules to the Content Commons and discarding (Section 2.3.11: Discarding Your Changes) the current operations on the module. When editing an existing module, this item also displays links for suggesting edits (Section 2.6) to original authors and creating a derived copy (Section 2.5) of an existing module
- View - This item displays links to preview the on-line (Section 2.3.10.1: Previewing Modules On-Line) version of the module, the print (Section 2.3.10.2: Previewing a Print Version of the Module) version, and the CNXML source (Section 2.3.10.4: Previewing the code source). When editing an existing module, this item also displays a summary of the differences (Section 2.3.10.3: Viewing Change Summary) between the version of the module in your work area and the last published version

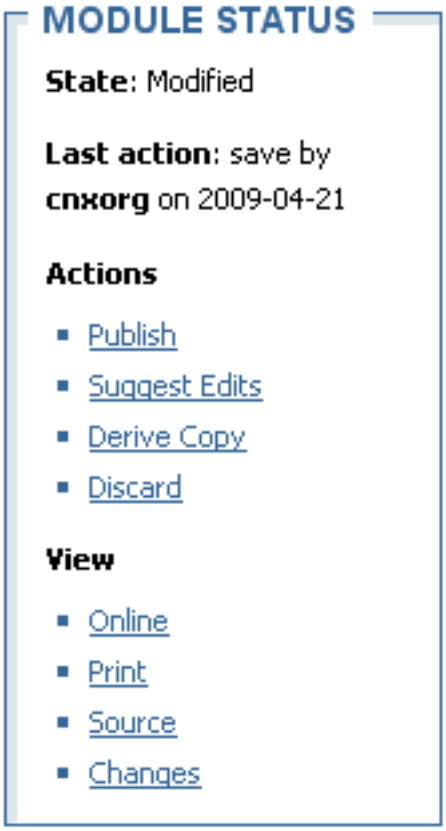


Figure 2.14: The "Module Status" sidebar.

2.3.3 Editing Modules

Once you have created (Section 2.1.3.1: Creating New Items) a new module in or added (Section 2.1.3.2: Adding Existing Items from the Content Commons) an existing module to your work area, 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 2.3.4: Editing the CNXML File) file.
2. Add (Section 2.3.5: Adding Associated Files to a Module) any other files that you want to associate with the module.
3. Edit the metadata (Section 2.3.6: Editing Metadata) to enter the appropriate search keywords and the module title.
4. Edit the module roles (Section 2.3.7: Editing Module Roles).
5. Add links (Section 2.3.9: Adding Featured Links to a Module) to the module.
6. Preview (Section 2.3.10: Previewing the Module) the module in both print and on-line versions.
7. Publish (Section 2.3.12: Publishing the Module) the module.

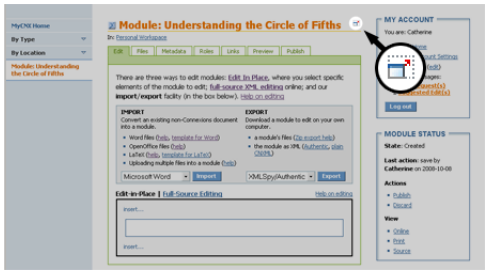
Each of these operations of module editing will be discussed in the following sections.

2.3.3.1 Show/Hide Toolbars

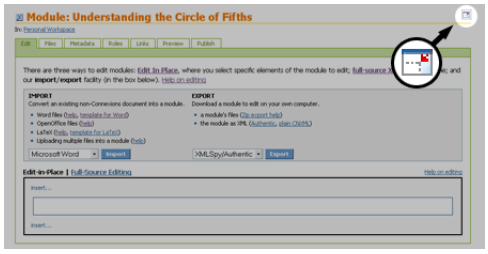
Each page in the authoring area, including the MyCNX Home Page (Section 2.1.1: MyCNX Home Page) and module editing areas, features a useful "Show/Hide Sidebars" icon. When you click on the icon, the right and left authoring sidebars are hidden and the content pane is expanded. This is especially useful when editing modules using Edit-In-Place.

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

Using the Show/Hide Toolbars icon



(a)



(b)

Figure 2.15: Demonstration of the Show/Hide Toolbars 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.

2.3.4 Editing the CNXML File

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 2.3.4.1: Import Content into the "index.cnxml" File)
- Export the index.cnxml file to an external XML editor (Section 2.3.4.2: Export the "index.cnxml" File to an External XML Editor)
- Edit the index.cnxml file with the Edit-In-Place editor (Section 2.3.4.3: Edit the index.cnxml file with the Edit-In-Place Editor)
- Edit the index.cnxml file with the Full Source editor (Section 2.3.4.4: Edit the "index.cnxml" File with the Full Source Editor)

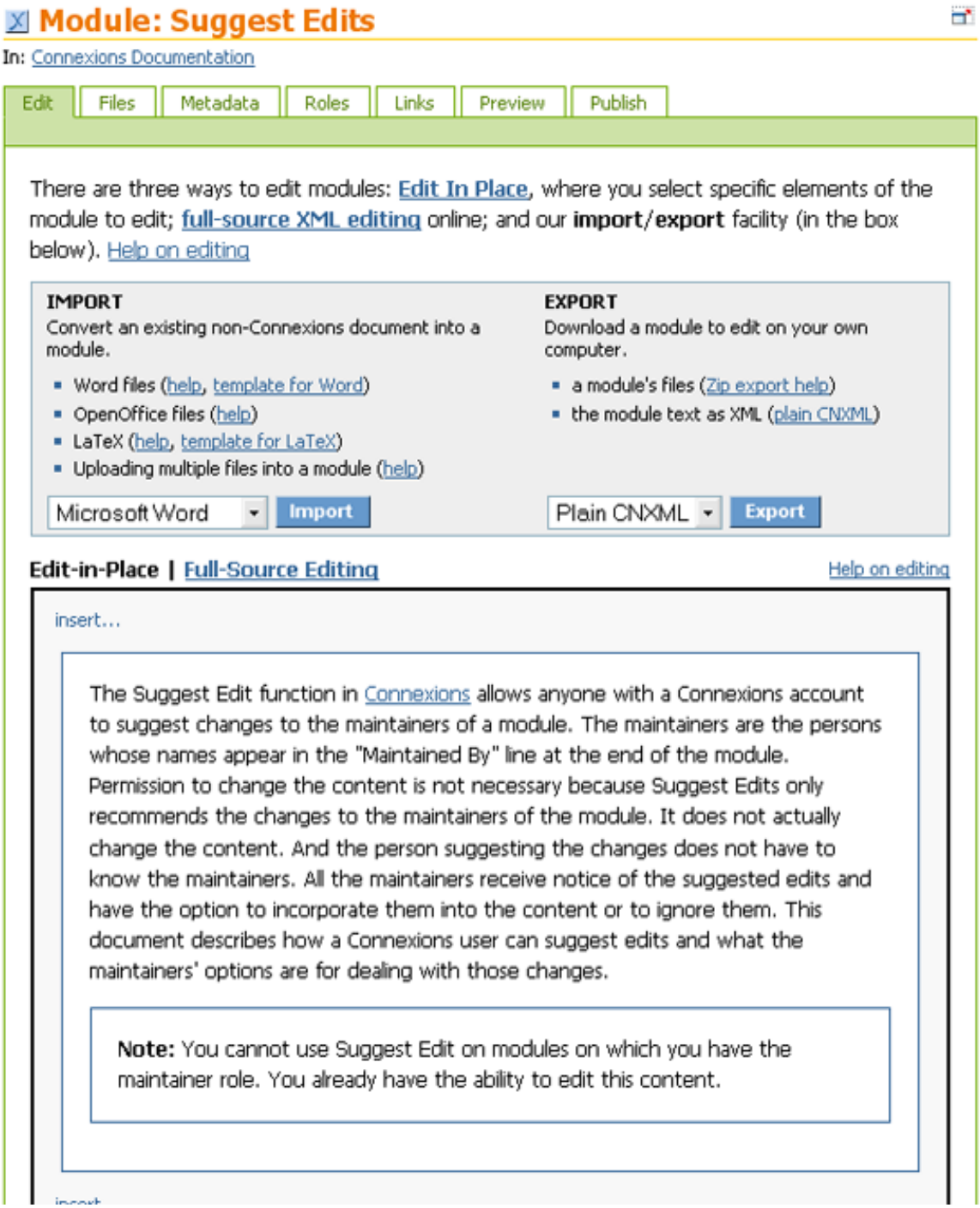


Figure 2.16: The "Edit" tab displaying contents of the "index.cnxml" file in the Edit-In-Place editor mode.

2.3.4.1 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[12] , OpenOffice Writer[13] , XMLSpy/Authentic[14], and Plain XML.
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.

2.3.4.2 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[15] **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[16], 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 2.3.4.1: Import Content into the "index.cnxml" File) mentioned earlier in this module.

2.3.4.3 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.

2.3.4.3.1 Inserting Content in an Empty Module

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

[12] http://cnx.rice.edu/help/UsingMSWord
[13] http://cnx.org/help/UsingOOWriter
[14] "Editing CNXML with Altova's Authentic" <http://cnx.org/content/m11160/latest/>
[15] http://www.altova.com
[16] "Editing CNXML with Altova's Authentic" <http://cnx.org/content/m11160/latest/>

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

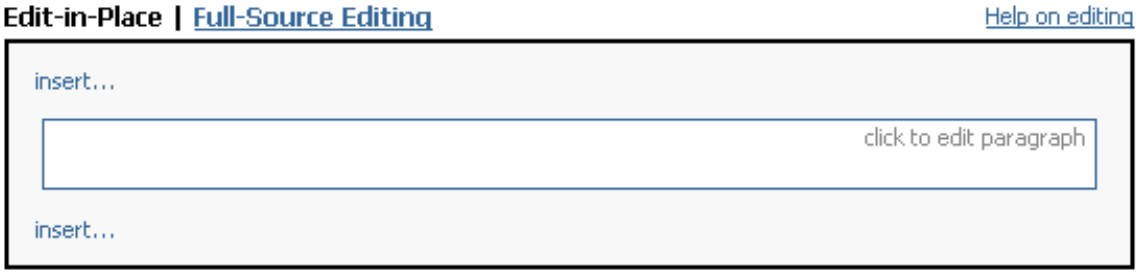


Figure 2.17: The "Click To Edit Paragraph" text box in the Edit-In-Place editor.

   The white box is replaced by a blue editing box.
2. Type the text you want to add in the blue editing box.

Figure 2.18: 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.

2.3.4.3.2 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.

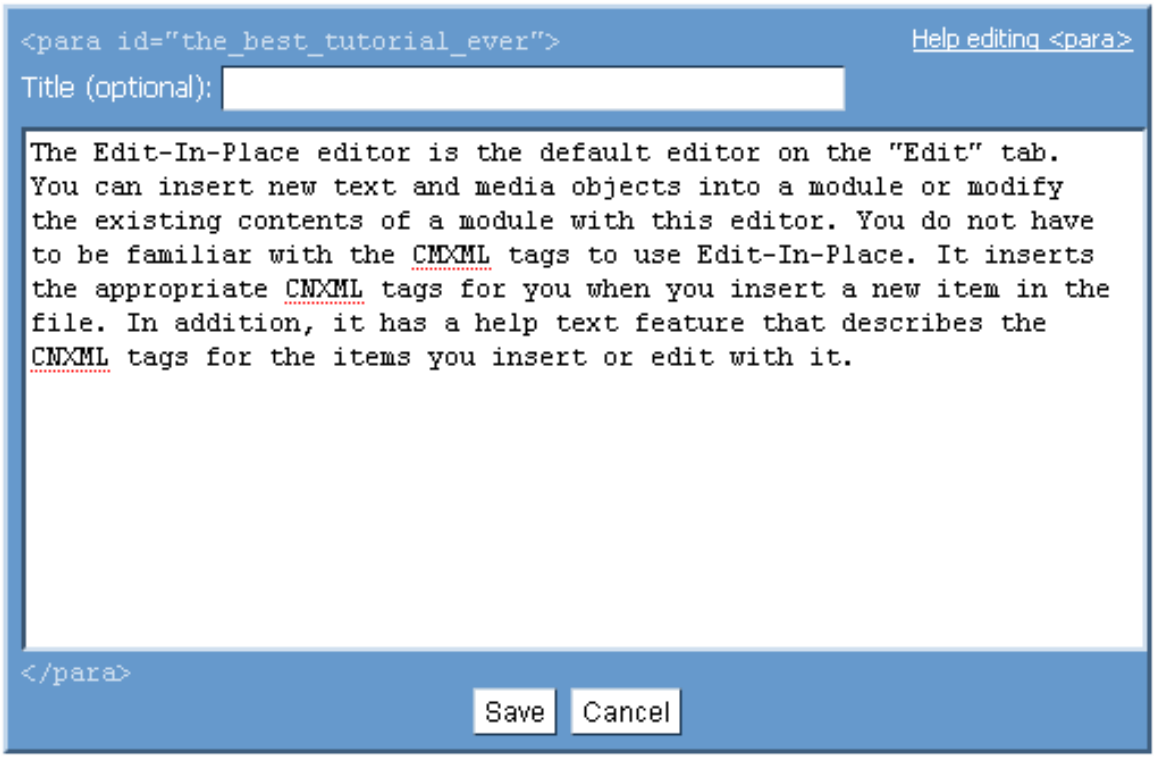


Figure 2.19: The blue text box in the Edit-In-Place editor.

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.
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.

2.3.4.3.3 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.

2.3.4.4 Edit the "index.cnxml" File with the Full Source Editor

You can edit the content and CNXML tags in the "index.cnxml" 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.cnxml" file are displayed in the Full Source editor mode.

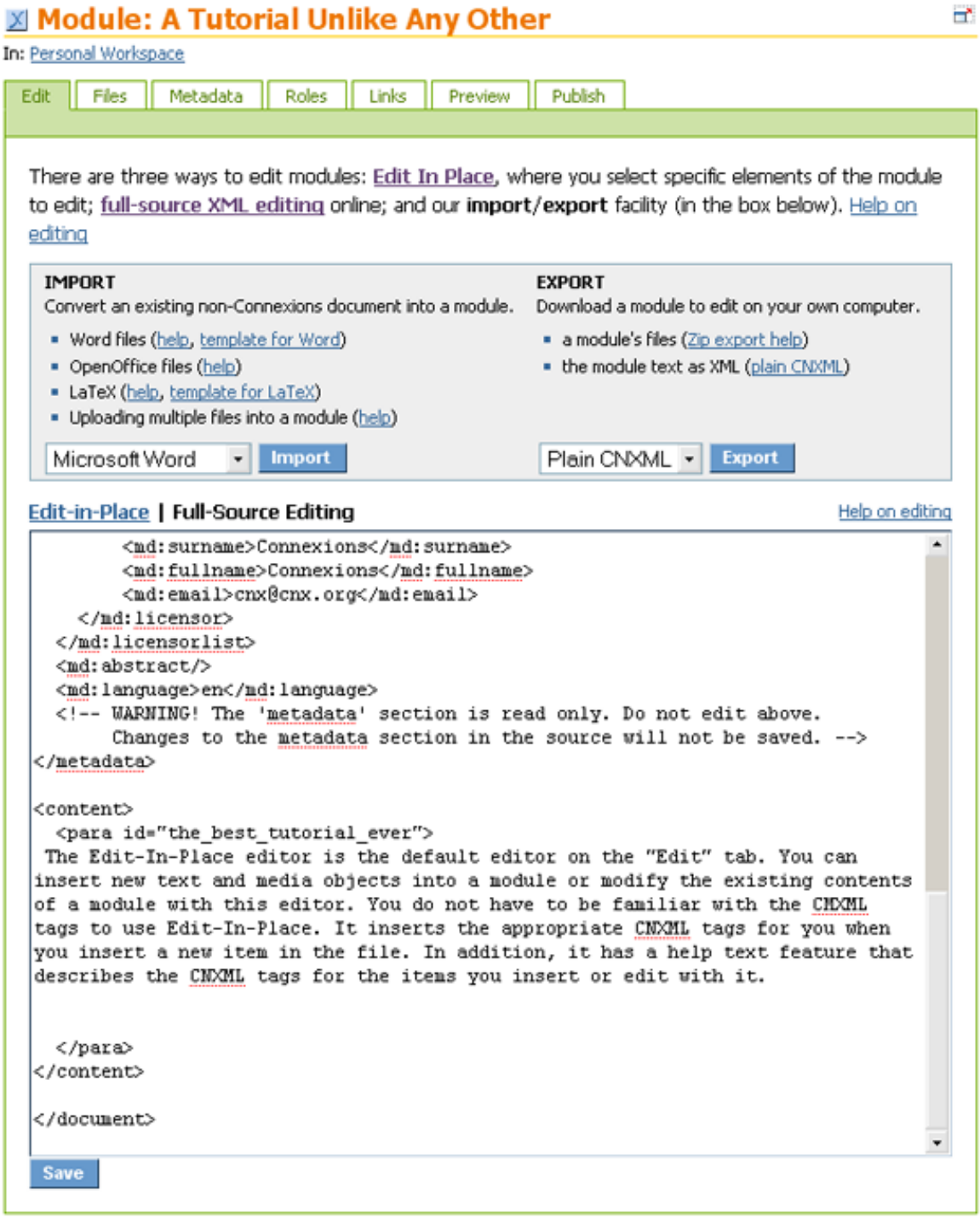


Figure 2.20: The "Edit" tab displaying the contents of the "index.cnxml" file in the Full Source editor mode.

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 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 (Section 2.3.12: Publishing the Module) later in this module.

2.3.4.4.1 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 2.21) appears when you click `Save`. Your changes are not saved until the errors are corrected.

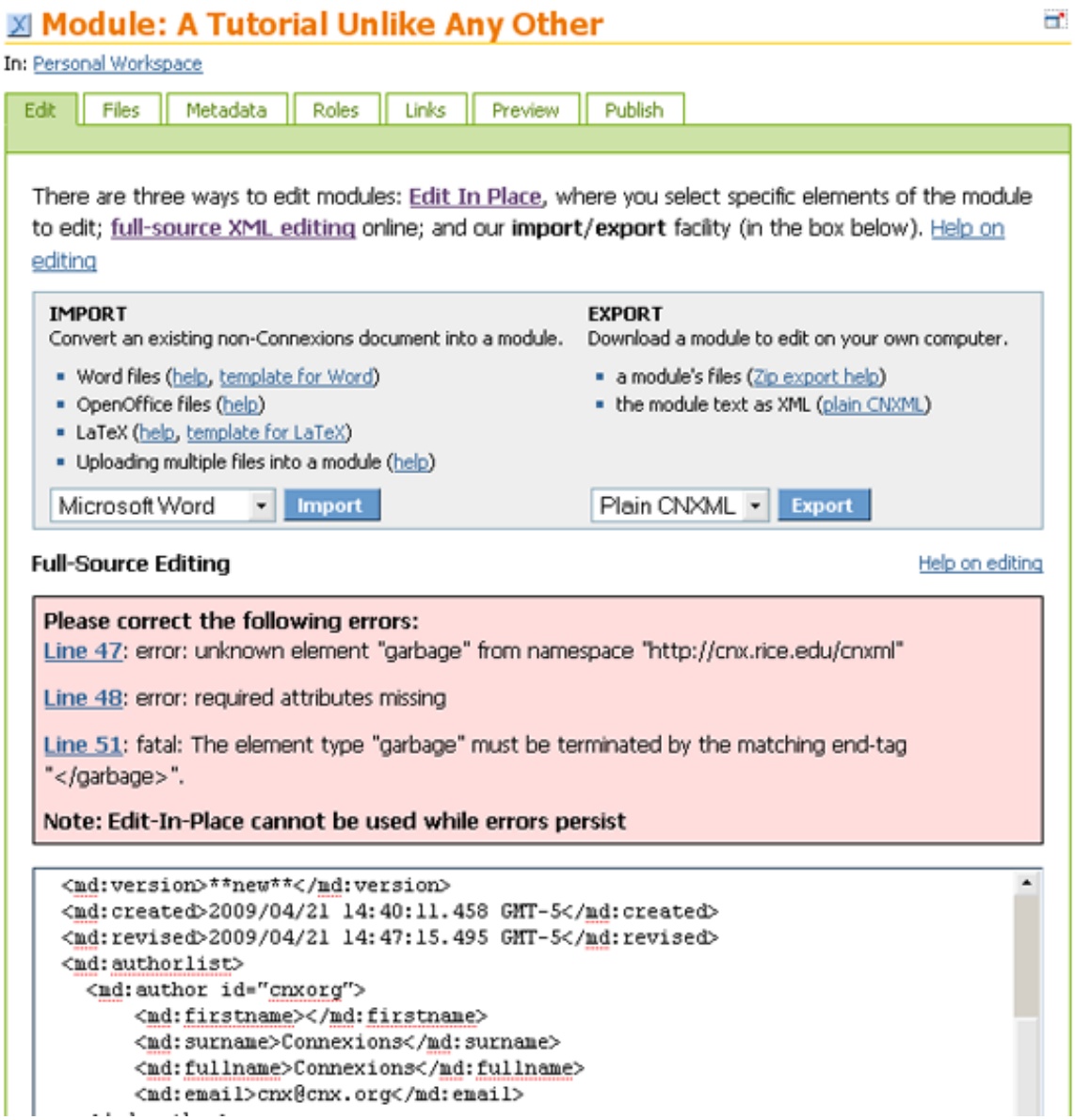


Figure 2.21: Example of validation errors in the Full Source editor

A description of the errors appears in the information box that appears above the content.

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

2.3.5 Adding Associated Files to a Module

In addition to the "index.cnxml" file, a module can contain other files (such as images and supplemental downloads) that are associated with the module. You can add these files from the "Files" tab. The file must exist before you add them to the module. When you add these items, you create an instance of the item in Connexions.

