

Welcome to the Atlas 4 Windows Project

Requirements

To successfully use this project, you need to have Java installed. Download and install the latest version of Java JRE from <http://java.sun.com>

As of this writing, jre-1_5_0_01-windows-i586-p.exe is the file you are looking for.

Install Java

1. Download JRE 5.0 from <http://java.sun.com> (JRE 5.0 Update 1, also known as JRE 1.5.0)
2. Install program and accept all default options.
3. Ensure that the JAVA_HOME environment variable was set right. To do so, open a command prompt and type: set. You should see the variable JAVA_HOME pointing to the newly created folder.

Installation

Unzip the content of the atlas4w.zip into a directory of your choice. For example purposes, we assumed that the atlas framework was decompressed in c:\atlas4w.

To try it out, go to the install directory:

```
» c:
» cd \atlas4w
```

Then, generate the atlas from the modules:

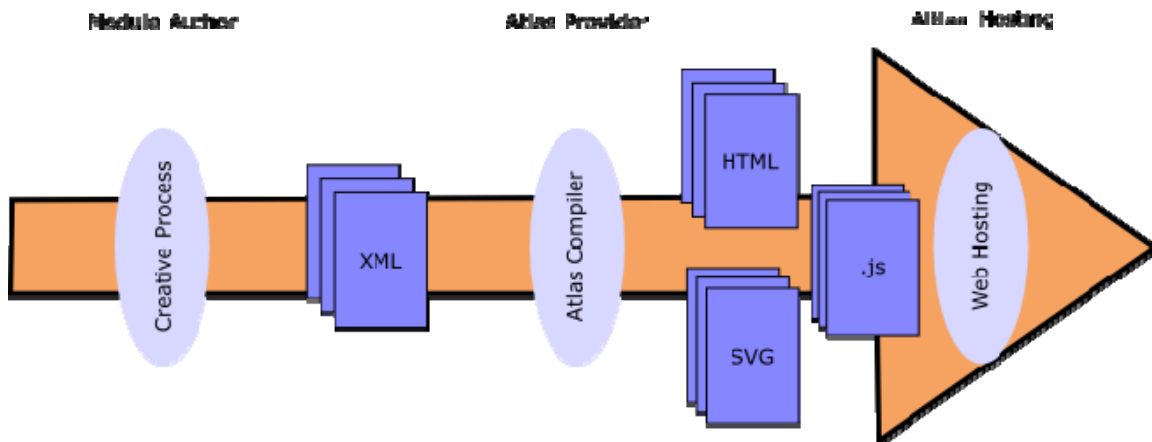
```
» gen
```

Finally, look at the resulting atlas. To do so, start Microsoft Internet Explorer and open the following document:

```
file:///C:/atlas4w/index.html
```

What is a module?

A module is a XML document, where the mark up language indicates to the atlas compiler what to display. Once an atlas is compiled, it can be deployed on a web server. The following picture depicts the process from module creation to atlas deployment.



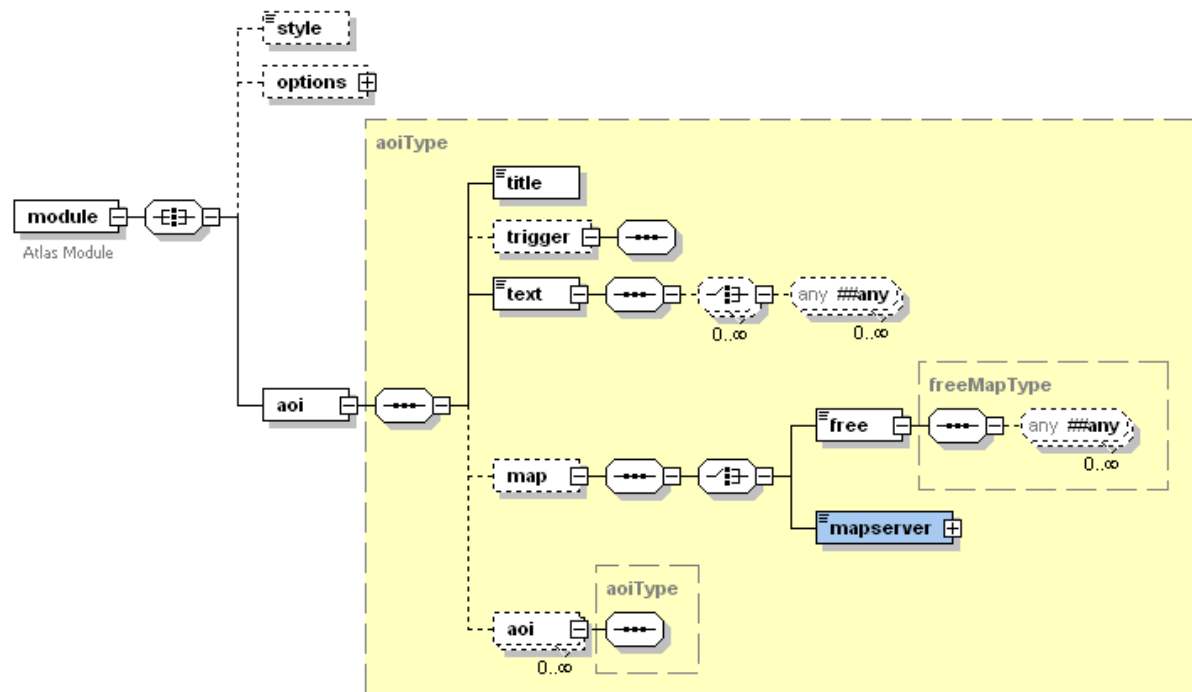
The previous figure shows the three perspectives:

- **Module Author:** The module author is left with the responsibility of creating XML documents that describes an atlas module. This document contains all the text that the author wants to publish. It also specifies the maps that should be displayed. And finally, it contains mark up elements that links features on the map to information found in the text.
- **Atlas Provider:** Responsible for transforming the work of the authors into a set of packages that can be published on the web. The atlas provider makes decision about the organization of the pages, the look-and-feel of the pages, the interaction level between the map and the text. The atlas provider produces a set of web files, including HTML, SVG, Javascript and Servlets that can be deployed on a web site.
- **Atlas Hosting:** In our model, this is the last step. The activity of atlas hosting entails to make available all of the atlas provider packages with all the supporting applications and databases.

Atlas4w is a project to help with the first perspective: that of the module author. The project strives to take a module written by an author and create an atlas-like web page representing it. This way, the author is provided with early feedback as his/her progress with module development.

Atlas Mark Up

The complete rules for the module XML mark up language can be found in a XML Schema (a.k.a. XSD), which is located in c:\atlas4w\schema\module.xsd



As a practical example, this is what a typical XML module file would look like:

```
<?xml version="1.0" encoding="UTF-8"?>

<mod:module
  xmlns:mod="http://www.gcrc.carleton.ca/Atlas/Module/1.0"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.gcrc.carleton.ca/Atlas/Module/1.0
  ../schema/module.xsd"
>
  <mod:options>
    <mod:model>
      <mod:option type="selectReplacesFocus" value="1"/>
      <mod:option type="disableUnfocus" value="0"/>
      <mod:option type="selectDisablesFocus" value="0"/>
    </mod:model>
  </mod:options>

  <!-- This is the main Area of Interest (AOI) -->
  <mod:aoi>
    <mod:title>Example Module</mod:title>
    <mod:text>
      <b>Welcome to the Example Module</b>
      <p>
        In this module, we would like to go over a number of topics:
        <ul>
          <li><mod:link layer="topLayer"
feature="glacier">Glaciers</mod:link></li>
          <li><mod:link layer="topLayer" feature="globalWarming">Global
Warming</mod:link></li>
```

```

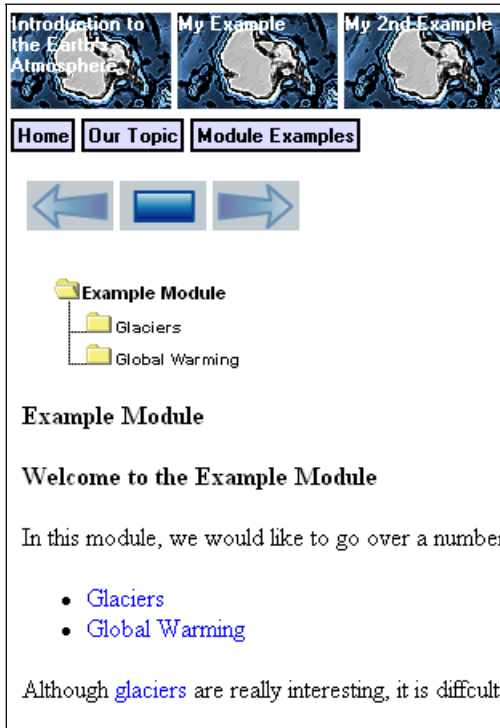
        </ul>
      </p>
    <p>
      Although <mod:link layer="topLayer"
feature="glacier">glaciers</mod:link> are really interesting,
      it is difficult to discuss them without explaining the impacts of
      <mod:link layer="topLayer" feature="globalWarming">global
warming</mod:link>.
    </p>
  </mod:text>

  <mod:aoi id="glaciers">
    <mod:title>Glaciers</mod:title>
    <mod:trigger layer="topLayer" feature="glacier"/>
    <mod:text>
      <h1>Glaciers</h1>
      <p>There are many types of glaciers.</p>
    </mod:text>
  </mod:aoi>

  <mod:aoi id="global">
    <mod:title>Global Warming</mod:title>
    <mod:trigger layer="topLayer" feature="globalWarming"/>
    <mod:text>
      <h1>Global Warming -- What is it?</h1>
      <p>Some explanations of global warming.</p>
    </mod:text>
  </mod:aoi>
</mod:aoi>
</mod:module>

```

The XML content displayed above can be found in the file
c:\atlas4w\modules\example.xml. The following picture shows the same module once
compiled by the atlas framework.



