



CNX.ORG & CNXML

Mini-training and guide on best usage for cnx.org and its coding languages.

Useful Links

- CNX Tutorial and Reference - <http://cnx.org/content/col10151/latest/>
- CNXML Tag Help - <http://cnx.org/eip-help/tags>
- Authoring Content - <http://legacy.cnx.org/help/authoring>
- Adding Multimedia To Your Content - <http://cnx.org/content/m12660/latest/>
- Guide to MathML - <http://cnx.org/content/col10963/latest/>

Nomenclature

Module vs. Collections

Collections and modules are the fundamental objects in the OpenStax CNX repository. *Modules* are the basic chunks of content in the repository. All content is composed into these simple and easily digestible modules. *Collections* are groups of modules combined in a particular order to create a book, manual, lab report, etc.

If you are viewing an individual module, you simply see its contents. However, if you are viewing the same module within a collection, not only do you see its contents, but you also can view other modules in the collection by expanding the contents tab for the collection.

Please note, once the editor is launched on the new site (cnx.org), "modules" will become "pages" and "collections" will become "books". It is more intuitive to think of these chunks of content as pages and books rather than modules and collections.

Subcollection is another term used often on CNX's Legacy site. It is helpful to think of subcollections like chapters in a textbook. If the "collection" is the entire book (i.e. AP Physics), then the subcollection might be the chapter (with several modules) on Particle Physics.

Section

This is a word to use with care when working on cnx.org. Try not to use the word "section" to refer to a module or a few modules (better known as a collection or a subcollection). In the Legacy editor, which is what we use now, a "section" is a chunk of content within a module. Within a section, one can put a combination of paragraphs, figures, notes, tables, etc. Generally speaking, almost all content in a module should fit within a section (with the occasional exception of one/a few introductory paragraphs to the module). In [this module](#), "Early Accelerators" and "Modern Behemoths and Colliding Beams" are examples of sections that group similar content within a module.

Paragraph

As mentioned above, a paragraph is usually the child of a section (it goes within a section). Although paragraphs can stand alone outside sections, if you have a more than one paragraph that you want to group together under a similar heading, put them in a section and give the section the appropriate title. Again, please reference the module linked to above.

Module/Collection Status: Created, Checked Out, Modified, Published

- **"Created"** describes an item (i.e. module or collection) that the user has started "from scratch" but not published. When logged in on cnx.org, if you go to "MyCNX" and click "Create a new module" or "Create a new collection" under "Create and edit content", the item you create will have the status "Created" until you publish it. Therefore, "Created" is a status typically reserved for new items that the user is not ready to make public yet. (Once an item is made public, it can be modified, but previous versions will remain accessible through the item's Version History.)
- **Checked out** describes an item that has been published at least once, then checked out by a user with maintainer or author privileges (these are assigned in the "Roles" tab of the module or collection and must be accepted by the other party). However, the item has not been modified from its last published state.
- **Modified** describes an item that has been published at least once, then checked out by a user with maintainer or author privileges (see above bullet) and changes have been made. The module has not been re-published with these changes.
- **Published** describes an item that has been made public. This could be an item that was recently created and published, or an older item that has been checked out,



modified, and re-published many times. When no one has the module checked out/modified, it is published.

Finding and Viewing Published Content on cnx.org

Old Site (legacy.cnx.org)

<http://cnx.org/content/col10151/latest/>

(Click the "Contents" tab, then click "Viewing and Finding Content". *Please note that although you are viewing this collection with the new site interface -- that's why it's pretty -- the guide itself concerns the Legacy site.*)

New Site (cnx.org)

<https://openstaxcnx.zendesk.com/hc/en-us/categories/200179264-Viewing-Content>

We are housing our "Help" section for the new site on zendesk, which is why the url looks different from a normal url for a collection on cnx.org. This documentation may be useful to further understand basic concepts of CNX, but won't be necessary until we launch the editor on the new site (date TBD).

Authoring Content

When to use Full-Source

- It is only possible to edit a module's **Glossary** content when you have clicked the Full-Source edit link at the top of the module editor.
- **If** you are making many content changes at once and wish to see them all at the same time and one of the below is true:
 - you know the tagging scheme well enough to code it from scratch...
 - the tags are already in place and you are merely modifying existing content in the module...

...Full-Source edit is useful, but be sure to save your work before returning to Edit-In-Place.

When to use Edit-in-Place

The CNX editor (or "Edit-in-Place") is helpful for creating elements -- such as sections, paragraphs, notes, etc. -- in a module without writing them manually in the code and tagging



them, and for when you wish to concentrate on only one item at a time. The CNX editor walks you through creating an element, whereas in Full-Source, you must know all the proper tags. (For example, in an "exercise" element, there should be <problem> tags and <solution> tags.) Unless you have a specific reason to use Full-Source, the editor is much easier because it provides the tags required to create and save different elements in the module.

The editor is also useful for isolating elements that are causing problems, because the save button on the Edit-In-Place tool will not work if there is a problem in the code, and will tell you where the error is more specifically than the Full-Source tool will.