2.3.5.1 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 2.1.3.5: Renaming Items in a Work Area) 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 2.1.3.5: Renaming Items in a Work Area) 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 2.4) must have a recognized MIME (Multipurpose Internet Mail Extension) type.

2.3.5.2 Adding a File to a Module

To add a file to a module, use the following steps:

1. From the Files page, click the `Add New File` button. A new page will load.
2. Click `Browse`. The "File Upload" dialog box displays.
3. Locate the appropriate file and click on its filename.
4. Click `Open`. The file path appears in the text box next to `Browse`.
5. Click `Save` to upload the file into Connexions. A confirmation message will display, along with a preview if the file is an image.

2.3.5.3 Renaming, Cutting, Copying, and Removing Items in a Module

You can rename, cut, copy, or remove any item that you add to a module. These operations are done at the module level in the same way as they are done at the work area level. See Removing Items from a Work Area (Section 2.1.3.4: Removing Items from a Work Area), Renaming Items in a Work Area (Section 2.1.3.5: Renaming Items in a Work Area), and Cutting or Copying Items in a Work Area (Section 2.1.3.6: Cutting or Copying Items in a Work Area), for a description of how to do these operations in work areas.

NOTE: You should not rename, cut, copy, or paste the "index.cnxml" file.

2.3.6 Editing Metadata

To enable users to search the Connexions Content Commons for a module, Connexions must have some information on the module. It is **very important** that you insert metadata (Figure 2.22) into the module.

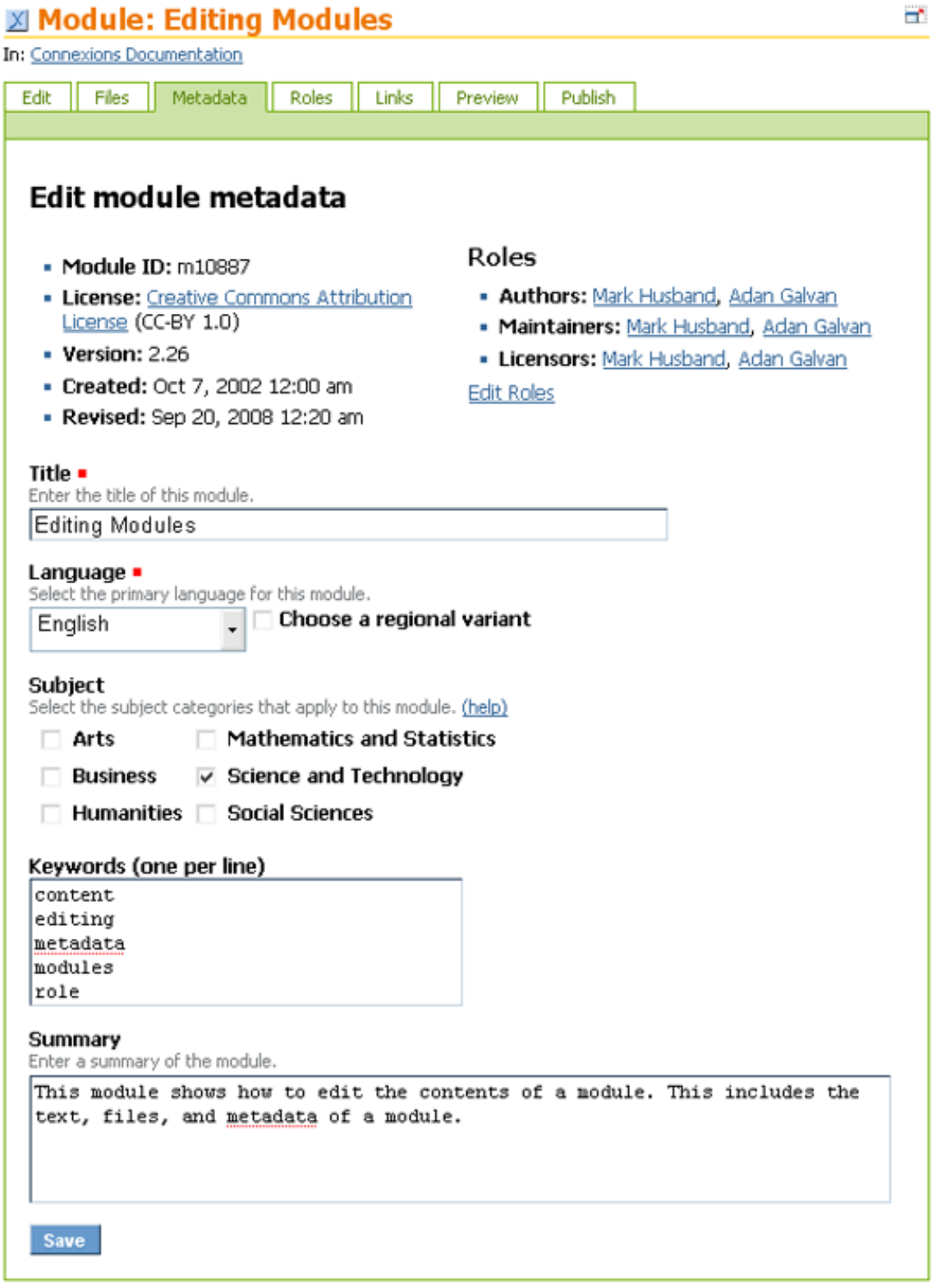


Figure 2.22: Example "Metadata" tab with entries for name, keywords, and abstract.

To insert metadata, use the following steps:

1. Click the "Metadata" tab to display it.
2. Type the name you want to give the module in the "Title" field. This name will appear wherever the module title appears in Connexions.
3. Select the primary language of the module in the "Language" dropdown menu. You may also specify a regional variant of the language by clicking the checkbox next to "Choose a regional variant" and then selecting the appropriate item.
4. Click on the boxes corresponding to the subject matter of the module under the "Subject" heading. You may select more than one subject.
5. Type the topic related terms that a user may use in a search for the module in the "Keywords" field. Enter each term on a separate line.
6. Type a brief description of the module topic in the "Summary" field. You may markup the summary with inline CNXML. Do not repeat the title of the module in the summary. The summary displays under the title in the module and restating the module title in the abstract is redundant.
7. Click `Save`. A message that the metadata was updated displays.

2.3.7 Editing Module Roles

As creator of a module, by default you hold all three roles[17] : author, maintainer, and copyright holder.

- **Authors** receive credit for writing the material. The author is the person who is responsible for the creation of the work, but may or may not be the person who actually enters the work into the Connexions. Any module work 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. 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.
- **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 modules, 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. The "Roles" tab has a search feature for locating these people. The current collaborators on the module are listed by role under the heading "Current Roles" on the "Roles" (Figure 2.23) tab. You can change the roles assigned to these people by selecting or deselecting the check boxes in the

[17] http://cnx.org/help/reference/roles

"Authors", "Maintainers", "Copyright Holders", "Editors" and "Translators" columns under the heading "Change Roles". The following paragraphs describe how to grant or update roles.

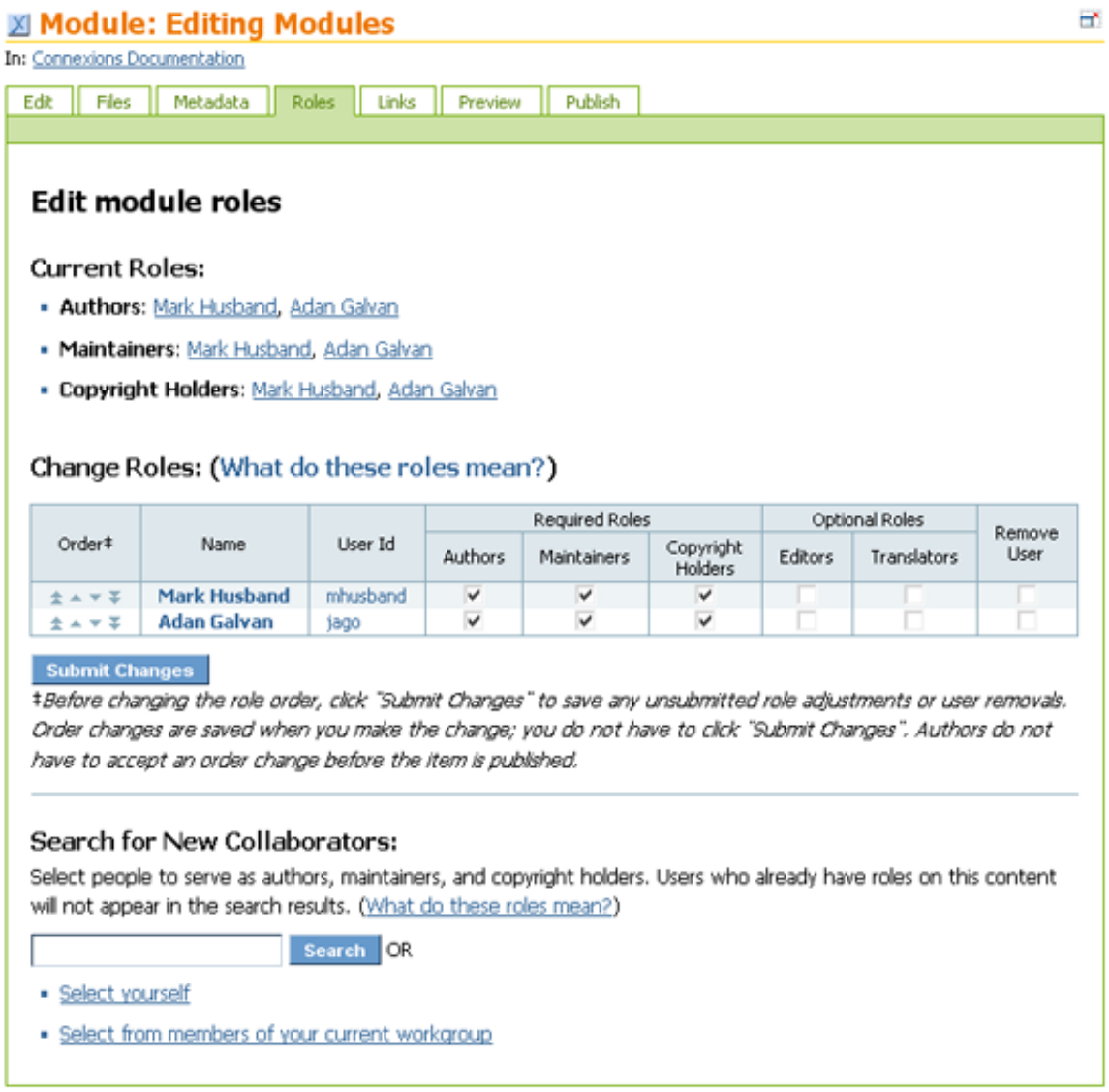


Figure 2.23: Example "Roles" tab.

When you grant or update roles for a collaborator in a module in a workgroup, these changes do not take affect until that person accepts the role change and you publish the module. The prospective collaborator must accept the role because Connexions provides its account holders with the ability to accept or reject a role in another person's module. If there are any pending roles (roles not yet accepted or rejected) for a module, Connexions will not let you publish that module. Once the collaborator accepts the role or updated role, you must publish the module for Connexions to recognize the collaborator. Connexions looks to the

information in a published module to see who is an author, maintainer, copyright holder, editor, or translator for the module.

> NOTE: All pending role additions or changes must be accepted or rejected by the prospective authors, maintainers, and copyright holders before you can publish a module in Connexions. And you must publish a module with the new roles before Connexions will recognize the new roles.

2.3.7.1 Granting Roles to Someone Outside of Your Workgroup

To grant one or more roles[18] in a module to a Connexions account holder outside of the current workgroup, use the following steps:

1. Click the "Roles" tab to display it.
2. Type the first or last name of the person you want to grant a role to in the text box next to `Search`.
3. Click `Search`. The Connexions accounts are searched and any Connexions accounts that match the search string are displayed.
4. 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.
5. 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.
6. Click `Submit Changes`. The next time the new collaborator logs into Connexions, he or she will be notified of a pending role request. Connexions will not apply the roles to the new collaborator until he or she accepts the role request.

2.3.7.2 Granting Roles to a Member of Your Workgroup

To grant one or more roles[19] in a module to a member of your current workgroup, use the following steps:

1. Click the "Roles" tab to display it.
2. 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. Locate the name of the person or persons in the table that you want to grant roles to and click the check boxes in the Authors, Maintainers, and Copyright Holders columns to select the roles for that person.
4. Click `Add People`. The name of the newly added collaborator appears with the names of authors, maintainers, copyright holders, editors, or translators 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.

A notification of a pending role request 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.

2.3.7.3 Changing the Order of Appearance for Author Names

To change the order in which the authors' names will appear on the module, use the following steps:

[18] http://cnx.rice.edu/help/reference/roles/document_view
[19] http://cnx.rice.edu/help/reference/roles/document_view

1. Click the "Roles" tab to display it.
2. Verify that all users who should have an author role on the module are listed in the "Change Roles" table. Select or deselect the check boxes in the "Adjust Roles" columns to make any necessary additions or changes to the module roles.
3. Click `Submit Changes` to save the role changes. The next step will erase any role changes that you have not saved with the `Submit Changes` button.
4. Click the up or down arrows in the "Order" column to move an author up or down in the order of appearance in the module. 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.

To have an author's name appear first in the module, 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.

2.3.7.4 Changing Role Assignments

To change the roles[20] in a module that were previously granted to a user, 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 change roles.
3. Click on a check box that contains a check mark to remove the check mark and to remove that role from the person. Click on an empty check box to make a check mark appear and to grant that role to the person.
4. Click `Submit Changes`. The "Change Roles" table is updated to show the role changes and to note the role changes that are pending acceptance.

A notification of a pending role change request 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.

2.3.7.5 Removing a Collaborator

To remove a collaborator from a module, 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 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.

2.3.8 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 2.24) sidebar will appear in the upper right corner of your Connexions pages.

[20] http://cnx.rice.edu/help/reference/roles/document_view

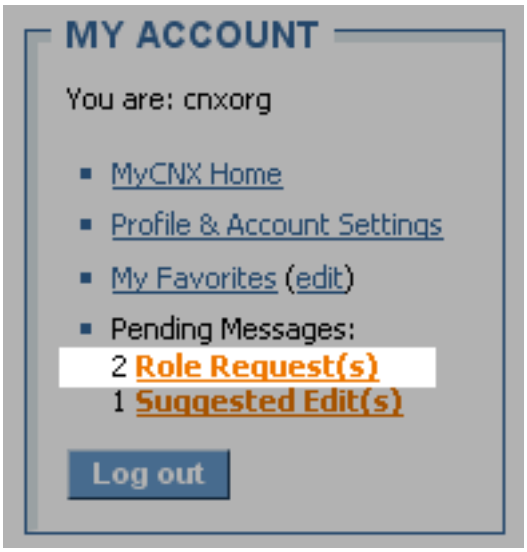
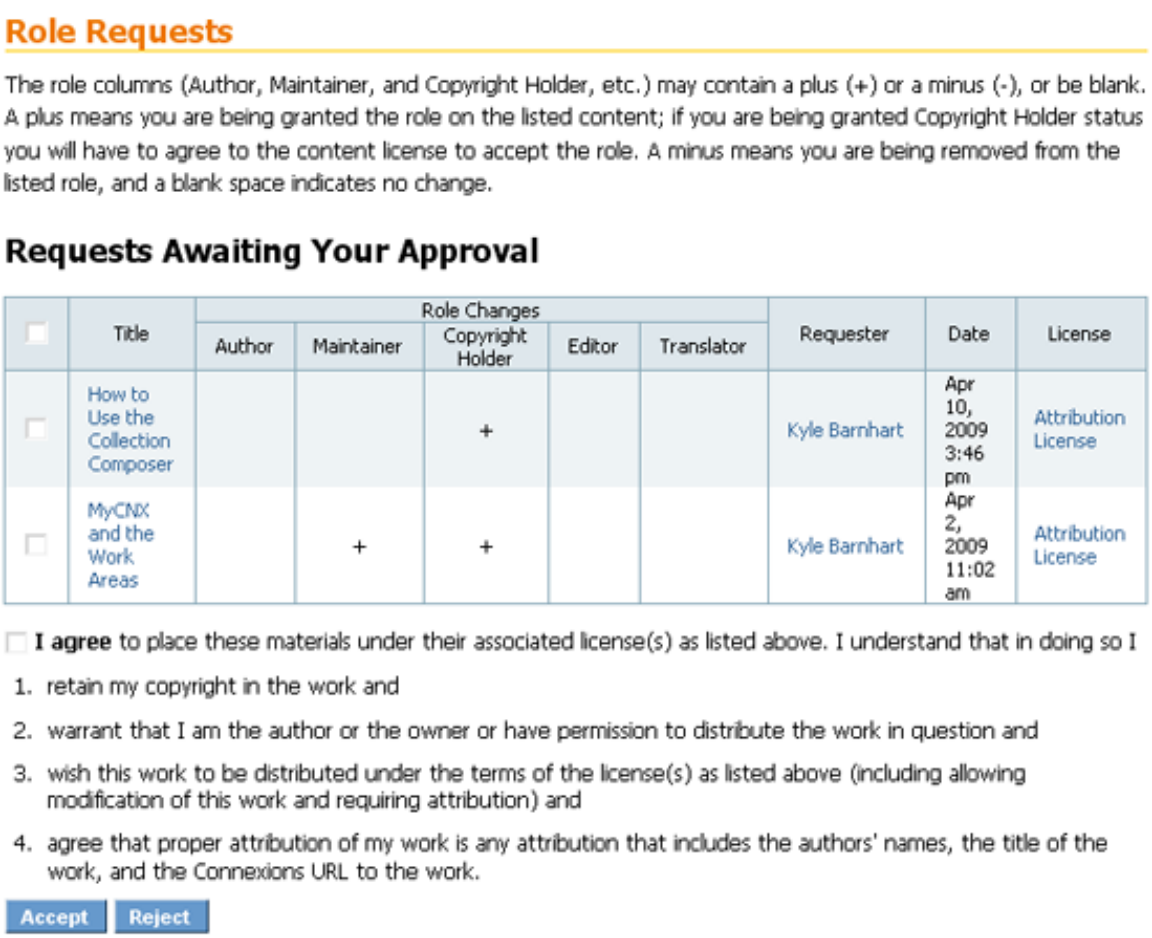


Figure 2.24: "Pending" sidebar with role requests.

Click the **Role Request** link to display the "Collaboration Requests" (Figure 2.25) screen. From this screen you can view and act upon:

- Requests awaiting your approval (Section 2.3.8.1: Requests Awaiting Your Approval)
- Requests you have made awaiting others' acceptance (Section 2.3.8.2: Requests You Have Made)
- Completed requests (Section 2.3.8.3: Completed Requests)



Role Requests

The role columns (Author, Maintainer, and Copyright Holder, etc.) may contain a plus (+) or a minus (-), or be blank. A plus means you are being granted the role on the listed content; if you are being granted Copyright Holder status you will have to agree to the content license to accept the role. A minus means you are being removed from the listed role, and a blank space indicates no change.

Requests Awaiting Your Approval

	Title	Role Changes					Requester	Date	License
		Author	Maintainer	Copyright Holder	Editor	Translator			
☐	How to Use the Collection Composer			+			Kyle Barnhart	Apr 10, 2009 3:46 pm	Attribution License
☐	MyCNX and the Work Areas		+	+			Kyle Barnhart	Apr 2, 2009 11:02 am	Attribution License

☐ **I agree** to place these materials under their associated license(s) as listed above. I understand that in doing so I

1. retain my copyright in the work and
2. warrant that I am the author or the owner or have permission to distribute the work in question and
3. wish this work to be distributed under the terms of the license(s) as listed above (including allowing modification of this work and requiring attribution) and
4. agree that proper attribution of my work is any attribution that includes the authors' names, the title of the work, and the Connexions URL to the work.

Accept Reject

Requests You Have Made

No requests pending

Requests You Have Made -- Completed

No requests completed

Figure 2.25: "Collaboration Requests" screen showing one request awaiting your approval and one awaiting another author's approval.

2.3.8.1 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 license is provided in the "License" column of the table.

2.3.8.2 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.

2.3.8.3 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.

2.3.9 Adding Featured Links to a Module

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 2.26) screen displays.

