

CHEMML

ConTEXT XML

Pragma ADE / Hasselt NL

Description

The chemical XML markup dialect provided by CONTEX_T is relatively simple, but sufficient for everyday chemistry. Of course we have elements that represent the building blocks of chemistry:

- atoms
- ions
- molecules

In addition there are methods for:

- reactions
- bonds

Reactions, molecules and ions can be annotated. For this we provide:

- captions

For the moment there are no provisions for structure formulas, but there will be some day (probably build on top of the CONTEX_T module PPCHTEX).

Beware: the spacing may be improved and therefore can change slightly in future versions.

Structure

The simplest building block is the atom:

```
<chem>
  <atom> H </atom>
</chem>
```

H

You can dress up an atom with a couple of ornaments:

```
<chem>
  <atom n="6" charge="-5" weight="652" protons="339"> Rx </atom>
</chem>
```

$${}_{339}^{652}\text{Rx}_6^{-5}$$

You (or nature) can combine atoms in molecules:

```
<chem> <molecule>
  <atom n="2"> H </atom>
  <atom>      O </atom>
</molecule> </chem>
```



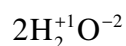
An efficient way of identifying multiple molecules is the following:

```
<chem> <molecule n="2">
  <atom n="2"> H </atom>
  <atom>      O </atom>
</molecule> </chem>
```



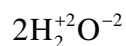
As you can see, a molecule is just a series of atoms and again, we can add a couple of ornaments.

```
<chem> <molecule n="2">
  <atom n="2" charge="+1"> H </atom>
  <atom      charge="-2"> O </atom>
</molecule> </chem>
```



Atoms with a charge are called ions. Because a charge can be associated to a combination of atoms, you can also package the atoms in an ion element.

```
<chem> <molecule n="2">
  <ion charge="+2"> <atom n="2"> H </atom> </ion>
  <ion charge="-2"> <atom>      O </atom> </ion>
</molecule> </chem>
```



You can influence the appearance of an ion. Take the following acid:

```
<chem> <ion charge="-1">
  <atom> C </atom>
  <atom> O </atom>
  <atom> O </atom>
  <atom> H </atom>
</ion> </chem>
```



A different way to visualize the charge is to use brackets around an ion. You can force this with a directive.

```

<chem>
  <?context-chemml-directive ion alternative b ?>
  <molecule>
    <ion charge="-1">
      <atom> C </atom>
      <atom> O </atom>
      <atom> O </atom>
      <atom> H </atom>
    </ion>
  </molecule>
</chem>

```



As you can see in the next example, the negative charge of the whole end up above the number of Y atoms.

```

<chem> <molecule>
  <ion charge="-1">
    <atom> X </atom>
    <atom n="3"> Y </atom>
  </ion>
</molecule> </chem>

```



A molecule seldom comes alone. This is why we have an encapsulating element dealing with reactions. This is a presentational element, which means that we just have a sequence of items to be represented.

```

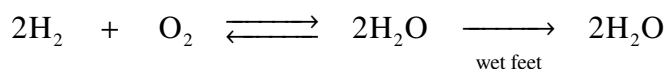
<formula>
  <chem>
    <reaction>
      <molecule n="2">
        <atom n="2"> H </atom>
      </molecule>
    <plus/>
    <molecule>
      <atom n="2"> O </atom>
    </molecule>
  <equilibrium/>
  <molecule n="2">
    <atom n="2"> H </atom>
    <atom> O </atom>
  </molecule>

```

```

<gives>
  <caption> wet feet </caption>
</gives>
  <molecule n="2">
    <atom n="2"> H </atom>
    <atom>      O </atom>
  </molecule>
</reaction>
</chem>
</formula>

```



The following special symbols are available:

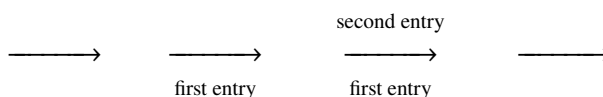
| element | representation |
|-------------|----------------|
| plus | + |
| minus | - |
| equal | = |
| gives | → |
| equilibrium | ⇌ |
| mesomeric | ↔ |

You can put some text (often expressing conditions) on top or below the arrows. In that case you need to add a couple or more elements and non empty alternatives of the arrow elements.

```

<formula>
  <chem>
    <reaction>
      <gives>
      </gives>
      <gives>
        <caption>first entry</caption>
      </gives>
      <gives>
        <caption>first entry</caption>
        <caption>second entry</caption>
      </gives>
      <gives/>
    </reaction>
  </chem>
</formula>

```

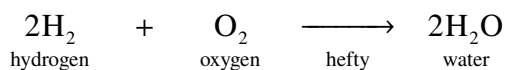


You can also add captions to molecules and ions. If you want a caption to an atom, you need to encapsulate it in a molecule.

```

<chem>
  <reaction>
    <molecule n="2">
      <atom n="2"> H </atom>
      <caption> hydrogen </caption>
    </molecule>
  <plus/>
  <molecule>
    <atom n="2"> O </atom>
    <caption> oxygen </caption>
  </molecule>
  <gives>
    <caption> hefty </caption>
  </gives>
  <molecule n="2">
    <atom n="2"> H </atom>
    <atom> O </atom>
    <caption> water </caption>
  </molecule>
  </reaction>
</chem>

```

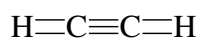


Another way to dress up a chemical formula is to visualize bonds. Therefore we provide a simple bond element.

```

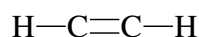
<chem> <molecule>
  <atom> H </atom>
  <bond n="1"/>
  <atom> C </atom>
  <bond n="2"/>
  <atom> C </atom>
  <bond n="1"/>
  <atom> H </atom>
</molecule> </chem>

```



You can also use a more verbose series of commands to achieve this visualization.

```
<chem> <molecule>
  <atom> H </atom>
  <singlebond/>
  <atom> C </atom>
  <doublebond/>
  <atom> C </atom>
  <singlebond/>
  <atom> H </atom>
</molecule> </chem>
```



Normally the typesetting engine will adapt itself automatically to either in-line or display chemistry. The next example shows in-line usage.

When we manage to bring `<chem> <atom> H </atom> </chem>` (hydrogen) in contact with `<chem> <atom> O </atom> </chem>` (oxygen) we get a bit of noise and fire but we end up with `<chem> <molecule> <atom n="2"> H </atom> <atom> O </atom> </molecule> </chem>` (water).

When we manage to bring H (hydrogen) in contact with O (oxygen) we get a bit of noise and fire but we end up with H₂O (water).

However, if you want to have a bit more control, you should use one of the following capsules. The normal `<chem>` element acts chooses one of these, depending on the circumstances.

| element | meaning |
|---------|-------------------|
| ichem | in-line Chemistry |
| dchem | display Chemistry |

Usage

This module is loaded as any module:

```
\usemodule[chemml]
```

XML example

Here we will collect a few examples.

Configuring

We will add some more control over spacing, lettering and color later.

Documentation

There is no additional documentation. The $\text{T}_{\text{E}}\text{X}$ counterpart of this module, namely $\text{PPCH}_{\text{T}_{\text{E}}\text{X}}$, uses a representation more in tune with the way one codes documents in $\text{T}_{\text{E}}\text{X}$.

Colofon

This manual is part of the $\text{CON}_{\text{T}_{\text{E}}\text{X}}\text{T}$ distribution, and is authored and maintained by Hans Hagen. $\text{CON}_{\text{T}_{\text{E}}\text{X}}\text{T}$ is developed at PRAGMA ADE, Hasselt, The Netherlands. This manual is produced on October 26, 2001.