Areas of Interest

The first step to writing an atlas module is to divide the content into a number of areas of interest (AOI). An area of interest represents one topic that logically fits into a page of the atlas. Remember that you are dealing with an on-line atlas, and not a book. Therefore, dividing your content into portions that are large enough to be informative and yet small enough to fit a web browser will be a great challenge.

Areas of interest can be inserted within one another to form a hierarchy. Therefore, you should aim to obtain a logical flow and an intuitive breakdown for your content. Birgit Woods has used the following organization for her module on the Dry Valley:

- Main AOI
 - Glaciers
 - Outlet Glaciers
 - Canada Glacier
 - Commonwealth Glacier
 - Taylor Glacier
 - Piedmont Glaciers
 - Wilson Glacier
 - Alpine Glaciers
 - Ferrar Glacier
 - Webb Glacier
 - Upper Wright Glacier
 - Lakes
 - Lake Bonney
 - Lake Frixell
 - Lake Vida
 - Lake Wright

This outline is offered as an example to illustrate a logical hierarchical break down of a module.

Within an area of interest, one attribute and two elements are compulsory:

1. **Identifier (id):** This is a programming requirement. You have to ensure that each AOI is assigned a unique id.
2. **Title:** This is the title of the area of interest. It is important to choose an evocative name since this is the name that is used throughout the module navigation system to get to this AOI.
3. **Text:** This is the actual text that is to be displayed in the module page that relates to this AOI. Within a text element, it is possible to insert HTML tags for

formatting the text. It is also possible to use hypertext links and image tags. Also, some module text tags are provided.

The following excerpt demonstrates the use of title and text within an AOI:

```
<mod:aoi id="glaciers">
  <mod:title>Glaciers</mod:title>
  <mod:text>
    <h1>Glaciers</h1>
    <p>There are many types of glaciers.</p>
  </mod:text>
</mod:aoi>
```

AOIs within AOI

An area of interest can be added to another on by inserting an `<aoi>` tag within another. In the first example, you can notice that the tags forming the beginning and the end of the AOI for “Global Warming” are fully within the ones for the main AOI.

Triggers

Within an `<aoi>` tag, it is possible to specify one or many triggers that are responsible to bringing the area of interest to the user’s attention. To do so, the author must define a `<trigger>` tag within the `<aoi>` tag identifying which condition triggers the user’s attention to this AOI.

```
<mod:aoi id="glaciers">
  <mod:title>Glaciers</mod:title>
  <mod:trigger layer="topLayer" feature="glacier"/>
  <mod:text>
    <h1>Glaciers</h1>
    <p>There are many types of glaciers.</p>
  </mod:text>
</mod:aoi>
```

In the example above, the “Glaciers” AOI will be shown to the user whenever the feature “glacier” from the layer “topLayer” is selected.

So far, defining layers and features is completely arbitrary. However, keep in mind that as the atlas framework evolves, these identifiers may be linked to the data shown in the map. For now, you can make up those names as you wish.

Links

When triggers are defined for AOIs, links can be used to jump to those AOIs from the text. Links are also used to establish a relationship between the text and the map. However, since maps are not implemented with the atlas4w framework yet, this feature can not be tested.

Links must be inserted within the text elements of an AOI.

```
<mod:text>
<b>Welcome to the Example Module</b>
<p>
In this module, we would like to go over a number of topics:
<ul>
  <li><mod:link layer="topLayer" feature="glacier">Glaciers</mod:link></li>
</ul>
</p>
```

```
</mod:text>
```

The example above illustrates how a link tag is inserted within a text element. Note that the text element contains HTML mark up (p, ul, li). Below is a picture that shows how such a link is rendered in the atlas.

In this module, we would like to go over a number of topics:

- [Glaciers](#)
- [Global Warming](#)

Although [glaciers](#) are really interesting, it is difficult to discuss th

At present, when the mouse is hovering over a link, all links associated with the same layer/feature pair are highlighted. Furthermore, all associated features on the map are also highlighted.

The same behaviour is observed when the mouse is hovering on the associated feature on the map.

When the link or the feature is clicked, the user's view is replaced with the selected area of interest. This is the relationship between <trigger> and <link> tags.

```
<mod:aoi>
  <mod:title>Example Module</mod:title>
  <mod:text>
    <b>Welcome to the Example Module</b>
  </mod:text>
  <mod:map>
    <mod:free>
      <embed src="../../../resource/territories/0007/territories.svg"
type="image/svg+xml" style="width:100%; height:100%"/>
    </mod:free>
  </mod:map>
</mod:aoi>
</mod:module>
```

Maps

To this point, maps in the atlas framework have been hand crafted. The aim is to have the atlas framework display maps that are generated by the web server from data located in a back-end database. Therefore, little effort is spent these days to support hand crafted maps.

However, if you wish to include an image of a map in your module, you can do so by inserting a <map> tag following the text tag.

Within the <map> tag, you must insert a <free> tag (meaning freehand). Within the <free> tag, you can use any HTML to import a map. For example, to import an image:

- ``