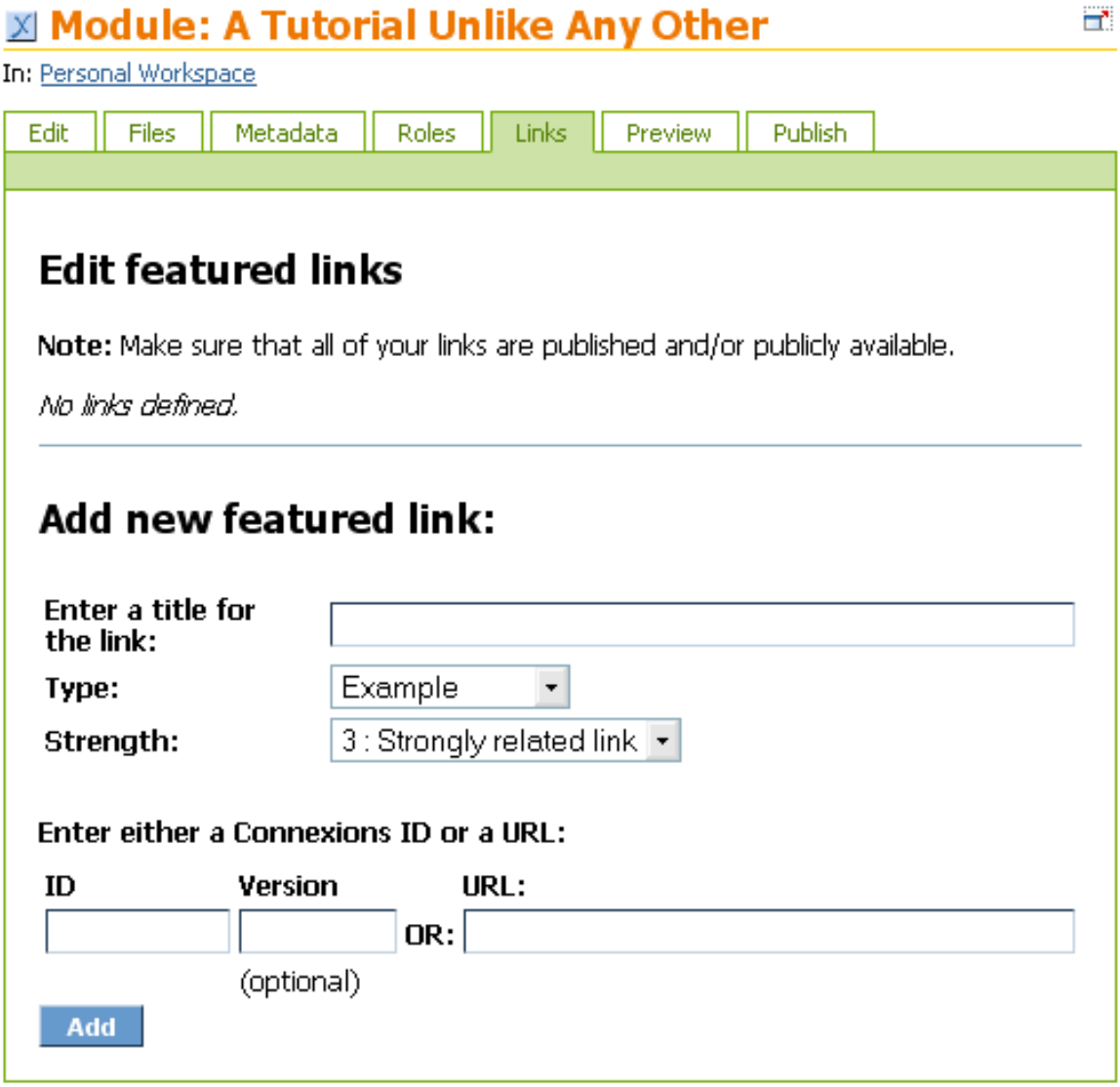


Figure 2.26: The "Links" tab.

From the "Links" tab you can:

- Add new links (Section 2.3.9.1: Adding New Links)
- Edit existing links (Section 2.3.9.2: Editing Existing Links)
- Delete a link (Section 2.3.9.3: Deleting a Link)

These links are viewable in the "Links" tab that displays with each module. The featured links appear under their specified category type, for example, the supplemental links are listed under the "Supplemental" heading.

2.3.9.1 Adding New Links

To add a new link to a module, use the following steps:

1. Type the label for the link in the "Name" field (required).
2. Click on the down arrow next to the "Link Type" field and select the type of link from the drop-down list.
3. Click on the down arrow next to the "Link Strength" field and select a number from the drop-down list for the strength of the link. 5 is the strongest link.
4. Identify the target of the link by typing in a module ID in the "Module ID" field or typing a URL in the "Link URL" field. You can optionally specify a version if you entered a module ID.
5. Click `Add` to add the link.

2.3.9.2 Editing Existing 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 "Name" field to change the label of the link.
2. Click on the down arrow next to the "Link Type" field and select another type of link from the drop-down list.
3. Click on the down arrow next to the "Link 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.

2.3.9.3 Deleting a 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.

2.3.10 Previewing the 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 2.3.10.1: Previewing Modules On-Line)
- Preview the module in a print version (PDF file) (Section 2.3.10.2: Previewing a Print Version of the Module)
- View the change summary (Section 2.3.10.3: Viewing Change Summary)
- View the markup source for the module (Section 2.3.10.4: Previewing the code source)

2.3.10.1 Previewing Modules On-Line

To view a module on-line, 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 does not contain the links and tabs at the beginning and the links at the end of the module. 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.

2.3.10.2 Previewing a Print Version of the Module

The print version of a module is formatted differently than the on-line version. This difference is necessary because of the differences in the on-line 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.

2.3.10.3 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 2.27). This option is useful when you checked out a module some time ago and do not remember what changes you made to it.

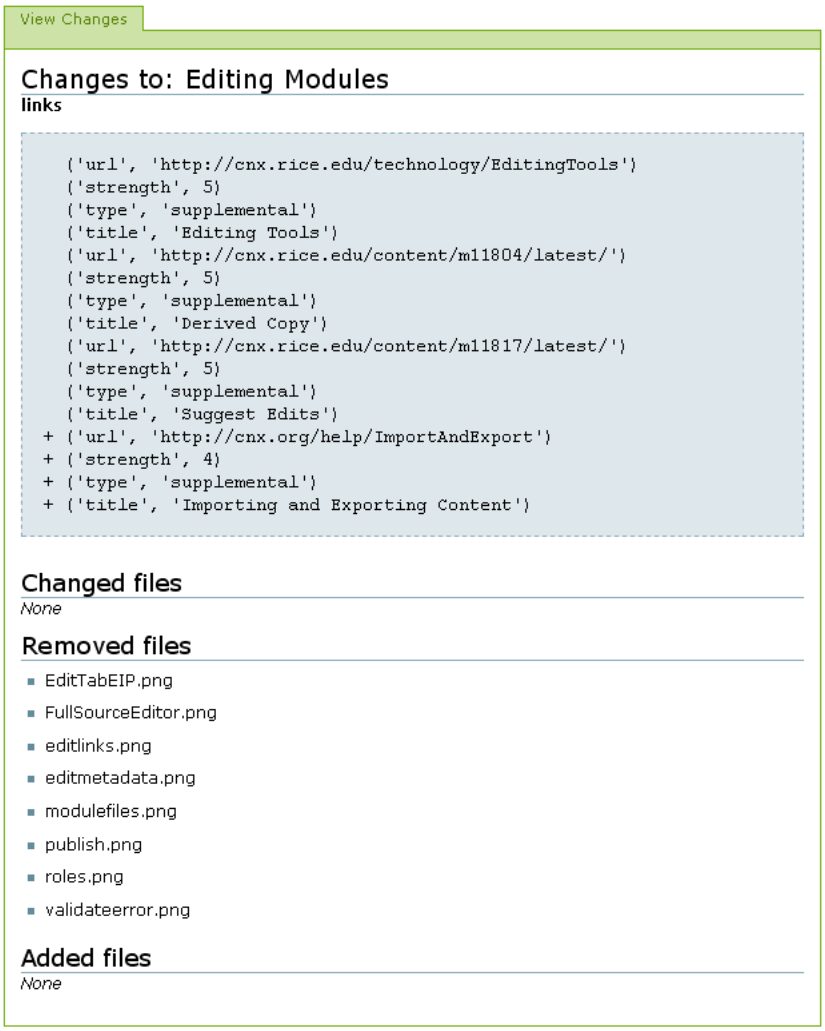


Figure 2.27: The "Changes to" summary.

The "Change To" 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.

2.3.10.4 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.

NOTE: This feature is useful when you are trying to find a markup error.

2.3.11 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. The module will revert back to the state it was in when you checked it out.

> NOTE: If the module is new and has never been published, you will lose all the content you entered and the module will be blank.

2.3.12 Publishing the Module

To make your module or your changes to an existing module viewable by anyone on the Internet, you must publish the module in Connexions. To publish a module, use the following steps:

1. Click on the "Publish" tab. The "Publish Module" (Figure 2.28) screen displays. This screen contains a list of reminders for you to check before you publish the module.
2. Type a brief description of your work on the module in the "Description of Changes" text box.
3. Click `Publish`. Your work on the module is entered into the Connexions Content Commons. Once in the Content Commons, anyone with Internet access can view the module.

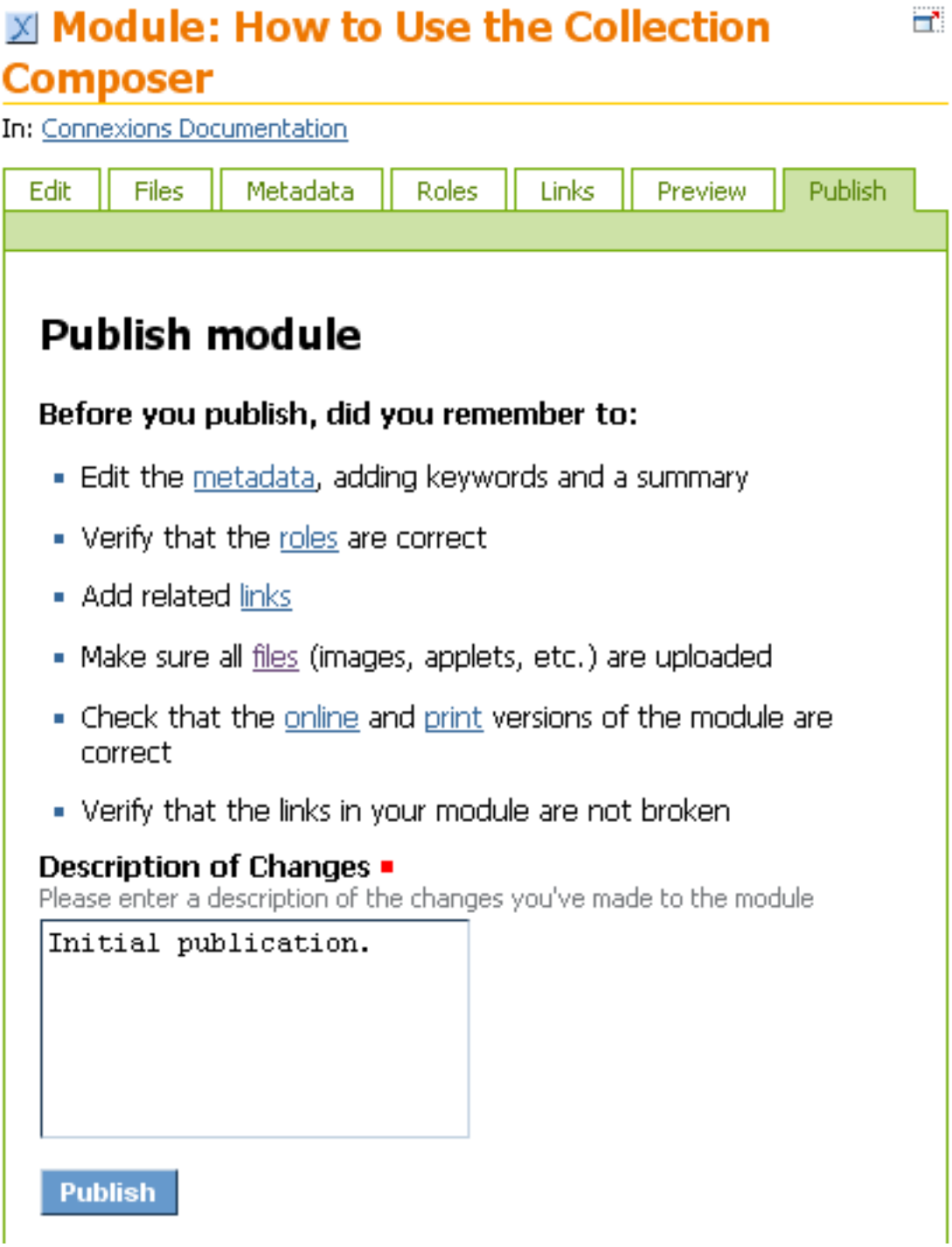


Figure 2.28: The "Publish" tab.

A new module would not necessarily have editing changes, so in the "Description of Changes" text box you might enter a comment like, "Created a new module".

> NOTE: The act of publishing a module is different from clicking on the `Save` button in an editing session. When you save your changes, only you and the people who have access to your workgoup can see your changes. When you publish a module, the entire world can view your work on the Internet.

2.4 Adding Multimedia to Your Connexions Content[21]

Contents

- Introduction (Section 2.4.1: Introduction)
- Making Multimedia Accessible for All Users (Section 2.4.2: Making Multimedia Accessible for All Users)
- Embedding Media Elements in Your Module (Section 2.4.3: Embedding Media Elements in Your Module)
  - The <media> Element (Section 2.4.3.1: The <media> Element)
  - The <figure> Element (Section 2.4.3.2: The <figure> Element)
  - The <subfigure> Element (Section 2.4.3.3: The <subfigure> Element)
  - Including Alternate Media for Print (Section 2.4.3.4: Including Alternate Media for Print)
- Media Subtypes (Section 2.4.4: Media Subtypes)
  - Images (Section 2.4.4.1: Images)
  - Video (Section 2.4.4.2: Video)
  - Flash objects (Section 2.4.4.3: Flash Objects)
  - Audio (Section 2.4.4.4: Audio Files)
  - Java Applets (Section 2.4.4.5: Java Applets)
  - LabVIEW Demonstrations (Section 2.4.4.6: LabVIEW Demonstrations)
  - Parameters (Section 2.4.4.7: Parameters)
- Using Third-Party Services (Section 2.4.5: Using Third-Party Services)
  - Embedding YouTube Videos (Section 2.4.5.1: Embedding YouTube Videos)
  - Embedding SlideShare Presentations (Section 2.4.5.2: Embedding SlideShare Presentations)

2.4.1 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.

> 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.

[21] This content is available online at <http://cnx.org/content/m12660/1.16/>.

Connexions supports the use of any media format that has a valid MIME type (see this page[22] 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[23] .

2.4.2 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.
- 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`:

[22] http://www.ietf.org/rfc/rfc1341.txt?number=1341
[23] techsupport@cnx.org

```
<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.

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[24].

2.4.3 Embedding Media Elements in Your Module

2.4.3.1 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:

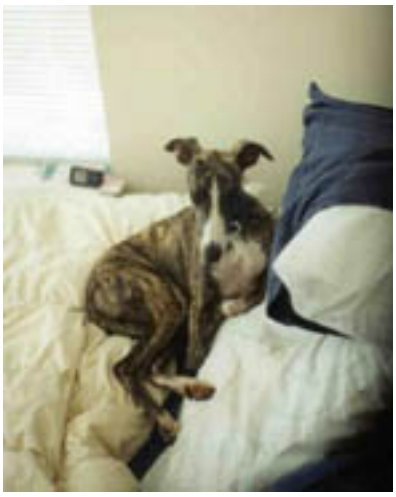
- A unique id
- Alt text
- A media subtype element, such as `<image>`

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

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

[24] "Accessibility Features of Connexions" <http://cnx.org/content/m17212/latest/>

This code results in the following display:



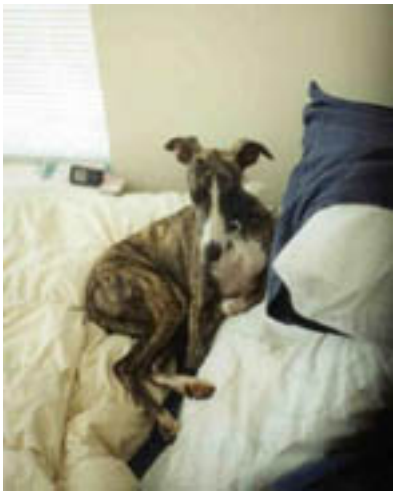
Note that the resulting image displays **inline**; when included as the child of another element, such as <para>, the media element will display in line with the surrounding text by default:

```
<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:

- <image>
- <video>
- <flash>

- <audio>
- <java-applet>
- <labview>
- <text>
- <download>

2.4.3.2 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 it's 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:

```
<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:

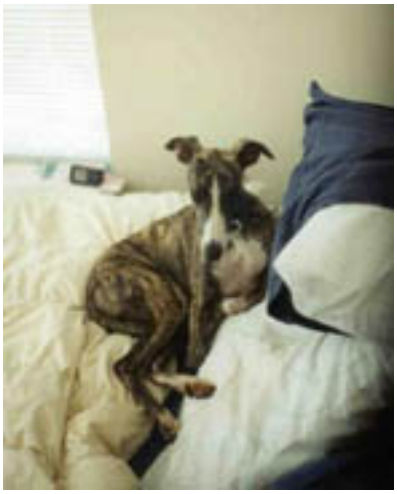


Figure 2.29

You'll notice that the figure is labeled as "Figure 2.29" - 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 2.29.

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:

- <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.

```
<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:

Mi Perro Benny

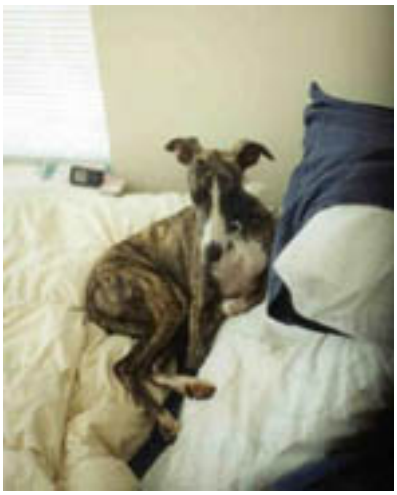


Figura 2.30: Este es mi perro Benny haciendo lo que hace mejor.

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.

2.4.3.3 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:

```
    <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:

(a) Blue on Blue (b) Orange on White

Figure 2.31: Two images displayed horizontally in one figure. (a) Subfigure 1. (b) Subfigure 2.

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:

(a) Blue on Blue

(b) Orange on White

Figure 2.32: Two images displayed vertically in one figure. (a) Subfigure 1. (b) Subfigure 2.

TIP: Only use subfigures when the images are related to one another.

2.4.3.4 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. The first element will be used in the online version of the module, while the second will be used for the print version.

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.

```
<figure id='printimage'>
   <media id="print_image_example">
      <image mime-type='image/png' src='image.png'/>
      <image mime-type='application/postscript' src='image.eps'/>
   </media>
</figure>
```

TIP: You can use any two media objects in this way to create print alternatives. When two media subtype objects are included inside a `media` element, the first will be used for the online version and the second will be used for the print version.

2.4.4 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.

2.4.4.1 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[25] can be used in a module.

2.4.4.1.1 Embedding Images

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

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

[25] http://www.duke.edu/websrv/file-extensions.html

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 2.33: Example Figure) for the previous CNXML example is:

Example Figure

Figure 2.33: A graphic image displayed as a figure.

2.4.4.1.2 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.
- 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.

2.4.4.1.2.1 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:

```
 <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 2.34: The photograph of the Taj Mahal in this figure and the following figures was taken by Steve Evans[26]. 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):

```
 <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:

[26] http://www.flickr.com/photos/babasteve/

Figure 2.35

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.

2.4.4.1.2.2 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):

```
 <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[27] . 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.

2.4.4.1.2.3 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 fie.
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:

```
    <figure id='thumbnail'>
   <media id="thumbnailmedia" alt="The Taj Mahal (thumbnail version)">
      <image mime-type='image/jpeg'
             src='TajMahal_medium.jpg'
             thumbnail='TajMahal_Thumb.jpg'/>
      </image>
   </media>
</figure>
```

This code results in the following linked image:

[27] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/teTeX/latex/latex2e-html/ltx-86.html

Figure 2.36

2.4.4.2 Video

2.4.4.2.1 Embedding Videos Attached to the Module

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

```
 <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" />
   </video>
</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 2.37: The Creative Commons movie: "Building on the Past". Click the Play button to start the movie.

2.4.4.3 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 2.38) appears in your module the same way in which a movie or an image appears:

This media object is a Flash object. Please view or download it at
<flash2.swf>

Figure 2.38: "Welcome to Connexions" example Flash object.

2.4.4.4 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:

```
<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 2.39) for the previous CNXML example is:

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

Figure 2.39: A chromatic scale performed on clarinet by Michael Lawrence.

2.4.4.5 Java Applets

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

```
<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>
   <caption>3D Animation of a Complex Sinusoid. Click the Run button to start the animation.</caption>
</figure>
```

The actual display (Figure 2.40) for the previous CNXML example is:

This media object is a Java applet. Please view or download it at
<#java_example>

Figure 2.40: 3D Animation of a Complex Sinusoid. Click the Run button to start the animation.

If the graphic does not appear in the figure above, you may need to install or upgrade the Java plug-in on your computer. Please go to java.com[28] to download the latest version of the Java plug-in.

2.4.4.6 LabVIEW Demonstrations

For information on including LabVIEW demonstrations in your module, please see Creating LabVIEW demonstrations for Connexions[29].

2.4.4.7 Parameters

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

Here is an example of a `param` tag for an audio (wav) object:

```
  <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"/>
```

[28] http://java.com
[29] "Creating LabVIEW demonstrations for Connexions" <http://cnx.org/content/m11601/latest/>

```
    </audio>
  </media>
</figure>
```

The actual display (Figure 2.41) for the previous example is

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

Figure 2.41

2.4.5 Using Third-Party Services

It is possible to embed videos and other multimedia files from sites such as YouTube[30] or SlideShare[31] 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 `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.

2.4.5.1 Embedding YouTube Videos

If you choose to include YouTube videos in your module, you will need to first copy the information provided in the **Embed** HTML snippet provided by YouTube for the video you wish to use. As an example, consider this video[32] of Neil Armstrong walking on the moon. The Embed snippet for this video is as follows:

```
<object width="425" height="344">
 <param name="movie" value="http://www.youtube.com/v/RMINSD7MmT4&hl=en&fs=1&rel=0"></param>
 <param name="allowFullScreen" value="true"></param>
 <param name="allowscriptaccess" value="always"></param>
 <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">
 </embed>
</object>
```

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**:

```
 <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">
```

[30] http://www.youtube.com/
[31] http://www.slideshare.net
[32] http://www.youtube.com/watch?v=RMINSD7MmT4

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 <flash>, not <video> - this is because the YouTube player is itself a flash object. 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 **mime-type** attribute is set to "application/x-shockwave-flash" - this is the format used by YouTube videos.
- 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 always of the form `http://www.youtube.com/v/[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 `&`. 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,

  ```
  src="http://www.youtube.com/v/RMINSD7MmT4&hl=en&fs=1&rel=0"
  ```

  becomes

  ```
  src="http://www.youtube.com/v/RMINSD7MmT4&amp;hl=en&amp;fs=1&amp;rel=0"
  ```

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

```
<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">
  <flash id="yt-video"
         mime-type="application/x-shockwave-flash"
         src="http://www.youtube.com/v/RMINSD7MmT4&amp;hl=en&amp;fs=1&amp;rel=0"
         width="425"
         height="344">
   <param name="allowscriptaccess" value="always" />
   <param name="allowsfullscreen" value="true" />
  </flash>
 </media>
</figure>
```

And results in the following:

One Small Step for (a) Man

This media object is a Flash object. Please view or download it at
<http://www.youtube.com/v/RMINSD7MmT4&hl=en&fs=1&rel=0>

Figure 2.42

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 `&`.
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.

