The Advanced CNXML*

Ricardo Radaelli-Sanchez

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

This is the final installment of my three part tutorial on the CNXML language. It is currently valid for the most recent release of the 0.3 language. The keywords contain a list of the tags described in this tutorial. Along with the example code in this module there is also an example module that has been growing throughout the tutorial.

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[1] . The `code` tag has a `type` 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 1

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

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

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`. The `problem` tag and the `solution` tag have no attributes.

*Version 2.22: Feb 24, 2004 11:51 am -0600
†http://creativecommons.org/licenses/by/1.0
[1]http://cnx.rice.edu/technology/cnxml/0.5/spec

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[2] .

Example 2

```
<exercise id='grilltest'>
  <problem>
    <para id='grilltestp1'>
      For food safety, a steak should be cooked to a minimum
      temperature of what?
    </para>
  </problem>
  <solution>
    <para id='sol1p1'>
      160&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 CALS Table

CNXML uses the industry standard CALS Table Model[3] 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 1: Steak Cooking Temperatures)). For a more complete description of the CALS Table consult the CALS Table Spec[4] .

3.1 table

The `table` tag marks the beginning of a table. It has an optional first child of name[5] and must contain one or more tgroup (p. 3) tags. The `table` tag also has many attributes, to find out more information consult the CALS Table Spec[6] .

[2] http://cnx.rice.edu/qml/intro/qml.xml
[3] http://www.oasis-open.org/specs/a502.htm
[4] http://www.oasis-open.org/specs/a502.htm
[5] "The Basic CNXML": Section Title <http://cnx.org/content/m9000/latest/#name>
[6] http://www.oasis-open.org/specs/a502.htm

3.2 tgroup

The `tgroup` tag marks the beginning of a new portion of a table (p. 2). 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. 3) or spanspec (p. 3), zero or one thead (p. 3) or tfoot (p. 3), and one tbody (p. 3) tag.

3.3 colspec

The `colspec` tag is an **empty tag** that specificies the column of a table (p. 2) or entrytbl (p. 3). The names and numbers specified as attributes are used for referencing by other tags.

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. 3) or entrytbl (p. 3) in different rows (p. 3).

3.5 thead

The `thead` tag identifies the heading row (p. 3) of a tgroup (p. 3) or entrytbl (p. 3). The `thead` tag can have zero, one, or more colspec (p. 3) tags and one or more row (p. 3).

3.6 tfoot

The `tfoot` tag identifies the rows (p. 3) of footer information that are displayed after the tbody (p. 3). The `tfoot` tag can have zero, one, or more colspec (p. 3) tags and one or more row (p. 3).

3.7 tbody

The `tbody` tag identifies the body of a tgroup (p. 3) or entrytbl (p. 3). The `tbody` tag must have one or more row (p. 3) tags.

3.8 row

The `row` tag identifies the row of information in a thead (p. 3), tbody (p. 3), or tfoot (p. 3). The `row` tag must have one or more entry (p. 3) or entrytbl (p. 3).

3.9 entrytbl

The `entrytbl` tag takes the place of an entry (p. 3), but fits into a single row (p. 3) of tbody (p. 3) in a tgroup (p. 3). The content model is the same as that of a tgroup (p. 3) except that tfoot (p. 3) is ommitted and `entrytbl` is self-excluding. Its children tags are zero, one, or more colspec (p. 3) or spanspec (p. 3), zero or one thead (p. 3) or tfoot (p. 3), and one tbody (p. 3) tag.

3.10 entry

The `entry` tag identifies an entry in a row (p. 3). The `entry` tag contains ASCII text and zero, one, or many cite[7], term[8], cnxn, link, code, emphasis[9], or media[10].

[7] "The Basic CNXML": Section Cite <http://cnx.org/content/m9000/latest/#cite>
[8] "The Basic CNXML": Section Term <http://cnx.org/content/m9000/latest/#term>
[9] "The Basic CNXML": Section Emphasis <http://cnx.org/content/m9000/latest/#emphasis>
[10] "The Intermediate CNXML" <http://cnx.org/content/m9006/latest/#media>

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. For more information, consult the CALS Table Spec[11] or the CNXML Spec[12] .

Example 3

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

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

Steak Cooking Temperatures

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

Table 1

4 Conclusions

This concludes the CNXML tutorial. Two useful resources are The CNXML Language Spec[13] and the example document[14] we've constructed over the three part course in CNXML.

[13] http://cnx.rice.edu/technology/cnxml/0.5/spec
[14] "Grilling the Best Steak" <http://cnx.org/content/m10281/latest/>