How to create a module or collection

1. Log in on cnx.org (currently, this will re-direct you to our Legacy site).
2. Click the "MyCNX" tab.
3. Click "Create a new collection" under "Create and edit content"
4. Accept the license agreement
5. Fill in Metadata (information about the collection). You may skip non-required fields (required fields display a red square).
6. On the next page, add subcollections and/or modules to your collection. Remember that you can use subcollections to organize modules, much like a chapter in a textbook that is divided further into sub-chapter sections. On cnx.org, the "chapter" is the subcollection and the "sub-chapter sections" are modules.
7. You may choose to assign "Roles" to other users, and/or to indicate a template (such as CCAP Physics) on the Parameters tab. All OpenStax College textbooks have a template created by our developers. This affects styles and other elements within the book.
8. Don't forget to publish your collection when you're ready to make a version public.

How to derive a module or collection

1. Log in on cnx.org (currently, this will re-direct you to our Legacy site) with a username that does NOT have maintainer or author permissions on the content you wish to derive.
2. Navigate to the desired content
3. Under "Reuse/Edit" toward the top right, click to "Reuse/Edit" the module or collection, as desired.
4. In the pop-up box, select "Derive a copy" and press "Continue"
5. Accept the license agreement
6. You may now make desired changes to the content





Important differences between deriving modules vs. collections:

When you “derive a copy” of a **collection**, the original modules are reused in this new collection. The only changes you may make to a derived collection are to:

- o remove and/or add modules
- o change the subcollection structure

If you wish to modify the contents of a module, you must “derive a copy” of that **module**, creating a new module (with a new module ID) so that changes you make will not affect the original content which another user created.

Tips and Time-Savers

- Avoid two people editing the same module at once. If two people are working on the same module and one publishes, all the work that has been done by the person who did not publish is lost.
- **DO NOT** attempt to derive a copy of a module when signed in as a user with maintainer or author permissions on that content. If you are not given the option to derive a copy, this is because the logged in user has one or both of these permissions on the given content. Changes you make to this content will affect the original modules, because you are editing the original modules, not derived copies.

Troubleshooting: Common Errors

(This is the list of possible errors that the Digital Media Specialists look for when checking CNXML written by Words & Numbers contractors.)



BAD LINKS to different parts of the module (or to other modules) will show up as “Reference” or “Section” on the Edit-In-Place view. However, all links will show up as “Reference” immediately after they are edited and saved; the page will have to be refreshed before their auto-generated anchor text will show up.



Element IDs

Nearly all elements have IDs. These appear on the opening element tag (i.e. `<section id="###">`, `<equation id="###">`, etc. These IDs will be auto-generated, but you can also write your own IDs if there's some kind of scheme you'd like to use (for example, W&N often uses a specific naming scheme with figure IDs). The main thing to be careful of with IDs is not duplicating them. This can happen if you copy from another section — for example, you copy all the code for a figure to give yourself a template for writing the next one — and do not remove the copied IDs. Don't worry: Full-Source and the editor will let you know if an ID is duplicated. You can simply remove it and replace it with your own or let one be auto-generated when you save.



We need those ID tags in order to link to different sections within the module.

Content Sections

After a few introductory paragraphs/figures/notes that can be outside a section, all sections should have titles.

Figures

- 1) IDs often look like this: `<figure id="CNX_Psych_02_05_variables">`
- 2) Figures should include `<media>` tags, beginning after the opening `<figure>` tag and ending before the `<caption>` tag
- 3) The caption should be in the caption field at the bottom of the box, or in the `<caption>` tag if you are editing by hand.
- 4) Links to figures (which typically appear in the few lines above the figure) need to be auto-generating with a number (i.e. "Figure 1").

Notes

- 1) Ensure they have both an ID and a class: `<note id="fs-idm46705312" class="psychology link-to-learning">`; the ID can be auto-generated, but the class is not, and the styling will not show up in any PDF if the class is not correct.
- 2) Be sure note titles do not include the word "Label" or "Feature" unless you are informed otherwise (consult the tagging legend)



Tables

- 1) Make sure formatting is consistent throughout the table (ex. content does not bleed over into other cells, font is consistent, etc.)
- 2) Ensure that content is meant to be displayed in a table (by looking at surrounding text for mention of a table)

Math

- 1) Math should be inside a `<m:math>` tag; equations should have an auto-generated ID and be in an `<equation>` tag.
- 2) If Math is in a table, make sure it is aligned correctly (ex. equal signs in a series of equations are aligned, etc.).
- 3) If Math is vertical and not in a table (and has been aligned with space tags instead, which is never a good idea), put it in a table aligned at the equal signs.
- 4) If you see the XML team has used the `"display=block"` attribute, note this possible coding error, as this attribute does not have a known function and seems to only cause alignment issues.



Many math issues can be solved by enclosing all the content inside the math tag with `<m:mrow>` tags

Summaries

They should have a auto-generated section ID and a class, such as `"section-summary"`

Review Questions

There should be a section ID and a class, such as `"review-questions"`



ALL SOLUTIONS THROUGHOUT END-OF-CHAPTER QUESTIONS should be contained in `<solution>` tags

Critical Thinking Questions, etc.

Be sure each of these sections has its own title, such as `"Critical Thinking Questions"` or `"Personal Application Questions"`, as well as its own class to ensure proper collation.

References

Be sure these are collating to bottom of module, if applicable, and that they have a class such as `"reference"`.



Glossary

Be sure terms have <term> tags within the content of the module (i.e. within the text itself), and are collating to bottom of module, if applicable.