2.4.5.2 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[33] taken from our standard Connexions presentation. The Embed snippet for this presentation is as follows: **Connexions: Create Globally, Educate Locally**

```
   <object style="margin:0px" width="425" height="355">
 <param name="movie"
  value="http://static.slidesharecdn.com/swf/ssplayer2.swf?doc=cnxdemo-090820214427-phpapp02&stripped_t
 <param name="allowFullScreen" value="true"/>
 <param name="allowScriptAccess" value="always"/>
 <embed
  src="http://static.slidesharecdn.com/swf/ssplayer2.swf?doc=cnxdemo-090820214427-phpapp02&stripped_tit
  type="application/x-shockwave-flash" allowscriptaccess="always" allowfullscreen="true"
  width="425" height="355">
 </embed>
</object>
```

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**:

```
 <embed
  src="http://static.slidesharecdn.com/swf/ssplayer2.swf?doc=cnxdemo-090820214427-phpapp02&stripped_tit
  type="application/x-shockwave-flash" allowscriptaccess="always" allowfullscreen="true"
  width="425" height="355">
 </embed>
```

The CNXML example below illustrates how to embed a SlideShare 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 SlideShare website**. The embed URL is the one found in the code snippet as shown above and is of the form `"http://static.slidesharecdn.com/swf/ssplayer2.swf?doc=[PresentationID]"` (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 the `&` characters with `&`. 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,

```
    src="http://static.slidesharecdn.com/swf/ssplayer2.swf?doc=cnxdemo-090820214427-phpapp02&
stripped_title=cnxdemo"
```

[33] http://www.slideshare.net/cnxorg/cnxdemo

becomes

```
src="http://static.slidesharecdn.com/swf/ssplayer2.swf?doc=cnxdemo-090820214427-phpapp02&amp;
stripped_title=cnxdemo"
```

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

```
<figure id="slidesharefigure">
 <media id="slidesharemedia" alt="Slide show introducing the ideas behind Connexions.">
  <flash id="slideshareflash" height="355" width="425" mime-type="application/x-shockwave-flash"
   src="http://static.slidesharecdn.com/swf/ssplayer2.swf?doc=cnxdemo-090820214427-phpapp02&amp;strippe
   <param name="allowScriptAccess" value="always"/>
   <param name="allowFullScreen" value="true"/>
  </flash>
 </media>
</figure>
```

And results in the following:

Connexions: Create Globally, Educate Locally

This media object is a Flash object. Please view or download it at
<http://static.slidesharecdn.com/swf/ssplayer2.swf?doc=cnxdemo-090820214427-phpapp02&stripped_title=cnxdemo>

Figure 2.43

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 SlideShare.
Step 3. Escape the ampersand characters in the URL by replacing `&` with `&`.
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.

2.5 Derived Copy[34]

2.5.1 Introduction

The Derive Copy function in Connexions allows you to create a derivative work that is based on any published module. A derivative work starts with a copy of an existing module, to which you make changes and then

[34] This content is available online at <http://cnx.org/content/m11804/1.5/>.

publish as your own module. The Derive Copy process is not a method of collaboration. If you want to recommend changes to an author of an existing module, use the Suggest Edits function.

> WARNING: Any derivative work you create is a snapshot of the original at the time you created the work. If at a later time the original module is updated, your copy will not reflect those updates.

Derivative works are permitted under the Creative Commons Attribution License[35] , which applies to all the Connexions content. When you publish a derived copy, Connexions automatically adds an attribution statement that identifies the original work and its author, which is legally required by the Creative Commons Attribution License.

2.5.2 Uses

There are two primary uses for the Derive Copy function:

- Creating a new module by using an existing module as a template. For example, you find an existing module that is constructed in a similar way to a module you intend to create. You can create a derived copy and use it as a starting point for your new module.
- Creating a module that is similar to an existing module, but with some changes. For example, you find an existing module that you want to use in your new course, but you want some changes in the module first and you do not have permission to modify the module. You contact the original author and suggest (Section 2.6) your changes. The original author is not sympathetic to your suggestions because they take the module in a direction that does not match their intent for the module. You feel strongly about getting your changes out to the world, so you create a derived copy of the module, insert your changes, and publish it as your own module.

2.5.3 Create a Derived Copy

You create a derived copy from the module editor, in your workspace or a workgroup.

1. Search[36] the Content Commons for the module you want to copy.
2. Add the module to your workspace or workgroup.
3. Click on the module title to open the module editor.
4. Click `Derive Copy` under the "Actions" heading in the "Status" sidebar. The "Derive a Copy from" screen displays.
5. Select the check box at the bottom of the screen to signify that you agree to the application of the Content Commons Attribution license to your new module.
6. Click `Yes, Create Copy`. The "File" tab of the module editor displays, listing the derived copy. At this point the new module is in the created state, in your workspace or workgroup.

All the content in the new module is exactly the same as the content in the original module. The metadata is the same, except for the roles. The roles now show you as an author, maintainer, and copyright holder in addition to the accounts from the original module. You can edit the content of the module, its files, metadata, roles, and links as necessary, just as you can for any module that you create. You may submit a request to remove the original authors from the module, but they will have to approve the request like any other role change.

When you publish your derived copy module, Connexions automatically adds a statement to the module that gives attribution to the original module and author. This statement is a link to the original module, so the reader can view the module on which your derived copy was based.

[35] http://creativecommons.org/licenses/by/1.0/
[36] http://cnx.rice.edu/content/search?words=

2.6 Suggest Edits[37]

The Suggest Edit function in Connexions[38] allows anyone with a Connexions account to suggest changes to the maintainers of a module. The maintainers are the persons whose names appear in the "Maintained By" line at the end of the module. Permission to change the content is not necessary because Suggest Edits only recommends the changes to the maintainers of the module. It does not actually change the content. And the person suggesting the changes does not have to know the maintainers. All the maintainers receive notice of the suggested edits and have the option to incorporate them into the content or to ignore them. This document describes how a Connexions user can suggest edits and what the maintainers' options are for dealing with those changes.

> NOTE: You cannot use Suggest Edit on modules on which you have the maintainer role. You already have the ability to edit this content.

2.6.1 Suggesting Edits to Maintainers

The Suggest Edits function is a form of collaboration. It allows you to participate in the content of a module. Your edits could be as simple as correcting a spelling error, adding a link, and changing a keyword, or they could be as substantial as the addition of new material to expand the the module topic.

To suggest edits for a module in connections, use the following steps:

1. Check out the module into a workspace.
2. Access the module in the module editor and make the necessary changes. These edits can be changes to the text, images, links, roles, or metadata.
3. (Optional) Click the **Changes** link under the View heading in the "Status" sidebar to display a summary of your changes to the module.
4. Click the **Suggest Edit** link under the "Actions" heading in the "Status" sidebar. The "Suggest Edits" screen appears.
5. Type an explanation of your changes in the "Description of Edits" text box. This should be a general description and your reason for suggesting the changes. It will be forwarded to the maintainers.
6. Click `Send Edits`. The "Manage Files" screen reappears with the message that the edits were submitted to the maintainers. They have the option to use or ignore your suggestions. When they act on the suggestion, they must enter a description of their action that will be available to you when you check the status of your submission.

If you try to publish the module after you have edited it, the "Publish module" page displays a note that you do not have permission to publish a new version of this module. The page lists two options available to you: Suggest your edits to the module's maintainers or Create a new module of your own based on this one. If you choose the "Suggest your changes" option, the same "Suggest Edits" screen in step 3 above appears. If you choose the "Create your own module" option, the Derive a Copy (Section 2.5) screen appears.

2.6.2 Accepting or Rejecting the Suggested Edits

The Suggest Edit function can be a tool for collecting changes and suggestions from many different sources for your content. No longer will content suggestions arrive in the form of a cryptic e-mail. You can view a summary of exactly what the changes are and where they go. When someone submits a suggested edit for one of your modules, you are notified the next time you log into Connexions. The suggested edits will appear in your My Account sidebar. You can view the changes from this link. Since these are suggested edits, you do have the option to use them or reject them.

[37] This content is available online at <http://cnx.org/content/m11817/1.9/>.
[38] http://cnx.rice.edu

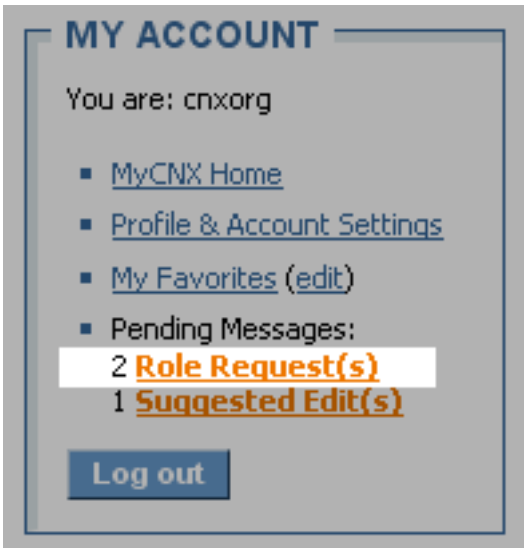


Figure 2.44: My Account sidebar with Suggested Edits highlighted

2.6.2.1 Reviewing Suggested Edits

To review the suggested edits for your content, use the following steps:

1. Click the **Suggested Edit** link in the "Pending" sidebar. The "Suggested Edits" screen displays. This screen identifies all the current suggested edits for your modules and any suggested edits that you have submitted. For your modules, this screen lists who submitted the edit, the date and time, the submitter's message, and the status of the submission.
2. Click **View** in the "Edit" column to review the suggested edit. A detail (Figure 2.45) of the edits displays on the screen. This display contains the same information as the previous screen, plus the details of the changes to the files, metadata, roles, and links.

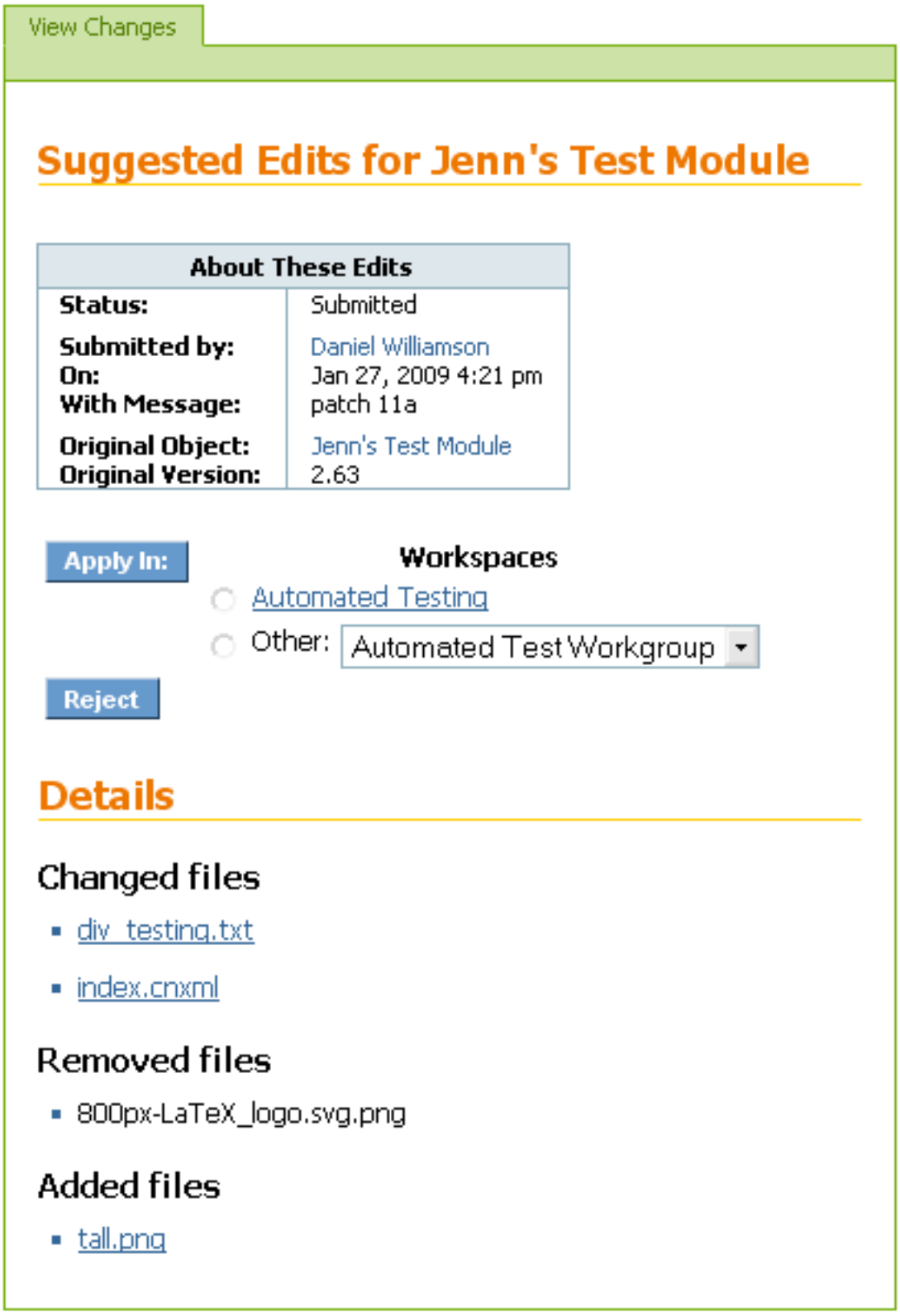


Figure 2.45: Details of a suggested edit to a module

The details identify any files in the module that were changed, removed, or added.

3. Click the filename link to view the contents of the file. Any deleted lines are marked with a "-" on the left. Any new lines are marked with a "+" on the left. The lines of text that were changed are shown twice. The first line is marked with a "-" on the left and contains the original text. The second line is marked with a "+" on the left and contains the changes.
4. Decide if you want to use the suggested edits or to ignore them. If you want to use the edits, go to Applying Suggested Edits (Section 2.6.2.2: Applying Suggested Edits). If you want to ignore the edits, go to Rejecting Suggested Edits. (Section 2.6.2.3: Rejecting Suggested Edits)

2.6.2.2 Applying Suggested Edits

To apply the suggested edits to your content, use the following steps:

1. Select the radio button for the workspace in which you want the changes applied. The Workspaces table lists the locations where the module already exists. You can select "Other" and specify a different workspace and Connexions will check out the module to that workspace.
2. Click `Apply In`. The "Accept Suggested Edits" screen displays.
3. Type a message about accepting the edits in the "Comments" text box and click `Update`. The submitter will see this message the next time he or she displays the "Suggested Edits" screen. The module edit screen displays a message that the suggested edits were applied.
4. Publish the module.

NOTE: You must publish the module to make the changes visible to all.

2.6.2.3 Rejecting Suggested Edits

To reject the suggested edits for your content, use the following steps:

1. Click `Reject`. The "Reject Suggested Edits" screen displays.
2. Type a message about rejecting the edits in the "Comments" text box and click `Update`. The submitter will see this message the next time he or she displays the "Suggested Edits" screen. The module edit screen displays a message that the suggested edits were rejected.

Chapter 3

XML and CNXML

3.1 Editing CNXML with Edit-In-Place

3.1.1 Using Basic CNXML in Edit-In-Place[1]

PLEASE NOTE: This module also contains information derived from The Advanced CNXML (Section 3.3.3) by Ricardo Radaelli-Sanchez[2] .

3.1.1.1 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!

Connexions[3] uses the **Connexions Markup Language** (**CNXML**) as its primary language for marking up and storing documents. CNXML is lightweight XML[4] 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 you 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 3.1.1.2.1: Emphasis) **tag**.

3.1.1.2 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.

3.1.1.2.1 Emphasis

As mentioned previously (p. 91), the `emphasis` tag is used to accent certain text. Note that this refers to **semantic** emphasis and not a typeface. Different **stylesheets** can render emphasis with different typefaces.

Example 3.1

[1] This content is available online at <http://cnx.org/content/m14394/1.4/>.
[2] http://cnx.org/member_profile/rars
[3] http://cnx.org
[4] http://en.wikipedia.org/wiki/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.

3.1.1.2.2 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 3.1.2.1: Para) or definition (Section 3.3.2.6: Definition) tag. The `term` tag has one optional **attribute**: `URL` - a URL specifying the source or definition of the term.

Example 3.2

```
<para id='gardenia'>
  <term url="http://en.wikipedia.org/wiki/Gardenia">Gardenias
  </term> can be tricky to maintain.  The soil around the
  roots of a <term>gardenia</term> must remain moist always,
  but too much water can damage the plant.  Also,
  <term>gardenias</term> enjoy the sun, but if the
  <emphasis>foliage</emphasis> gets wet to bring the gardenia
  into the shade.
</para>
```

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.

3.1.1.2.3 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 3.3

```
<para id='pollen'>
  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.
</para>
```

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 counter-top and clothing stains!

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

3.1.1.2.4 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.

3.1.1.2.5 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 3.4: Cite Example

```
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*.

3.1.1.2.6 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 3.5: Quote Example

```
<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."

3.1.1.2.7 Foreign

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

Example 3.6: Foreign Example

```
<para id='plantquote2'>
  All flowers have a scientific name, often derived from
  Latin.  <foreign>Gardenia augusta</foreign> is the name
  of a type of gardenia found in Japan.
</para>
```

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

3.1.1.2.8 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 3.7: 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 3.1.2.6: Code).

3.1.2 Advanced CNXML using Edit-in-Place[5]

3.1.2.1 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 3.1: A Paragraph Box) will appear. You can now insert text in the white box, including inline tags (Section 3.1.1.2: Inline Tags). Note the `id="element-143"` in the upper left hand part of the blue box in Figure 3.1 (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 3.1.1.2.4: 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

Figure 3.1: When you click "insert and choose paragraph", a box like this should appear.

Example 3.8: Submitted by J. Cameron Cooper

[5] This content is available online at <http://cnx.org/content/m14395/1.4/>.

```
<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
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.
</para>
<para id="intro3">
  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.
</para>
<para id="intro4">
  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!
</para>
<para id="intro5">
  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.
</para>
```

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!

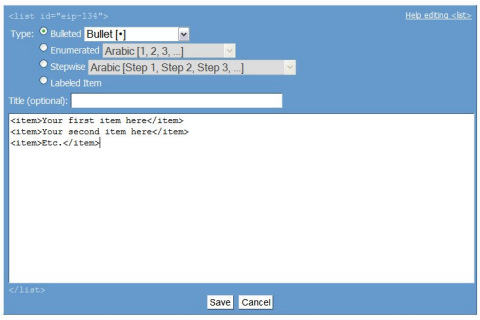
If you have a lot of leaves or similar small stuff to move, you can use a similar technique. Get a tarp, toss the leaves and weeds and whatnot in the middle, and then drag the whole thing away.

3.1.2.2 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 3.2: 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

(a) Enumerated List



(b) Bulleted List

Figure 3.2: (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.

Example 3.9: 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.
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 3.10: Bulleted List

```
<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

3.1.2.3 Equation

The `equation` tag is used to set off and number equations in CNXML documents. If you have MathML (Section 4.1) 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 4.1) tags when displaying math.

If you look at Figure 3.3 (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.

Adding an Equation

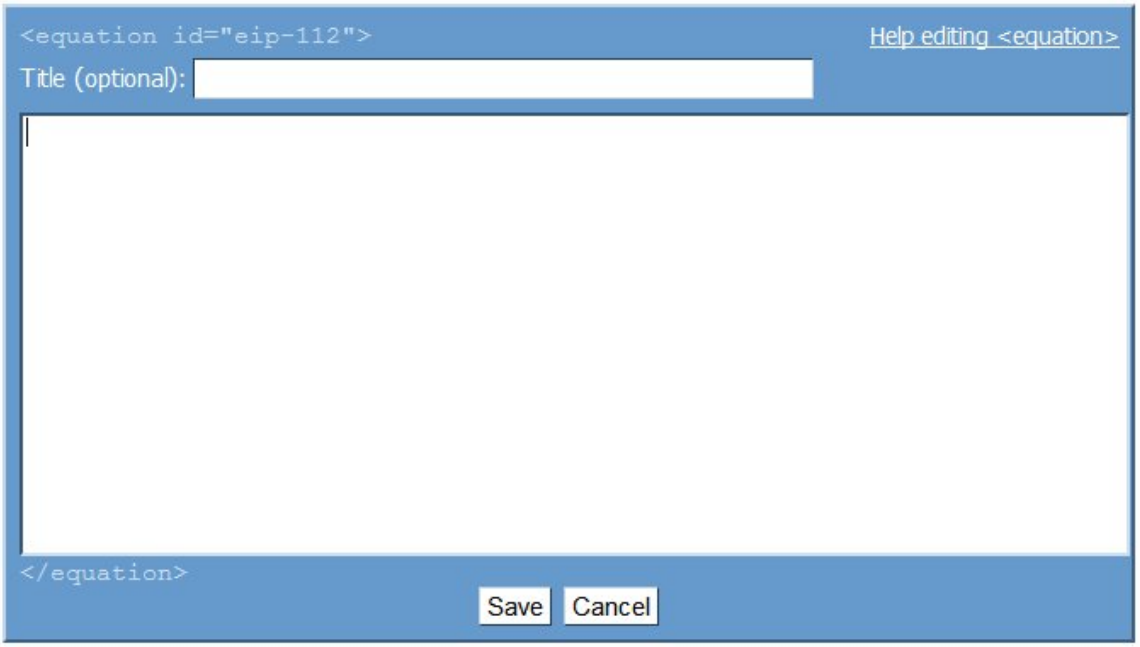


Figure 3.3

As with lists, you can add an optional `title` at the beginning of each equation.

Example 3.11: Using Images as Equations

```
  <equation id="eqn14">
 <media id="img12" display="block" alt="1+2=3"
<image mime-type='image/gif' src='euler.gif' />
</equation>
```

displays as:

$$1+2=3 \tag{3.1}$$

Example 3.12: ASCII equations

```
    <equation id='eqn15'>
  <title>Simple Arithmetic</title>
    11+27=38
</equation>
```

This equation will display as:

Simple Arithmetic

$$11+27=38 \tag{3.2}$$

3.1.2.4 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 3.4: 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

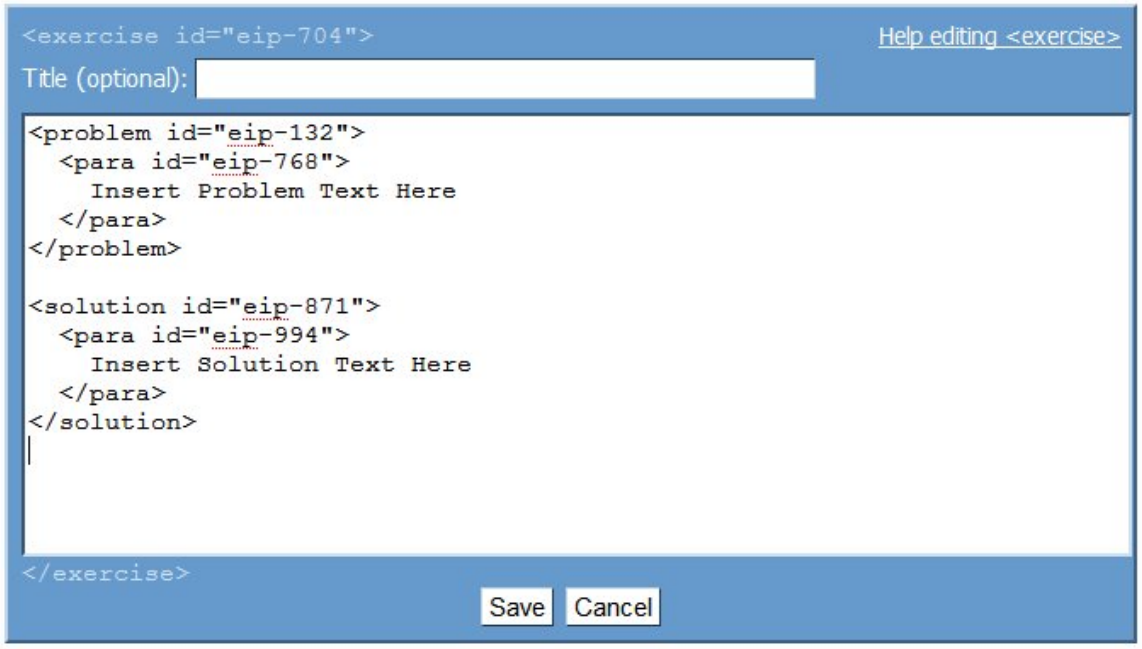


Figure 3.4

To continue utilizing edit-in-place to edit your exercise, press the `Save` button (see Figure 3.5 (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

Figure 3.5: 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[6].

Example 3.13

```
    <exercise id='hyd_test'>
  <problem id="id9">
    <para id='hyd_testp1'>
      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?
    </para>
  </problem>
  <solution id="id10">
    <para id='hyd_sol1p1'>
      Highly acidic soil produces blue flowers.  Highly
      basic soil produces pink flowers. Neutral soil produces
      very pale cream flowers.
    </para>
  </solution>
</exercise>
```

[6] "QML 1.0" <http://cnx.org/content/m10136/latest/>

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.

3.1.2.5 Figure

The `figure` tag provides the structure for creating a figure within a document. They can contain either two or more subfigure (Section 3.3.2.3: Subfigure) tags, or a single media (Section 2.4), table (Section 3.1.3), or code (Section 3.1.2.6: Code) tag.

Adding a Figure

Figure 3.6: 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 3.3.1.2.2: 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 2.4).

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

Example 3.14: Example of a Figure

```
<figure id='blossom'>
   <title>Momosa Blossom</title>
   <media id="image-example" display="block" alt="A Momosa Blossom.">
      <image id="flower" mime-type="image/jpeg" src="alb_jul_flo_1.jpg">
   </media>
   <caption>
      Picture taken by Jenn Drummond (CC Attribution).
   </caption>
</figure>
```

This code will display as:

Momosa Blossom

Figure 3.7: Picture taken by Jenn Drummond (CC Attribution).

3.1.2.6 Code

As seen in Using Basic CNXML in Edit-in-Place (Section 3.1.1.2.8: Code), you can add inline code to your document; edit-in-place also allows you in insert a block of code (Figure 3.8: Adding a Block of Code), separate from text.

Adding a Block of Code

Figure 3.8: 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 (> and <, 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 3.15 (A Block of Code, Using CDATA).

Example 3.15: A Block of Code, Using CDATA

Using CDATA in a Code Block

Figure 3.9

When saved, Figure 3.9 (Using CDATA in a Code Block) will display as:

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

3.1.2.7 Note

As mentioned in Using Basic CNXML in Edit-in-Place (Section 3.1.1.2.3: 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.

Adding a Note using Edit-in-Place

Figure 3.10: As with `code`, notes require a unique ID when the display attribute is "block".

Example 3.16

```
<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!

3.1.2.8 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 3.11: 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

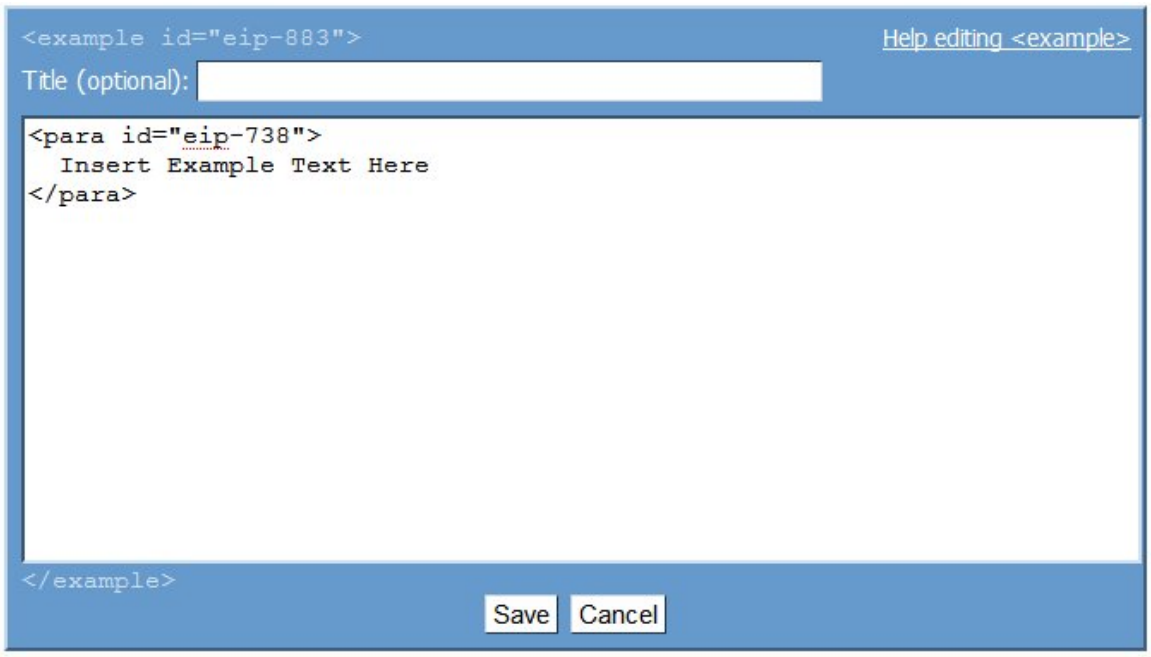


Figure 3.11

Example 3.17
Here is the code for Example 3.16:

```
<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>
  <para id="notep2">
    The above markup will display as:
  </para>
  <note>
    Gardening requires a lot of intense physical exertion.
    Please drink plenty of water to avoid dehydration!
  </note>
</example>
```

3.1.2.9 CALS Table

The final element you can add using Edit-in-Place is `table`. To learn more about adding and editting tables using Edit-in-Place, see CALS Table (Section 3.1.3). For a more complete description of the CALS Table consult the CALS Table Spec[7] .

3.1.3 CALS Table[8]

CNXML uses the industry standard CALS Table Model[9] 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 3.12 (Adding a Table using Edit-in-Place).

[7] http://www.oasis-open.org/specs/a502.htm
[8] This content is available online at <http://cnx.org/content/m14396/1.5/>.
[9] http://www.oasis-open.org/specs/a502.htm

Adding a Table using Edit-in-Place

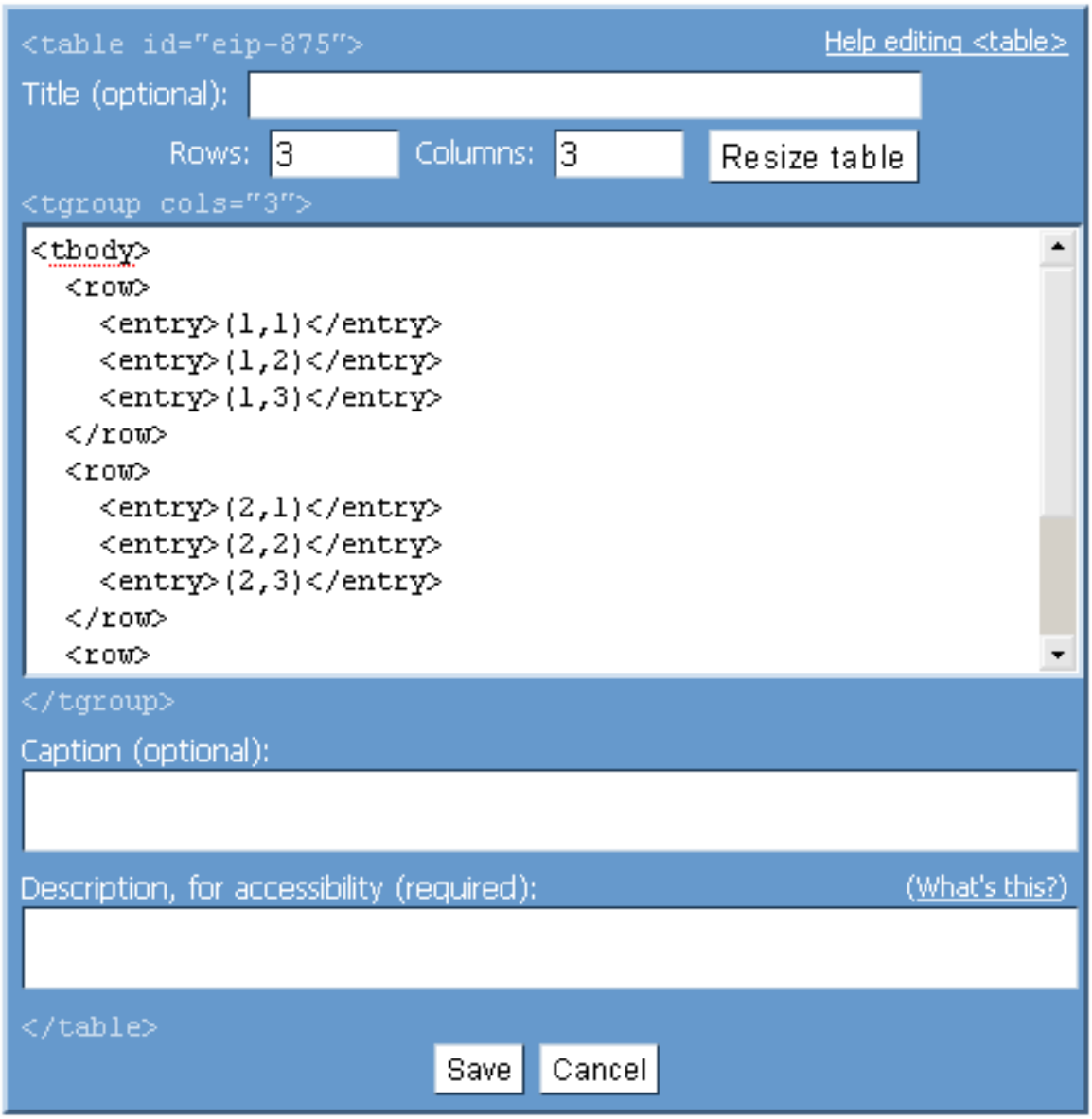


Figure 3.12

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

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.

Once your table is correctly sized, you can start editing the entries of your table, as shown in Figure 3.13 (Editing Entries of a CALS Table).

Editing Entries of a CALS Table

Figure 3.13

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[10] .

3.1.3.1 table

The `table` tag marks the beginning of a table. It has an optional first child of name (Section 3.3.1.2.2: Title) (as you can see in the Edit-in-Place interface) and must contain one or more tgroup (p. 112) 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[11] .

3.1.3.2 tgroup

The `tgroup` tag marks the beginning of a new portion of a table (p. 112). It has a required attribute `cols` which is the number of columns in the `tgroup`; as mentioned previously (p. 110), this attribute can be set using the Edit-in-Place interface. Its children tags are zero, one, or more colspec (p. 112) or spanspec (p. 112), zero or one thead (p. 112) or tfoot (p. 112), and one tbody (p. 112) tag.

3.1.3.3 colspec

The `colspec` tag is an **empty tag** that specificies the column of a table (p. 112) or entrytbl (p. 112). The names and numbers specified as attributes are used for referencing by other tags.

3.1.3.4 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. 113) or entrytbl (p. 112) in different rows (p. 112).

3.1.3.5 thead

The `thead` tag identifies the heading row (p. 112) of a tgroup (p. 112) or entrytbl (p. 112). The `thead` tag can have zero, one, or more colspec (p. 112) tags and one or more row (p. 112).

3.1.3.6 tfoot

The `tfoot` tag identifies the rows (p. 112) of footer information that are displayed after the tbody (p. 112). The `tfoot` tag can have zero, one, or more colspec (p. 112) tags and one or more row (p. 112).

3.1.3.7 tbody

The `tbody` tag identifies the body of a tgroup (p. 112) or entrytbl (p. 112). The `tbody` tag must have one or more row (p. 112) tags.

3.1.3.8 row

The `row` tag identifies the row of information in a thead (p. 112), tbody (p. 112), or tfoot (p. 112). The `row` tag must have one or more entry (p. 113) or entrytbl (p. 112).

3.1.3.9 entrytbl

The `entrytbl` tag takes the place of an entry (p. 113), but fits into a single row (p. 112) of tbody (p. 112) in a tgroup (p. 112). The content model is the same as that of a tgroup (p. 112) except that tfoot (p. 112) is ommitted and `entrytbl` is self-excluding. Its children tags are zero, one, or more colspec (p. 112) or spanspec (p. 112), zero or one thead (p. 112) or tfoot (p. 112), and one tbody (p. 112) tag.

[10] http://www.oasis-open.org/specs/a502.htm
[11] http://www.oasis-open.org/specs/a502.htm

3.1.3.10 entry

The `entry` tag identifies an entry in a row (p. 112). The `entry` tag contains ASCII text and zero, one, or many cite (Section 3.1.1.2.5: Cite), term (Section 3.1.1.2.2: Term), cnxn (Section 3.1.1.2.4: Link), link (Section 3.1.1.2.4: Link), code (Section 3.1.1.2.8: Code), emphasis (Section 3.1.1.2.1: Emphasis), or media (Section 2.4).

3.1.3.11 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 3.18. For more information, consult the CALS Table Spec[12] .

Example 3.18

[12] http://www.oasis-open.org/specs/a502.htm

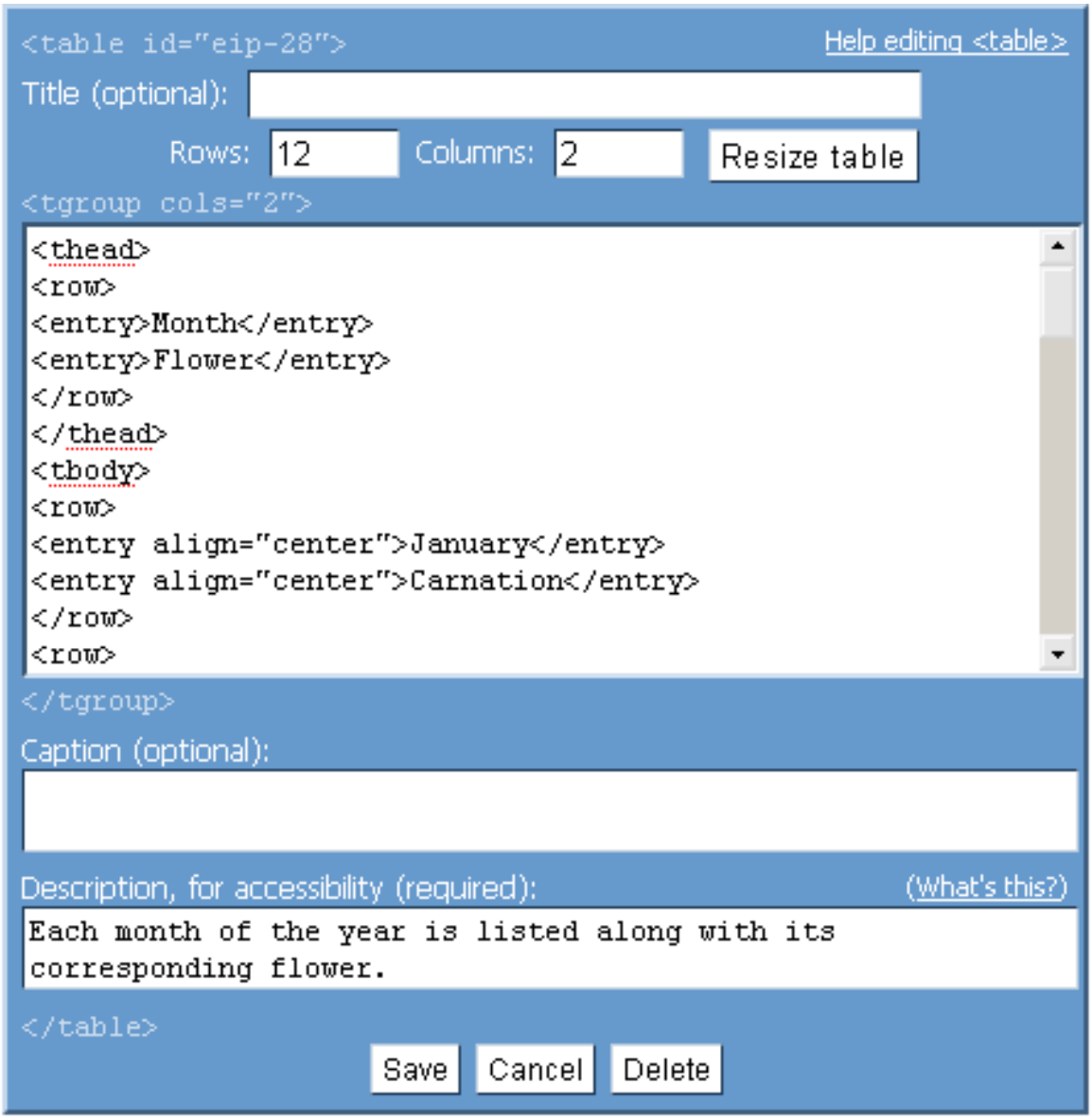


Figure 3.14

```
    <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>
<row>
<entry>June</entry>
<entry>Honeysuckle</entry>
</row>
<row>
<entry>July</entry>
<entry>Larkspur</entry>
</row>
<row>
<entry>August</entry>
<entry>Lily</entry>
</row>
<row>
<entry>September</entry>
<entry>Forget-Me-Not</entry>
</row>
<row>
<entry>October</entry>
<entry>Marigold</entry>
</row>
<row>
<entry>November</entry>
<entry>Chrysanthemum</entry>
</row>
<row>
<entry>December</entry>
<entry>Narcissus</entry>
</row>
</tbody>
```

This code will end up looking like this (Table 3.1: Birth Flowers):

Birth Flowers

Month	Flower
January	Carnation
February	Violet
March	Daffodil
April	Sweet Pea
May	Sunflower
June	Honeysuckle
July	Larkspur
August	Lily
September	Forget-Me-Not
October	Marigold
November	Chrysanthemum
December	Narcissus

Table 3.1

3.2 Editing XML with the Full Source Editor

3.2.1 XML Basics[13]

3.2.1.1 What is XML?

The eXtensible Markup Language (**XML**) is a **meta-markup language** defined by the World Wide Web Consortium (W3C)[14] . 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[15] , the markup language used by the Connexions Project[16] 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 3.19

```
<para>
  This is a paragraph in <term>CNXML</term>.  Notice that the markup
  contains tags that express the meaning of the text.
</para>
```

[13] This content is available online at <http://cnx.org/content/m9002/2.24/>.
[14] http://www.w3.org
[15] http://cnx.rice.edu/cnxml
[16] http://cnx.rice.edu

<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.

XML allows the separation of presentation from content. For example, HTML has tags such as <u> and , 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.

3.2.1.2 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, red and blue is fine, but red and blueis incorrect because the and 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.

3.2.1.3 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.

3.2.1.4 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 3.2.1.5: 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 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:

- `&` refers to an ampersand (&)
- `<` refers to a less-than symbol (<)
- `>` refers to a greater-than symbol (>)
- `&quot;` refers to a double-quote mark (")
- `&apos;` refers to an apostrophe (')

Example 3.20
Suppose you have a document with the following:

```
<para id="p1">The firm was known as Scrooge and Marley.</para>
```

you could replace 'and' with the entity reference `&`:

```
<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 `;`.

3.2.1.5 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 hexidecimal representation of a Unicode code point.

Example 3.21
The hexidecimal 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 `&#x00F8;` and `&#248;` So you could write

```
<emphasis>The majestik m&#x00F8;&#x00F8;se</emphasis>
```

or

```
<emphasis>The majestik m&#248;&#248;se</emphasis>
```

or even

```
<emphasis>The majestik m&#x00F8;&#248;se</emphasis>
```

to get

The majestik møøse

3.2.2 Combining XML Languages[17]

XML[18] 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 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[19]). 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 3.22
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:

```
<document xmlns="http://cnx.rice.edu/cnxml"
          xmlns:md="http://cnx.rice.edu/mdml/0.4"
          xmlns:bib="http://bibtexml.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">
```

[17] This content is available online at <http://cnx.org/content/m10159/2.16/>.
[18] http://www.w3.org/XML
[19] http://www.w3.org/TR/REC-xml-names/

3.3 Editing CNXML with the Full Source Editor

3.3.1 The Basic CNXML[20]

3.3.1.1 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[21] uses the Connexions Markup Language (CNXML) as its primary language for storing documents. Now let's get started!

3.3.1.2 CNXML Tags

3.3.1.2.1 Document

All CNXML documents have as their root the `document` tag. Everything about the document including it's 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.

3.3.1.2.1.1 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
- table
- div
- section
- subfigure
- example
- footnote
- problem
- solution
- block quotes
- media
- meaning

[20] This content is available online at <http://cnx.org/content/m9000/2.36/>.
[21] http://cnx.org

- proof
- list
- preformat
- block code
- figure
- block notes

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 you own ids if you wish.

Example 3.23
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.

3.3.1.2.1.2 Namespaces

The `document` tag should also contain any **namespace** declarations. Namespaces (Section 3.2.2) 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 namspace `xmlns:bib="http://bibtexml.sf.net/"`, and the question markup language `xmlns:q="http://cnx.rice.edu/qml/1.0"`.

Example 3.24
This what the document tag should look like.

```
<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.

3.3.1.2.2 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 3.25

```
<title>Grilling a Good Steak</title>
```

NOTE: Please see the CNXML tag list in Edit-In-Place[22] to see if a tag can be named.

3.3.1.2.3 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.

3.3.1.2.3.1 Structure Tags

Some of the structure tags are section (Section 3.3.1.2.3.1.2: Section), para (Section 3.3.1.2.3.1.1: Para), document (p. 120), and content (Section 3.3.1.2.3: Content).

We have already discussed the document (p. 120) and content (Section 3.3.1.2.3: Content) tags, so we will proceed with a short description and examples of the other other tags listed.

3.3.1.2.3.1.1 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 3.26

```
<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
```

[22] http://cnx.org/eip-help/tags

```
  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>
```

3.3.1.2.3.1.2 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 3.3.1.2.2: Title).

Example 3.27

```
<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.

3.3.1.2.3.2 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.

3.3.1.2.3.2.1 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.

Example 3.28

```
<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 <emphasis>perfectly</emphasis> seasoned to
  your taste.  Either way, this document shows how a good steak can be
  prepared.
</para>
```

3.3.1.2.3.2.2 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 3.3.1.2.3.1.1: 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).

Example 3.29

```
<para id='marinade'>
  To ensure the best flavor possible, it is necessary to marinate the
  beef.  A steak <term url='http://marinade.com'>marinates</term> when
  left to sit in a prepared sauce, or <term>marinade</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
  <emphasis>always</emphasis> be done in the refrigerator and
  <emphasis>not</emphasis> at room temperature.
</para>
```

3.3.1.2.3.2.3 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 3.30

```
<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>
```

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.

3.3.1.2.3.2.4 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.

3.3.1.2.3.2.5 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).

3.3.1.2.3.2.6 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. 126).

Example 3.31

```
  <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."

3.3.1.2.3.2.7 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. 126).

Example 3.32

```
  <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*.

3.3.1.3 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>.
Finally, a good resource is the <cite>Steak Lover's Cookbook --
William Rice</cite>.
      </para>
    </section>

  </content>
```

```
</document>
```

See how Connexions[23] would render this example (Section 5.1).

3.3.1.4 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: `<?xml version="1.0" encoding="utf-8"?>` , and must not be preceeded by any blank lines or whitespace. CNXML 0.6 only uses one schema, so there is no need to specificy specific DTDs. Below is an example of a correct CNXML 0.6 document tag containing the proper namespaces.

Example 3.33

```
<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">
```

3.3.1.5 Conclusions

Remember that when composing documents it is always best to consult the CNXML Tag List[24] for any questions regarding the exact usage of CNXML tags. For more advanced topics see The Intermediate CNXML (Section 3.3.2) or The Advanced CNXML (Section 3.3.3), which concludes the cooking lesson.

3.3.2 The Intermediate CNXML[25]

3.3.2.1 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 `id` attribute and can contain most tags as children, the first being an optional title (Section 3.3.1.2.2: Title). For specifics you should always consult the CNXML Spec[26] .

Example 3.34

```
<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'/>
```

[23] http://cnx.org
[24] http://cnx.org/eip-help/tags
[25] This content is available online at <http://cnx.org/content/m9006/2.22/>.
[26] http://cnx.rice.edu/technology/cnxml/0.5/spec

```
    </media>
  </figure>
</example>
```

3.3.2.2 Figure

The `figure` tag provides the structure for creating a figure within a document. They can contain either two or more subfigure (Section 3.3.2.3: Subfigure) tags, or a single media (p. 129), table (p. 137), or code (Section 3.3.3.1: Code) tag.

The `figure` tag has two attributes:

- `id` - a unique ID, required
- `orient` - defines how multiple subfigure (Section 3.3.2.3: Subfigure)s are to be displayed. It takes two values, `vertical` or `horizontal`, and will default to `horizontal`.

The optional first tag of the `figure` tag is title (Section 3.3.1.2.2: 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. 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[27] 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 3.35

```
<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>
```

[27] http://www.ietf.org/rfc/rfc1341.txt?number=1341

3.3.2.3 Subfigure

The `subfigure` tag is used when you want to include more than one media (p. 129), code (Section 3.3.3.1: Code) or table (p. 137) within the same figure (Section 3.3.2.2: Figure).

The usage of the `subfigure` tag is similar to that of figure (Section 3.3.2.2: Figure). It has an optional `id` attribute, an optional first child title (Section 3.3.1.2.2: Title) tag, a single media (p. 129), code (Section 3.3.3.1: Code) or table (p. 137), followed by an optional caption (p. 129).

Now the `orient` attribute for figure (Section 3.3.2.2: 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 3.36

```
<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='subfingnystrip1'>
    <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

```
<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 mime-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.jpg'/>
    </media>
  </subfigure>
  <caption>
```

```
    Upon successful completion of these documents, you
    should be able to grill a steak that looks just as good!
  </caption>
</figure>
```

3.3.2.4 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. 129), which is optional, and `item`, which is where the list information is stored.

Example 3.37: Example List

```
<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

Example 3.38: 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:

```
<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

3.3.2.5 Equation

The `equation` tag is used to set off and number equations in CNXML documents by using ASCII text, MathML and embedded media (p. 129) to display math.

NOTE: It is strongly encouraged, however, to use equation with MathML (Section 4.1) tags when displaying math.

3.3.2.5.1 ASCII Text and Images

The first child of equation is an optional title (Section 3.3.1.2.2: Title) followed by any number of media (p. 129) tags.

Example 3.39

```
<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

$$11+27=38 \tag{3.3}$$

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>
```

3.3.2.6 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 3.3.1.2.3.2.2: Term), `meaning` and example (Section 3.3.2.1: Example). How to use definition is a little confusing, so don't forget to check out Example 3.40.

The first child tag should be term (Section 3.3.1.2.3.2.2: 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 3.3.2.1: Example)s. This process repeats for all meanings.

Example 3.40

```
<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>
      </media>
    </figure>
  </example>
</definition>
```

Definition 3.1: 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

T-Bone Steak

Figure 3.15

3.3.2.7 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:

- `id` - required, unique ID
- `type` - required, specificies the type of rule (e.g. theorem, axiom, rule of thumb, etc.)

It may also have an optional title (Section 3.3.1.2.2: Title) and it must have one or more statement (Section 3.3.2.7.1: Statement) tags and zero or more proof (Section 3.3.2.7.2: Proof) or example (Section 3.3.2.1: Example) tags.

3.3.2.7.1 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 3.3.1.2.3.1.1: Para) and equation (Section 3.3.2.5: Equation).

3.3.2.7.2 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 3.3.1.2.3.1.1: Para), equation (Section 3.3.2.5: Equation), figure (Section 3.3.2.2: Figure), or list tag.

3.3.2.7.3 Using rule

Example 3.41

```
<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 3.1: 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.lylemariam.com/murphy.htm*.

3.3.2.8 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 3.3.3).

3.3.3 The Advanced CNXML[28]

3.3.3.1 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[29] . 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 3.42

```
<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`

3.3.3.2 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 used in place of the problem and solution tags. For more information, please see the information about QML[30] .

Example 3.43

```
<exercise id='grilltest'>
  <problem>
    <para id='grilltestp1'>
      For food safety, a steak should be cooked to a minimum
```

[28] This content is available online at <http://cnx.org/content/m9007/2.24/>.
[29] http://cnx.rice.edu/technology/cnxml/0.5/spec
[30] http://cnx.rice.edu/qml/intro/qml.xml

```
      temperature of what?
    </para>
  </problem>
  <solution>
    <para id='sol1p1'>
      160&deg; 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

3.3.3.3 CALS Table

CNXML uses the industry standard CALS Table Model[31] 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 3.2: Steak Cooking Temperatures)). For a more complete description of the CALS Table consult the CALS Table Spec[32] .

3.3.3.3.1 table

The `table` tag marks the beginning of a table. It has an optional first child of title (Section 3.3.1.2.2: Title) and must contain one or more tgroup (p. 137) tags. The `table` tag also has many attributes, to find out more information consult the CALS Table Spec[33] .

3.3.3.3.2 tgroup

The `tgroup` tag marks the beginning of a new portion of a table (p. 137). 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. 137) or spanspec (p. 137), zero or one thead (p. 138) or tfoot (p. 138), and one tbody (p. 138) tag.

3.3.3.3.3 colspec

The `colspec` tag is an **empty tag** that specificies the column of a table (p. 137) or entrytbl (p. 138). The names and numbers specified as attributes are used for referencing by other tags.

3.3.3.3.4 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. 138) or entrytbl (p. 138) in different rows (p. 138).

[31] http://www.oasis-open.org/specs/a502.htm
[32] http://www.oasis-open.org/specs/a502.htm
[33] http://www.oasis-open.org/specs/a502.htm

3.3.3.3.5 thead

The `thead` tag identifies the heading row (p. 138) of a tgroup (p. 137) or entrytbl (p. 138). The `thead` tag can have zero, one, or more colspec (p. 137) tags and one or more row (p. 138).

3.3.3.3.6 tfoot

The `tfoot` tag identifies the rows (p. 138) of footer information that are displayed after the tbody (p. 138). The `tfoot` tag can have zero, one, or more colspec (p. 137) tags and one or more row (p. 138).

3.3.3.3.7 tbody

The `tbody` tag identifies the body of a tgroup (p. 137) or entrytbl (p. 138). The `tbody` tag must have one or more row (p. 138) tags.

3.3.3.3.8 row

The `row` tag identifies the row of information in a thead (p. 138), tbody (p. 138), or tfoot (p. 138). The `row` tag must have one or more entry (p. 138) or entrytbl (p. 138).

3.3.3.3.9 entrytbl

The `entrytbl` tag takes the place of an entry (p. 138), but fits into a single row (p. 138) of tbody (p. 138) in a tgroup (p. 137). The content model is the same as that of a tgroup (p. 137) except that tfoot (p. 138) is ommitted and `entrytbl` is self-excluding. Its children tags are zero, one, or more colspec (p. 137) or spanspec (p. 137), zero or one thead (p. 138) or tfoot (p. 138), and one tbody (p. 138) tag.

3.3.3.3.10 entry

The `entry` tag identifies an entry in a row (p. 138). The `entry` tag contains ASCII text and zero, one, or many cite (Section 3.3.1.2.3.2.5: Cite), term (Section 3.3.1.2.3.2.2: Term), cnxn, link, code, emphasis (Section 3.3.1.2.3.2.1: Emphasis), or media (p. 129).

3.3.3.3.11 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 3.44. For more information, consult the CALS Table Spec[34] or the CNXML Spec[35] .

Example 3.44

```
<table id='grilltemp' frame='all'>
  <title>Steak Cooking Temperatures</title>
  <tgroup cols='2' colsep='1' rowsep='1'>
    <thead>
      <row>
        <entry>Temperature (&deg;F)</entry>
        <entry>Description</entry>
      </row>
    </thead>
    <tbody>
```

[34] http://www.oasis-open.org/specs/a502.htm
[35] http://cnx.rice.edu/technology/cnxml/0.5/spec

```
      <row>
        <entry align='center'>140</entry>
        <entry align='center'>Rare</entry>
      </row>
      <row>
        <entry align='center'>150</entry>
        <entry align='center'>Medium Rare</entry>
      </row>
      <row>
        <entry align='center'>160</entry>
        <entry align='center'>Medium</entry>
      </row>
      <row>
        <entry align='center'>165</entry>
        <entry align='center'>Medium Well</entry>
      </row>
      <row>
        <entry align='center'>170</entry>
        <entry align='center'>Well</entry>
      </row>
    </tbody>
  </tgroup>
</table>
```

Steak Cooking Temperatures

Temperature (°F)	Description
140	Rare
150	Medium Rare
160	Medium
165	Medium Well
170	Well

Table 3.2

3.3.3.4 Conclusions

This concludes the CNXML tutorial.

3.3.4 CNXML Reference Extensions[36]

3.3.4.1 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

[36] This content is available online at <http://cnx.org/content/m11215/1.6/>.

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.

3.3.4.1.1 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)*

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.

3.3.4.1.2 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 3.46 (Linking to Definitions in a Glossary)).

3.3.4.2 Including a Glossary

It is very easy to include a glossary in your CNXML document. In the Basic CNXML Tutorial (Section 3.3.1) 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 specification[37] .

Example 3.45: Glossary Example
Following is an example of the code necessary to add a glossary with one definition.

```
<glossary>
  <definition id='quardef'>
    <term>quarter</term>
    <meaning><name>Meaning Name</name>One fourth of something.</meaning>
    <example id='def'>
      <para id='par'>
```

[37] http://cnx.rice.edu/technology/cnxml/0.5/spec

```
"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 3.46: 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 `src` attribute in the term tag, one can link to a definition. Simply set the value of the `src` 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 3.45: Glossary Example):

```
<term src='#quardef'>quarter</term>
```

3.3.4.3 Including BibteXML

It is very easy to include a bibteXML reference section in your CNXML document. In the Basic CNXML Tutorial[38] it is stated that the structure usually resembles the following:

Document

- `name`
- `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`

[38] "Themes" <http://cnx.org/content/m0000/latest/>

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 3.47: BibteXML Example

```
<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;amp; Sons Ltd.</bib:publisher>
      <bib:year>1989</bib:year>
      <bib:address>Chichester</bib:address>
    </bib:book>
  </bib:entry>
</bib:file>
```

NOTE: Each tag in the example code begins with a namespace prefix. In the case of bibteXML, the prefix is `bib:`.

Example 3.48: 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 `src` attribute in the cite tag, one can link to a bibliographic entry. Simply set the value of the `src` 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 3.47 (BibteXML Example):

```
<cite src='#esbensen'>Multivariate Analysis in Practice</cite>
```

3.3.4.3.1 BibteXML Tags

By looking at Example 3.47 (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 Homepage[39] .

[39] http://bibtexml.sourceforge.net/

3.3.4.3.1.1 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 3.3.4.3.1.2: Entry)` tags.

3.3.4.3.1.2 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

3.3.4.3.1.3 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 3.3.4.3.1.4: BibteXML Metadata Tags)`. For more information on exactly which children a specific container may possess, please see the BibteXMLHomepage[40] .

3.3.4.3.1.4 BibteXML Metadata Tags

Every child listed as a possible child of `entry (Section 3.3.4.3.1.2: Entry)` can contain metadata tags. These metadata tags are now listed.

Metadata Tag List

- address
- author
- booktitle
- chapter
- edition
- editor
- howpublished
- institution
- journal

[40] http://bibtexml.sourceforge.net/

- month
- note
- number
- organization
- pages
- publisher
- school
- series
- title
- type
- volume
- year

Children
Every metadata tag can contain unicode text.

Chapter 4

MathML Editor

4.1 Content MathML[1]

The authoritative reference for Content MathML is Section 4 of the MathML 2.0 Specification[2] . 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.

4.1.1 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.

[1] This content is available online at <http://cnx.org/content/m9008/2.15/>.
[2] http://www.w3.org/TR/MathML2/chapter4.html

4.1.1.1 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 the MathML 2.0 Specification[3] . 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.

```
<m:math>
  <m:ci>
    <m:msub>
      <m:mi>x</m:mi>
      <m:mn>2</m:mn>
    </m:msub>
  </m:ci>
</m:math>
```

This would display as x_2.

The `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 `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.

4.1.1.2 Functions and Operators

In order to apply a function to a variable, make the function the first argument of an 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 `ci` tag denoting f.)

```
<m:math>
  <m:apply>
    <m:ci type='fn'>f</m:ci>
    <m:ci>x</m:ci>
  </m:apply>
</m:math>
```

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 **empty tags** and they directly follow the apply tag. "The sine of x" is similar to the example above.

[3] http://www.w3.org/TR/MathML2/chapter3.html

```
<m:math>
  <m:apply>
    <m:sin/>
    <m:ci>x</m:ci>
  </m:apply>
</m:math>
```

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 nth-power of something), and root (for taking the nth-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."

```
<m:math>
  <m:apply>
    <m:minus/>
    <m:ci>x</m:ci>
  </m:apply>
</m:math>
```

This will display as $-x$.

In contrast to representing the negative of a variable, the negative of a number may be coded as follows:

```
<m:math><m:cn>-1</m:cn></m:math>
```

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.

```
<m:math>
  <m:apply>
    <m:times/>
    <m:ci>a</m:ci>
    <m:apply>
      <m:plus/>
      <m:ci>b</m:ci>
      <m:ci>c</m:ci>
    </m:apply>
```

```
    </m:apply>
  </m:math>
```

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.

```
<m:math>
  <m:apply>
    <m:eq/>
    <m:apply>
      <m:plus/>
      <m:apply>
<m:times/>
<m:ci>a</m:ci>
<m:ci>b</m:ci>
      </m:apply>
      <m:apply>
<m:times/>
<m:ci>a</m:ci>
<m:ci>c</m:ci>
      </m:apply>
    </m:apply>
    <m:apply>
      <m:times/>
      <m:ci>a</m:ci>
      <m:apply>
<m:plus/>
<m:ci>b</m:ci>
<m:ci>c</m:ci>
      </m:apply>
    </m:apply>
  </m:apply>
</m:math>
```

This will display as $ab + ac = a\,(b+c)$.

4.1.2 Integrals

The operator for an integral is `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."

```
<m:math>
```

```
        <m:apply>
    <m:int/>
    <m:bvar><m:ci>x</m:ci></m:bvar>
    <m:lowlimit><m:cn>0</m:cn></m:lowlimit>
    <m:uplimit><m:ci>b</m:ci></m:uplimit>
    <m:apply>
      <m:ci type='fn'>f</m:ci>
      <m:ci>x</m:ci>
    </m:apply>
  </m:apply>
</m:math>
```

This will display as $\int_0^b f(x)\,dx$.

4.1.3 Derivatives

The derivative operator is `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."

```
<m:math>
          <m:apply>
    <m:diff/>
    <m:bvar>
      <m:ci>x</m:ci>
    </m:bvar>
    <m:apply>
      <m:ci type="fn">f</m:ci>
      <m:ci>x</m:ci>
    </m:apply>
  </m:apply>
</m:math>
```

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."

```
<m:math>
          <m:apply>
            <m:diff/>
    <m:bvar>
      <m:ci>x</m:ci>
      <m:degree><m:cn>2</m:cn></m:degree>
    </m:bvar>
    <m:apply><m:ci type="fn">f</m:ci>
      <m:ci>x</m:ci>
    </m:apply>
  </m:apply>
</m:math>
```

This will display as $\frac{d^2}{dx^2} f(x)$.

4.1.4 Vector and Matrices

Vectors are created as a combination of other elements using the `vector` tag.

```
<m:math>
          <m:vector>
    <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.

4.1.5 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[4] from the XML Specification for more information. The MathML specification contains a list of character entities with their corresponding Unicode code points[5] .

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 charmap utility and GNOME applet provide access to all Unicode characters.

4.1.6 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[6] .

4.2 Introduction to the MathML Editor[7]

4.2.1 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 4.4) with examples showcasing the features.

At the end of this module are nuances and limitations (Section 4.2.15: Nuances / Limitations) of the editor. Please, let us know which ones you'd really like to see incorporated!

4.2.2 Opening the Editor

When editing a Module (Section 2.3.4.3: Edit the index.cnxml file with the Edit-In-Place Editor) using using Mozilla's Firefox browser, click on a part of the module to open a blue editing box (Figure 4.1). On the top-right hand side of the box is a "MathML Editor" link which will open up the editor (Figure 4.2).

[4] http://www.w3.org/TR/2003/REC-MathML2-20031021/chapter6.html#chars.unicodechars
[5] http://www.w3.org/TR/MathML2/bycodes.html
[6] http://www.w3.org/TR/MathML2/
[7] This content is available online at <http://cnx.org/content/m24561/1.1/>.

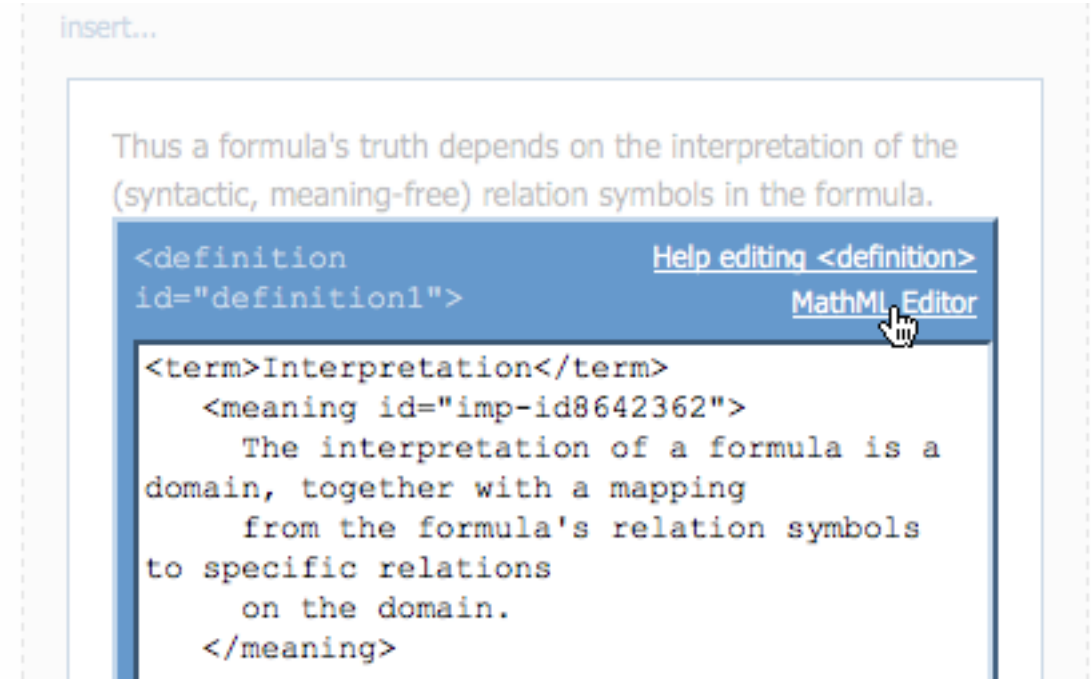


Figure 4.1: Begin editing and in the top-right corner is a MathML Editor link

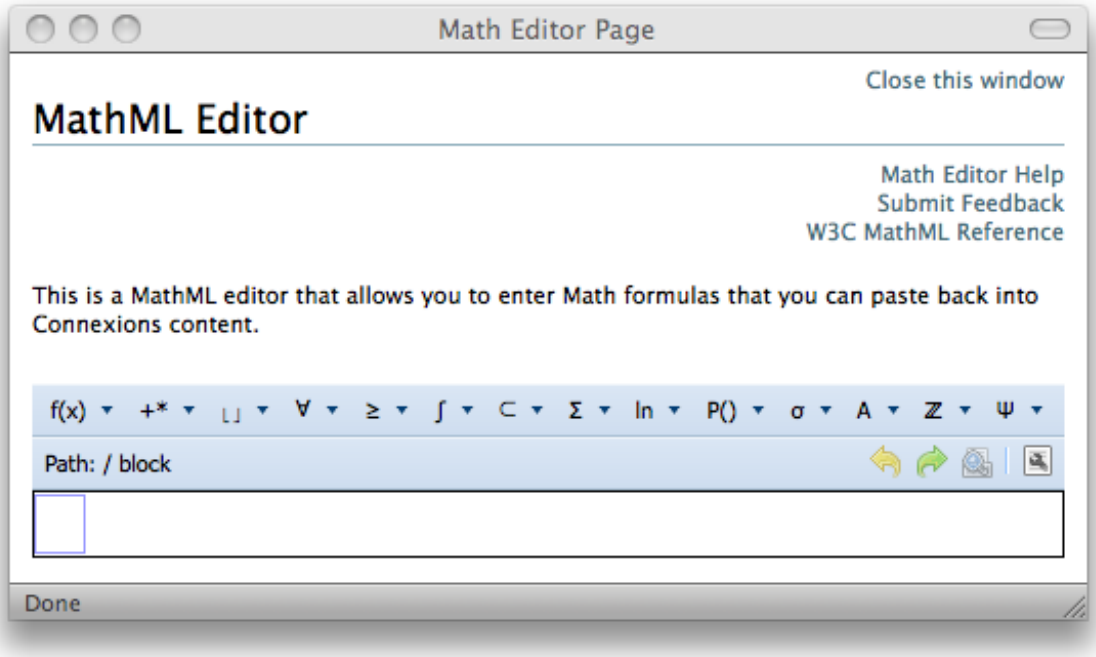


Figure 4.2: Once clicked, a popup window will appear containing the Math Editor

4.2.3 Using the Popup Window

The popup window (Figure 4.2) 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.

4.2.4 User Interface

The Editor has 4 main sections, detailed below. The toolbar (Section 4.2.5: 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 4.2.6: Editing Area) is located below the toolbar and contains the math that is being edited.

4.2.5 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 4.2.6: Editing Area). **Note:** The editor does not infer multiplication and addition. See Nuances (Section 4.2.15: Nuances / Limitations) for how to insert next to existing math by wrapping (Section 4.2.15.2: Wrapping Math with Math) existing math. Explain the different sections, when it's enabled, how things get inserted, and Keyboard Entry (Section 4.2.12: 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 4.2.6: 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 4.2.7: Math Tree)) and the path represents where in the tree the cursor (Section 4.2.10: Cursor) currently is. The path (and context (Section 4.2.11: 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 4.2.12: 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.

4.2.6 Editing Area

This is the main area for creating math. It begins empty, but math can be pasted directly in here from Connextions. The tutorials 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 4.2.7: Math Tree).
- Colors (Section 4.2.8: Colors) are used in this area to denote required information and contextual clues.
- Content vs Presentation Math (Section 4.2.9: Content vs. Presentation) discusses the two different types of math the editor supports.
- The cursor (Section 4.2.10: 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 4.2.11: Context) provides visual cues.
- Keyboard (Section 4.2.12: Keyboard Input) strokes are discussed in detail.
- Finally, empty blocks (Section 4.2.14: Blocks) are discussed below.

4.2.7 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 4.3. 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.

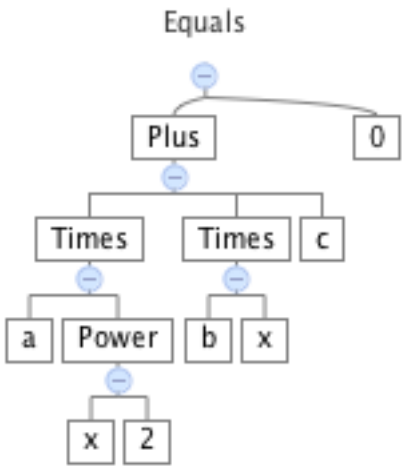


Figure 4.3: $ax^2 + bx + c = 0$ as a tree

4.2.8 Colors

Color notation (legend)

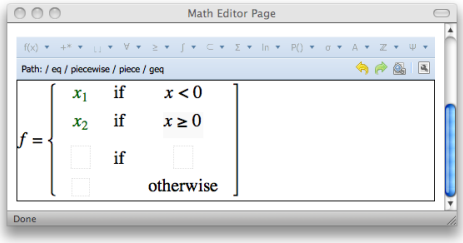
$$\text{f} = \begin{cases} x_1 & \text{if } x < 0 \\ x_2 & \text{if } \quad > 0 \\ & \text{otherwise} \end{cases}$$

- f : The location where text is currently being entered is represented as a box with a blue border (see Text Input (Table 4.2: 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 4.2.9: 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 4.2.11: 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 4.2.14: Blocks) for details.
- x_2 : The current selection is denoted by a light blue background. See Copy and Paste back to Modules for details.

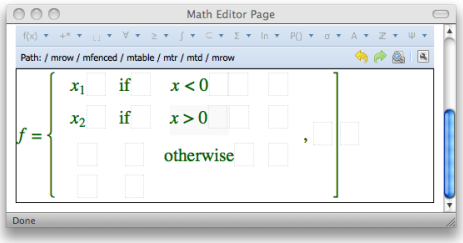
4.2.9 Content vs. Presentation

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



(a)



(b)

Figure 4.4: 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 ...

4.2.10 Cursor

The Math Editor can be used entirely from the keyboard (See Keyboard Input (Section 4.2.12: 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 4.2.12: 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 4.2.11: 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^` and parse the expression (Section 4.2.12: 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 4.2.10.1) for what can be done next.

4.2.10.1

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

4.2.11 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 4.2.10: Cursor) relative to existing math in the editing area (Section 4.2.6: Editing Area) and is displayed using a gray background (Section 4.2.8: 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.

4.2.12 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:

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 4.2.5: 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 4.2.5: 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 `[U+2318]` key on Apple computers)
- Navigation keys move the cursor through the math
- Modification keys change the math in some way

Category	Key	Condition	Action
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)
	C	Math is selected	Copies the selected Math to the clipboard
	V	Math is selected	Pastes MathML from the Clipboard (can be from other sources)
	Z		Undoes one step in the editor
	Y	Ctrl+Z was just pressed	Redoes one step in the editor
	E		Opens full-source editing
Navigation	Tab Shift+Tab		Moves to the next/previous free block
	Left / Right		Moves to the previous/next element in the Math
	Shift+Left / Shift+Right	After / Before the Context (Section 4.2.11: Context)	Selects the Context element (right next to the cursor)
		Before / After the Context	Selects the Context's parent
Modification	Enter		Attempts to parse the text entered next to the cursor

continued on next page

	Delete / Backspace	Cursor next to Math	Selects the Math Node (subsequent delete will remove the math)
		Math selected	Removes the node and replaces it with an empty block (a second press will remove the block as well)
		Cursor in block	Removes the empty block if it is allowed in MathML (in "a+b+c" any one variable can be removed, but addition requires at least 2 things to add)

Table 4.1

Text Input Examples

Type	**Input**	**Math Output**
Calculator	`a*x^2+b*x+c=1/2`	$ax^2 + bx + c = \frac{1}{2}$
	`a && b || c != a -> b`	$a \ and \ (b \ or \ c) \neq a \to b$
	`sin(x)^2+cos(x)^2=1`	$\sin^2(x) + \cos^2(x) = 1$
	`x_1+x_2<x_3`	$x_1 + x_2 < x_3$
Templates	`sum=n*(n-1)/2`	$\sum \quad = \frac{n(n-1)}{2}$
MathML	`<pi />`	π
	`&#1207x;`	

Table 4.2

4.2.13 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.

4.2.14 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)

4.2.15 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.

4.2.15.1 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 and manually replacing the <interval/> with a <condition/>.

4.2.15.2 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. 160) or with help of the toolbar (Toolbar, p. 160). 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:

Step 1. Move the cursor to the left of $bx + c$ but make sure the context (Section 4.2.11: Context) is **only** around $bx + c$ and that $bx + c$ is **not** selected (Section 4.2.10.1). This can be done by clicking the "+" sign.
Step 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.

Step 1. Select $bx + c$ but make sure **only** $bx + c$ is selected (Section 4.2.10.1). This can be done by double clicking the "+" sign.
Step 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.
Step 3. From the toolbar (Section 4.2.5: Toolbar) select the combiner operation. This should create at least one empty block.
Step 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.
Step 5. Select another empty block.
Step 6. From the toolbar, insert the operation.

4.3 MathML Editor: The Basics[8]

4.3.1 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:

[8] This content is available online at <http://cnx.org/content/m26312/1.2/>.

- how to access (Section 4.3.2: Accessing the editor) and use the editor,
- how to create (Section 4.3.3: Creating expressions) MathML expressions,
- how to navigate (Section 4.3.4: Navigating through an expression) through the structure of an expression,
- how to edit (Section 4.3.5: Editing expressions) expressions,
- how to insert (Section 4.3.6: Inserting expressions into a module) expressions created in the editor into a module,
- how to delete (Section 4.3.7: Deleting expressions) expressions,
- how to use the menu buttons (Section 4.3.8: 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 4.3.3: Creating expressions) and also using the toolbar menus (Section 4.3.8: Creating expressions with the toolbar menus).

You will also learn about the basic layout (p. 163) of the MathML editor, the context (Section 4.3.4.1: What is going on here?) of your cursor within the editing area, and how to select or highlight (Section 4.3.6.1: 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 4.2).

4.3.2 Accessing the editor

Currently, the MathML editor is only supported in Mozilla Firefox[9] . If you use Internet Explorer or another browser, you can download Firefox for free under the link.

When editing a module (Section 2.3.4.3: Edit the index.cnxml file with the Edit-In-Place Editor) using Mozilla's Firefox browser, click on an element within the module to open a blue editing box (Figure 4.5).

[9] http://www.mozilla.com/firefox/

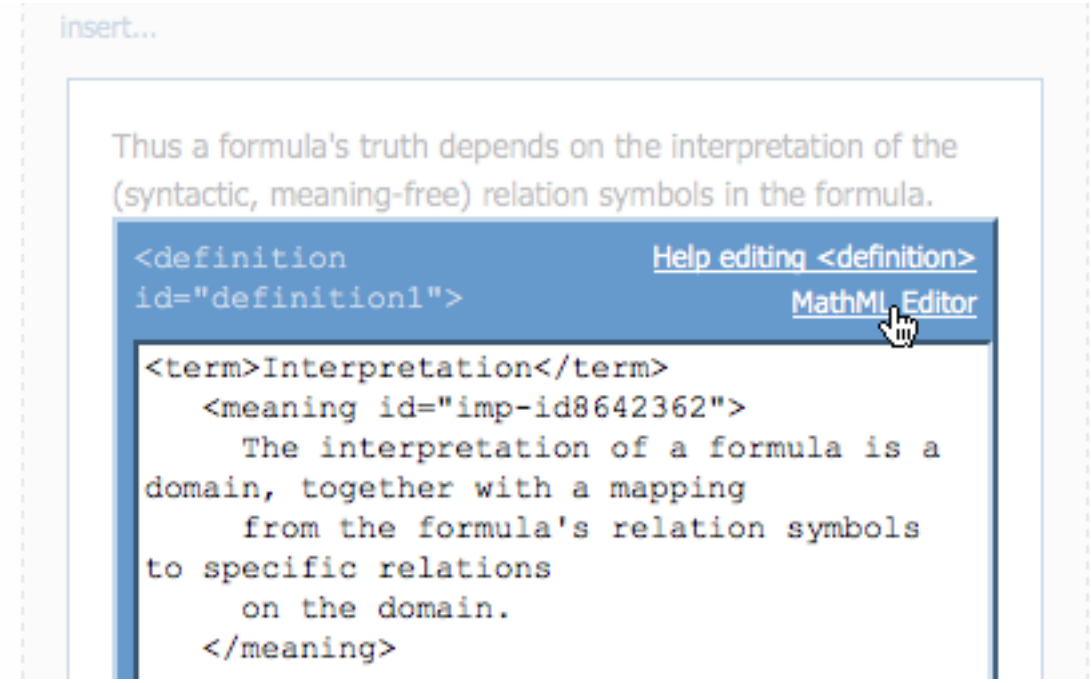


Figure 4.5: Begin editing and in the top-right corner is a MathML Editor link

On the top-right hand side of the box is a "MathML Editor" link which will open up the editor (Figure 4.6).

Figure 4.6: Once clicked, a popup window will appear containing the Math Editor

The popup window (Figure 4.6) 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 4.2.5: Toolbar) and the main editing area (Section 4.2.6: Editing Area).

The toolbar consists of

- a Menu (Menu, p. 153) containing mathematical operators,
- the Path (Path, p. 154) bar to help you find the context of your cursor,
- buttons for Undo/Redo,
- a Preview (Preview, p. 154) button, which will display the math as it will look in a published module,
- and a View Source (View Source, p. 154) 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 4.3.6: Inserting expressions into a module) below for how to move math from Connexions to the math editor and back.

4.3.3 Creating expressions

When you first open (Section 4.3.2: Accessing the editor) the MathML editor, the Editing area should have one blank box in it, called a "block (Section 4.2.14: 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 4.2.12: 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 4.3.8: Creating expressions with the toolbar menus).

4.3.3.1 An example

With a blank MathML editor open (Section 4.3.2: Accessing the editor), try typing in the following (or copying it from here and pasting into the editor):

```
x^2+b*x
```

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 `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*x`.

4.3.4 Navigating through an expression

If you have just created (Section 4.3.3: 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. 154) bar. In this position, the Path bar should display `Path: / plus` and the entire expression should have a light grey background.

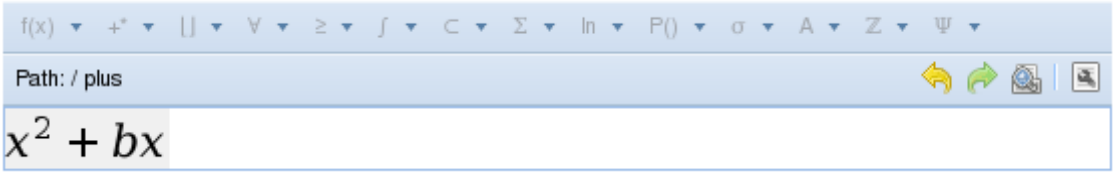


Figure 4.7: The context of the cursor is the entire plus operation.

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.

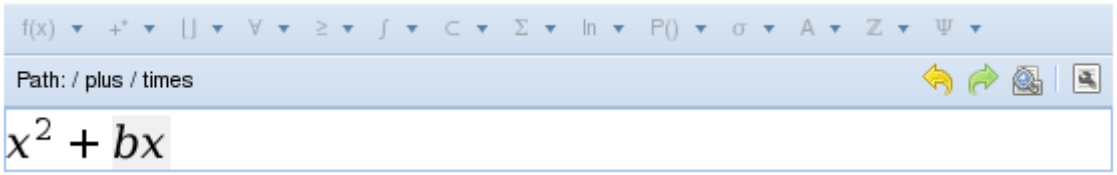


Figure 4.8: The context of the cursor is thetimes operation.

Press the left arrow again. Now the x should appear within a block and be selected. The path will change again to `Path: / plus / times / ci` The term `ci` represents an "identifier", in this case our variable x.

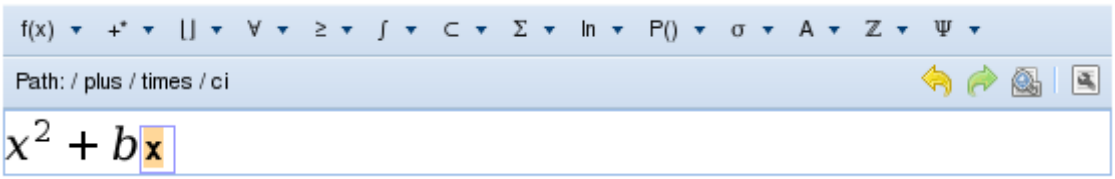


Figure 4.9: The context of the cursor is the ci block.

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 `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 `Path: / plus` and the entire expression should have a light grey background.

4.3.4.1 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 `times`. The product "b times x" is just one argument within the `plus` operation. The other argument of the `plus` operation is of course x^2. This `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 `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 4.2) for an example (Section 4.2.7: 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 4.2.11: 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 4.3.5.2: Creating a new operation).

4.3.5 Editing expressions

4.3.5.1 Adding an argument to an existing operation

Be sure the editing area of the MathML editor contains the expression we entered (Section 4.3.3: 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 `times` operation. Then type in `+c` and press `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 `Enter`. Experiment with adding more arguments to the `plus` operation just after the `power` operation or just before the `times` operation.

4.3.5.2 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 `power` operator, while we want to make the entire `power` operation one argument of a `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 4.2.10: Cursor) of our cursor carefully, or we will get different math. Make sure your cursor is on the left hand side of the `power` operation, the path says `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 `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 `plus` operation, where the path bar would read `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:

$(ax)^2 + bx + c$

4.3.6 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.

4.3.6.1 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 4.2.15.2: Wrapping Math with Math)" your existing math.

4.3.7 Deleting expressions

There are several ways to delete part or all of your expression. You can either move your cursor to the context (Section 4.2.11: Context) of the portion of your expression you want to delete, then push `Delete`, or you can select (Section 4.3.6: 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`.

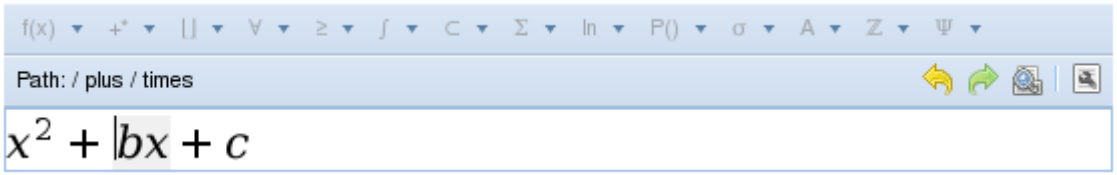


Figure 4.10: The context of the cursor is the times operation.

Press `Delete` once, and that part of your expression should be highlighted in blue.

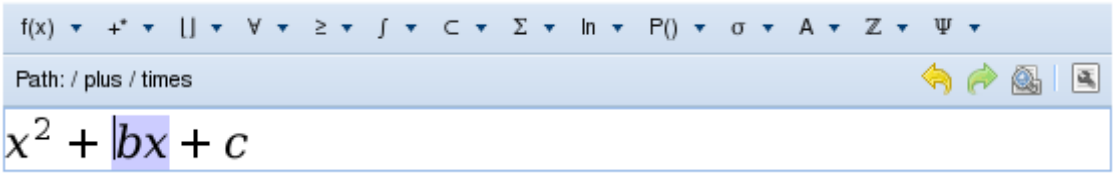


Figure 4.11: The times operation is highlighted.

Press `Delete` again and that portion will delete.

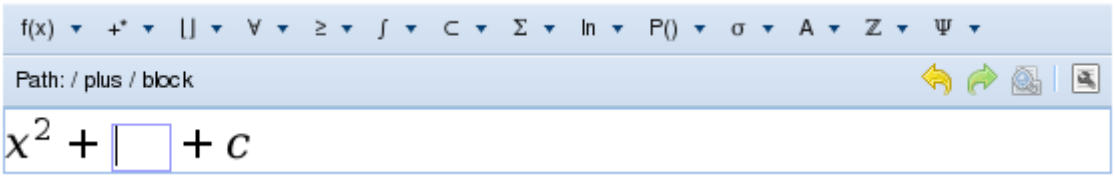


Figure 4.12: The cursor will move to the next argument.

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 `ci` element c.

4.3.8 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 4.6) 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 $\sqrt[6]{3}$).

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 4.3.8.1: Creating a new expression) using only the toolbar, or find a more complicated and in-depth example in the module MathML Editor: Intermediate (Section 4.4).
- 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.

4.3.8.1 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 4.3.3: Creating expressions). We'll first insert $x^2 + bx$ (p. 169), then add c (Section 4.3.8.2: Adding an argument to an existing operation) as an argument to the existing `plus` operator, and finally add a (Section 4.3.8.3: 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 4.3.7: Deleting expressions) it first.

The expressions x^2 and bx stand in relation to one another as arguments of a `plus` operator, so we will start by inserting a `plus` operator and then inserting successively smaller chunks of the expression.

Step 1. Click the `plus` operator from the toolbar.
Step 2. To create the x^2 expression, click within the first block in the `plus` operation, then select the `power` operator from the toolbar.
Step 3. Click within the first block in the `power` operation, and type in x.
Step 4. Click within the second block in the `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 `plus` operation, and select the `times` operator from the toolbar.
Step 6. Click within the first block of the `times` operation, and type in b.
Step 7. Click within the second block of the `times` operation, and type in x.

4.3.8.2 Adding an argument to an existing operation

There are two ways we can add the final argument c to the `plus` operator. One way would have been to add an extra argument to the `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 `plus` operation. Essentially we are inserting the entire operator as an argument of the first `plus` operator, but the editor will correctly condense this and turn all three blocks into arguments of one `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 `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 `plus` operator, and then add c as the second operator. This is known as "wrapping (Section 4.2.15.2: 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 `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`.

4.3.8.3 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.

4.4 MathML Editor: Tutorials[10]

4.4.1 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 (Simple text entry, Special characters)
-
-
-

4.4.2 Quadratic Equation

As a simple example, we'll step through several ways of writing the well-known Quadratic Equation (with real or complex coefficients):

$$ax^2 + bx + c = a\left(x - \frac{-b + \sqrt{b^2 - 4ac}}{2a}\right)\left(x - \frac{-b - \sqrt{b^2 - 4ac}}{2a}\right)$$

[10] This content is available online at <http://cnx.org/content/m24560/1.1/>.

4.4.2.1 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.
Step 2. Enter the following into the main editing area "`a*x^2+b*x+c=a*(x-(-b+root)/(2*a))*(x-(-b+root)/(2*a))` ". See below for details (p. 171) .
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.

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.

4.4.2.2 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 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.

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.

4.4.2.3 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:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

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 4.4.2.1: 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[11] 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 4.4.3: 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` "
- A variable with both subscripts and superscripts can be entered using "`msubsup` "
- Backets like $(-\infty, 0]$ can be added using "`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, "`munderover` " places math directly above or below.
- A table can be added using "`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:

$$\sum_{i=1}^{n}\left(i^2\right)$$

$$\sum_{i\in S}\left(i^2\right) \quad S=\{\,i\,|i>0 \;\; and \;\; i\leq n\}$$

Changing the range these operations required switching to the MathML source and being familiar with the W3C MathML Specification (Section 4.4.3: W3C MathML Specification) . To change the former to the latter, we start with a clean "`sum` " operation. Then, to decrease the amount of hand editing, we can type "`i in S` " to the right of the equal sign. Then, we switch to MathML Source and replace every occurrence of "`interval` " with "`condition` " and removing the special "<`block ...`> " element just above the </`condition`> .

4.4.3 W3C MathML Specification

The W3C MathML Specification[12] 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[13]) as well as a way to lay out variables and formulas (Presentation MathML[14]).

[11] http://www.unicode.org/charts/PDF/U2200.pdf
[12] http://www.w3.org/TR/MathML2/
[13] http://www.w3.org/TR/MathML2/chapter4.html
[14] http://www.w3.org/TR/MathML2/chapter3.html

Chapter 5

Appendix

5.1 Grilling a Good Steak[1]

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

1. pour beer into large bowl
2. add chili powder to tase
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 3.3.2) which focuses on more advanced CNXML tags. For more marinades see the Angus Beef website[2] . Finally, a good resource is the *Steak Lover's Cookbook – William Rice*.

[1] This content is available online at <http://cnx.org/content/m10275/2.6/>.
[2] http://www.2eatcab.com

5.2 Grilling a Better Steak[3]

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.

5.2.1 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

5.2.2 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 tase
3. squeeze half lime into beer marinade
4. place steak in beer, let soak for 30 minutes

[3]This content is available online at <http://cnx.org/content/m10278/2.9/>.

T-Bone Steak

Figure 5.1: Upon successful completion of these modules, you should be able to grill a steak that looks just as good!

Steaks

(a) T-Bone (b) New York Strip

Figure 5.2: Upon successful completion of these modules, you should be able to grill a steak that looks just as good!

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!

How to grill the steak will be covered in The Advanced CNXML (Section 3.3.3). For more marinades see the Angus Beef website[4] . Finally, a good resource is the *Steak Lover's Cookbook – William Rice; Paperback.*

5.3 Grilling the Best Steak[5]

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

(a) T-Bone (b) New York Strip

Figure 5.4: Upon successful completion of these modules, you should be able to grill a steak that looks just as good!

5.3.1 Ingredients

Before we begin to cook I have compiled a list of ingredients.

Ingredients

- Salt
- Fresh ground pepper
- Lime
- Beer

[4] http://www.2eatcab.com
[5] This content is available online at <http://cnx.org/content/m10281/2.7/>.

- Chili powder
- T-Bone

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*.

5.3.2 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

5.3.3 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.

Temperature(F)	Description
140	Rare
150	Medium Rare
160	Medium
165	Medium Well
170	Well

Table 5.1

Remember that for safety's sake, always cook your steak to 160 F or until meat is no longer pink.

5.3.4 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 5.1 ***(Solution on p. 185.)***
For food safety, a steak should be cooked to a minimum temperature of what?

For more marinades see the Angus Beef website[6] . Finally, a good resource is the *Steak Lover's Cookbook – William Rice; Paperback.*

5.4 Example module for use of cnxml/mathml tags[7]

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 5.2: 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
See Also: enigma, **puzzle**

5.4.1 Section Name

A document can have sections, however they are **not** required.

5.4.1.1 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 5.5 (Optional Figure Name) shows how a figure will display in our system.

[6] http://www.2eatcab.com
[7] This content is available online at <http://cnx.org/content/m11216/1.3/>.

Optional Figure Name

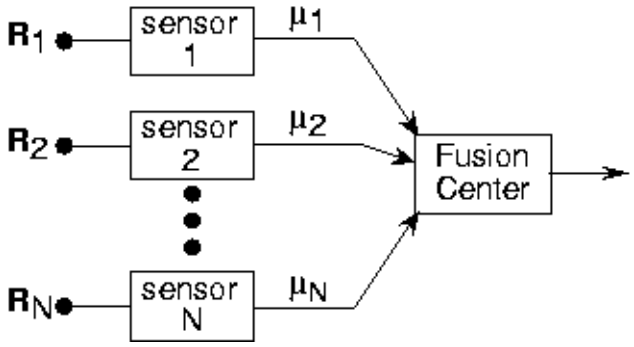


Figure 5.5: The caption of the figure would be here.

There are four different options for list. One type of list is a `named-item` list.

List name

first name - description of first item
second name - description of second item
third name - description of third item
fourth name - description of fourth item

See the CNXML-0.5 Spec[8] for more information regarding list types.

Example 5.1: 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 5.2: Temperatures in 5 cities on 11/16/2002).

Temperatures in 5 cities on 11/16/2002

City	Degrees Fahrenheit	Degrees Centigrade
Houston	54	12.22
Chicago	37	2.78
Minneapolis	31	-0.56
Miami	78	25.56
Phoenix	70	21.11
Mean	**54.000**	**12.22**
Median	**54.000**	**12.22**
Variance	**330.00**	**18.166**
SD	**101.852**	**10.092**

Table 5.2

[8] http://cnx.rice.edu/cnxml/0.5/spec/

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.

```
≫ syms t
≫ laplace(exp(t))
ans = 1/(s-1)
≫ laplace(t*exp(-t))
ans = 1/(s+1>^2
```

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

$$\begin{aligned} P(f) &= \int_{-\infty}^{\infty} p(t)\, e^{-(i2\pi f t)} dt \\ &= \int_0^{\Delta} e^{-(i2\pi f t)} dt \\ &= \frac{1}{-(i2\pi f)} \left(e^{-(i2\pi f \Delta)} - 1 \right) \end{aligned} \tag{5.1}$$

See our discussion of Content MathML (Section 4.1) for a more basic discussion of math. See also our MathML Extensions page[9] for more `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 5.2 ***(Solution on p. 185.)***
What is a composite number?

[9] http://bunker.ece.rice.edu:8080/mntb/wikis/mathml/CSymbol

Solutions to Exercises in Chapter 5

Solution to Exercise 5.1 (p. 181)
160 F or until the juices run clear and the meat is no longer pink
Solution to Exercise 5.2 (p. 184)
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 NameOne 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 3.15

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.

Bibliography

[1] Harald Martens; Tormod Nas. *Multivariate Calibration.* John Wiley & Sons Ltd., Chichester, 1989.

[2] Kim Esbensen; Tonje Midtgaard; Suzanne Schonkopf. *Multivariate Analysis in Practice.* Camo AS, Trondheim, 1994.

[3] William Shakespeare. A midsummer night's dream. In Alfred Harbage, editor, *William Shakespeare: The Complete Works*, chapter 2.1, pages lines 107–114. The Viking Press, 1969.

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
Alt text, 67
alternate, § 2.4(66)
annotation, § 1.3(9)
apply, § 4.1(145)
attribute, 92, § 3.2.1(116), 117
audio, § 2.4(66)
authentic, 44
author, § 1.1(1)
Authors, 34, 53

B
beef, § 5.1(175)
bibtexml, § 3.3.4(139)
block, 70
bug, § 1.1(1), § 1.2(8)
bugs, 8

C
CALS, § 3.1.3(109)
CALS table, § 3.1.3(109)
caption, § 3.3.2(128)
categories, § 3.1.2(95), § 3.3.3(136)
category, § 3.1.2(95), § 3.3.3(136)
character references, 118
ci, § 4.1(145), 146
cn, § 4.1(145), 146
CNXML, § 3.1.1(91), 91, § 3.1.3(109), § 3.3.1(120), § 3.3.2(128), § 5.2(176), § 5.3(180), § 5.4(182)
cnxml example, § 5.1(175)
codeblock, § 3.3.2(128)
codeline, § 3.3.2(128)
collaborate, § 2.6(87)
collection, § 2.2(28)
collection composer, § 2.2(28)
Connexions, § 3.1.1(91), § 3.3.1(120), § 4.3(160)
Connexions Markup Language, 91
connexions project, § 1.1(1)
content, § 2.3(38), § 3.1.1(91), § 3.3.1(120), § 4.2(151)
Content Commons, 9
content mathml, § 4.1(145)
Content Section, 120
copy, § 2.5(85)
Copyright Holders, 34, 53
course, § 2.2(28)
course composer, § 2.2(28)
csymbol, § 4.1(145), 146

D
definition, § 3.1.2(95), § 3.3.3(136), § 3.3.4(139)
derivative, § 2.5(85)
derive, § 2.5(85)
differentiation, § 4.1(145)
document, § 3.1.1(91), § 3.3.1(120)
documentation, § 4.3(160)
DTD, 117, § 3.2.2(119)
dynamic reference, 71

E
editing, § 2.3(38), § 4.2(151)
editor, § 4.3(160)
Editors, 53
elem, § 3.1.2(95), § 3.3.3(136)
embed tag, 82, 84
empty tag, 112, 117, 137
empty tags, 146
enigma, 182
entity, § 3.2.1(116)
entity references, 118
equation, § 3.3.2(128)
error, § 1.2(8)
error message, 8
example, § 3.1.2(95), § 3.3.3(136)
exercise, § 3.1.2(95), § 3.3.3(136)

F
figure, § 3.3.2(128)
flash, § 2.4(66)
function, § 4.1(145)

G
gardenia, 92
Gardenias, 92, 92
general information, § 1.1(1)
glossary, § 3.3.4(139)
grilling, § 5.1(175), § 5.2(176), § 5.3(180)
group, § 2.2(28), § 3.1.2(95), § 3.3.3(136)

H home page, § 1.1(1)

I I have read the Connexions Site License and I agree to be bound by its terms., 7
ID, 95
identifier, 146
image, § 2.4(66)
inline, 69, 70
instructor, § 1.1(1)
integration, § 4.1(145)

J java applet, § 2.4(66)

L locate, § 1.3(9)
log in, § 1.1(1)
longdesc, 67

M Maintainers, 34, 53
marinade, 175, 176, 181
marinates, 175, 176, 181
markup, § 3.1.1(91), § 3.3.1(120)
markup language, 116
math, § 4.2(151)
mathml, § 4.1(145), § 4.3(160), § 5.4(182)
matrix, § 4.1(145)
meaning, § 3.1.2(95), § 3.3.3(136)
media, § 3.3.2(128)
media subtype, 69
meta-markup language, 116
metadata, 12, § 2.3(38)
Metadata Section, 120
mime-type, 83
module, § 2.2(28), § 2.5(85)
modules, § 2.3(38), § 4.2(151)
movie, § 2.4(66)
mpeg, § 2.4(66)
multimedia, § 2.4(66)
multimedia object, § 2.4(66)
music, § 2.4(66)
My Account, 6, 8
MyCNX, 4

N n-ary operator, 147
namespace, § 3.1.1(91), § 3.2.2(119), § 3.3.1(120), 121
namespace prefix, 119
namespaces, 119
nesting, § 3.2.1(116)
notes, § 1.3(9)

O operator, § 4.1(145)

P patch, § 2.6(87)
prefix, § 4.1(145)
Prefix notation, 145
problem, § 3.1.2(95), § 3.3.3(136), 182
proof, § 3.1.2(95), § 3.3.3(136)
puzzle, 182

Q quicktime, § 2.4(66)

R real audio, § 2.4(66)
report, § 1.1(1)
role, § 2.3(38)
root node, 119
rule, § 3.1.2(95), § 3.3.3(136)

S section, § 3.3.2(128)
snapshot, § 2.5(85)
solution, § 3.1.2(95), § 3.3.3(136)
sounds, § 2.4(66)
src, 83, 84
statement, § 3.1.2(95), § 3.3.3(136)
steak, § 5.1(175), § 5.2(176), § 5.3(180)
student, § 1.1(1), § 1.3(9)
stylesheets, 91
subfigure, § 3.3.2(128)
suggest edits, § 2.6(87)

T T-Bone, 134, 181
table, § 3.1.2(95), § 3.1.3(109), § 3.3.3(136)
tables, § 3.1.3(109)
tag, § 3.1.1(91), 91, § 3.3.1(120)
Tags, 117
task, § 1.2(8)
term, § 3.1.2(95), § 3.3.3(136)
Title, 120
token, § 4.1(145), 146
Translators, 53
tutorial, § 2.2(28), § 3.2.2(119), § 5.2(176), § 5.3(180)

U unicode, § 3.2.1(116), 118

V valid, § 3.2.1(116), 117
vector, § 4.1(145)
view, § 1.3(9)
vocabulary terms, 182

W well-formed, § 3.2.1(116), 117

X XML, § 3.1.1(91), § 3.2.1(116), 116, § 3.2.2(119), § 3.3.1(120)
XML declaration, 117
xmlspy, 44

Attributions

Collection: *Connexions Tutorial and Reference*
Edited by: Connexions, Adan Galvan, Brent Hendricks, Mark Husband
URL: http://cnx.org/content/col10151/1.21/
License: http://creativecommons.org/licenses/by/1.0

Module: "Introduction to Connexions"
By: Mark Husband
URL: http://cnx.org/content/m10884/2.26/
Pages: 1-8
Copyright: Mark Husband, Connexions
License: http://creativecommons.org/licenses/by/1.0

Module: "Reporting a Connexions Bug"
By: Mark Husband
URL: http://cnx.org/content/m11836/1.7/
Pages: 8-9
Copyright: Mark Husband
License: http://creativecommons.org/licenses/by/1.0

Module: "Viewing Connexions Content"
By: Connexions, Mark Husband
URL: http://cnx.org/content/m11837/1.14/
Pages: 9-12
Copyright: Connexions, Mark Husband
License: http://creativecommons.org/licenses/by/1.0

Module: "MyCNX and the Work Areas"
By: Connexions, Mark Husband, Adan Galvan
URL: http://cnx.org/content/m10885/2.22/
Pages: 13-28
Copyright: Connexions, Mark Husband, Adan Galvan
License: http://creativecommons.org/licenses/by/1.0

Module: "How to Use the Collection Composer"
By: Connexions, Sarah Coppin, Mark Husband
URL: http://cnx.org/content/m10288/2.24/
Pages: 28-38
Copyright: Connexions, Sarah Coppin, Mark Husband
License: http://creativecommons.org/licenses/by/1.0

Module: "Editing Modules"
By: Connexions, Mark Husband, Adan Galvan
URL: http://cnx.org/content/m10887/2.28/
Pages: 38-66
Copyright: Connexions, Mark Husband, Adan Galvan
License: http://creativecommons.org/licenses/by/1.0

Module: "Adding Multimedia to Your Connexions Content"
By: Connexions, Adan Galvan, Mark Husband
URL: http://cnx.org/content/m12660/1.16/
Pages: 66-85
Copyright: Adan Galvan, Mark Husband
License: http://creativecommons.org/licenses/by/1.0

Module: "Derived Copy"
By: Mark Husband
URL: http://cnx.org/content/m11804/1.5/
Pages: 85-86
Copyright: Mark Husband
License: http://creativecommons.org/licenses/by/1.0

Module: "Suggest Edits"
By: Mark Husband
URL: http://cnx.org/content/m11817/1.9/
Pages: 87-90
Copyright: Mark Husband
License: http://creativecommons.org/licenses/by/1.0

Module: "Using Basic CNXML in Edit-In-Place"
By: Elizabeth Gregory, Connexions
URL: http://cnx.org/content/m14394/1.4/
Pages: 91-95
Copyright: Elizabeth Gregory
License: http://creativecommons.org/licenses/by/2.0/
Based on: The Basic CNXML
By: Ricardo Radaelli-Sanchez, Brent Hendricks
URL: http://cnx.org/content/m9000/2.34/

Module: "Advanced CNXML using Edit-in-Place"
By: Elizabeth Gregory, Connexions
URL: http://cnx.org/content/m14395/1.4/
Pages: 95-109
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: "CALS Table"
By: Elizabeth Gregory
URL: http://cnx.org/content/m14396/1.5/
Pages: 109-116
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: "XML Basics"
By: Connexions, Sarah Coppin, Brent Hendricks, Chuck Bearden
URL: http://cnx.org/content/m9002/2.24/
Pages: 116-119
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/
Page: 119
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: 120-128
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: 128-136
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: 136-139
Copyright: Ricardo Radaelli-Sanchez
License: http://creativecommons.org/licenses/by/1.0

Module: "CNXML Reference Extensions"
By: Adan Galvan
URL: http://cnx.org/content/m11215/1.6/
Pages: 139-144
Copyright: Adan Galvan
License: http://creativecommons.org/licenses/by/1.0

Module: "Content MathML"
By: Sarah Coppin, Brent Hendricks
URL: http://cnx.org/content/m9008/2.15/
Pages: 145-151
Copyright: Sarah Coppin, Brent Hendricks
License: http://creativecommons.org/licenses/by/1.0

Module: "Math Editor Introduction"
Used here as: "Introduction to the MathML Editor"
By: Philip Schatz
URL: http://cnx.org/content/m24561/1.1/
Pages: 151-160
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: 160-170
Copyright: Natalie Weber
License: http://creativecommons.org/licenses/by/3.0/

Module: "Math Editor Tutorials"
Used here as: "MathML Editor: Tutorials"
By: Philip Schatz
URL: http://cnx.org/content/m24560/1.1/
Pages: 170-173
Copyright: Philip Schatz
License: http://creativecommons.org/licenses/by/3.0/

Module: "Grilling a Good Steak"
By: Sarah Coppin, Ricardo Radaelli-Sanchez
URL: http://cnx.org/content/m10275/2.6/
Page: 175
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: 176-180
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: 180-182
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: 182-184
Copyright: Charlet Reedstrom
License: http://creativecommons.org/licenses/by/1.0

Connexions Tutorial and Reference
Tutorial and reference materials for getting started with Connexions

About Connexions
Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.