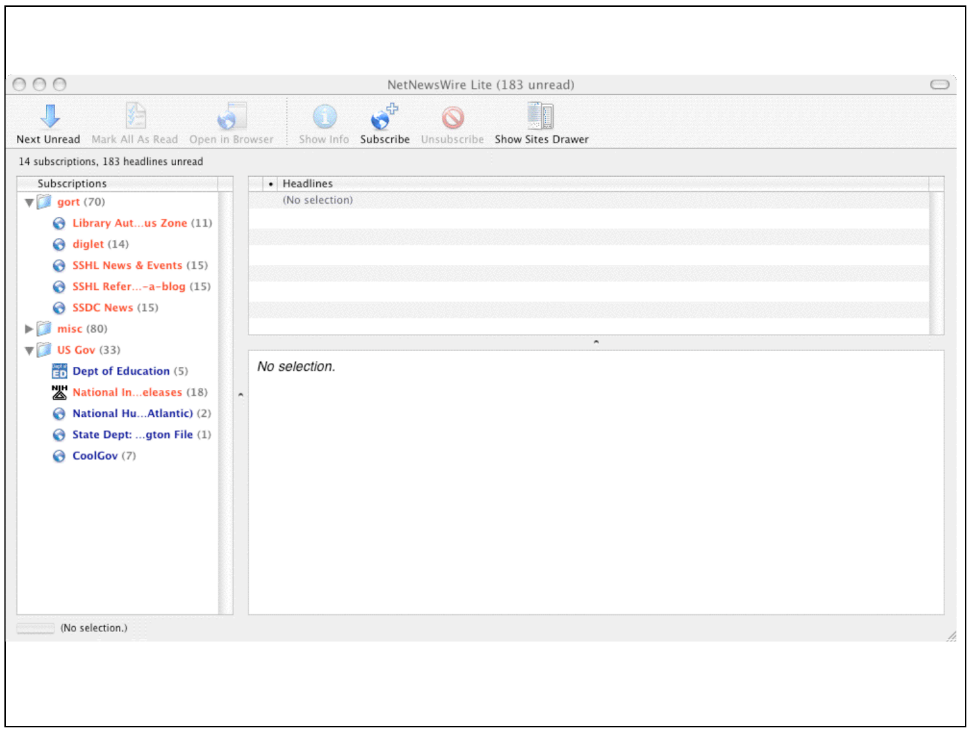
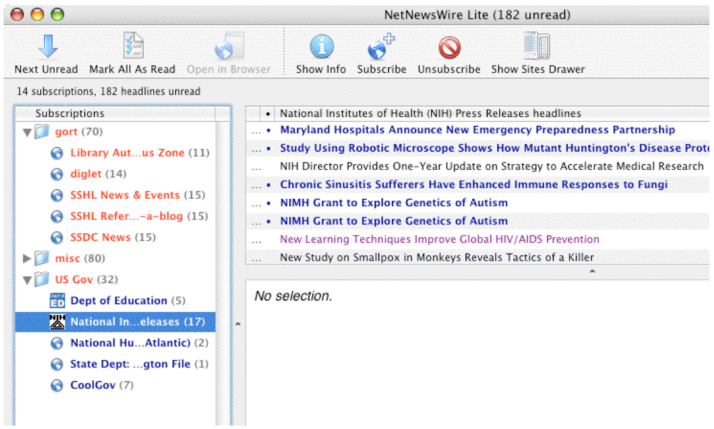
RSS, etc.

James A. Jacobs
Data Services Librarian
University of California San Diego
jajacobs@ucsd.edu



Here is a typical RSS-reader (this one is NetNewsWire Lite for the Mac).

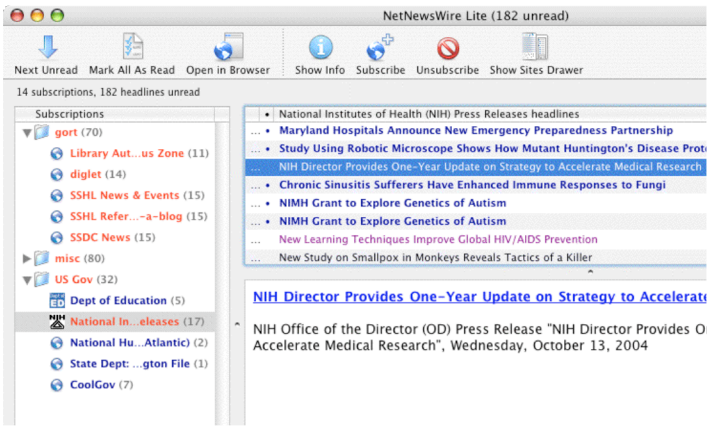
Feeds or "subscriptions" are listed in the left pane. These are items I've chosen and set up to read regularly.



Highlighted is a feed from the national institutes of health.

In the second pane, at the top right, are headlines from this feed.

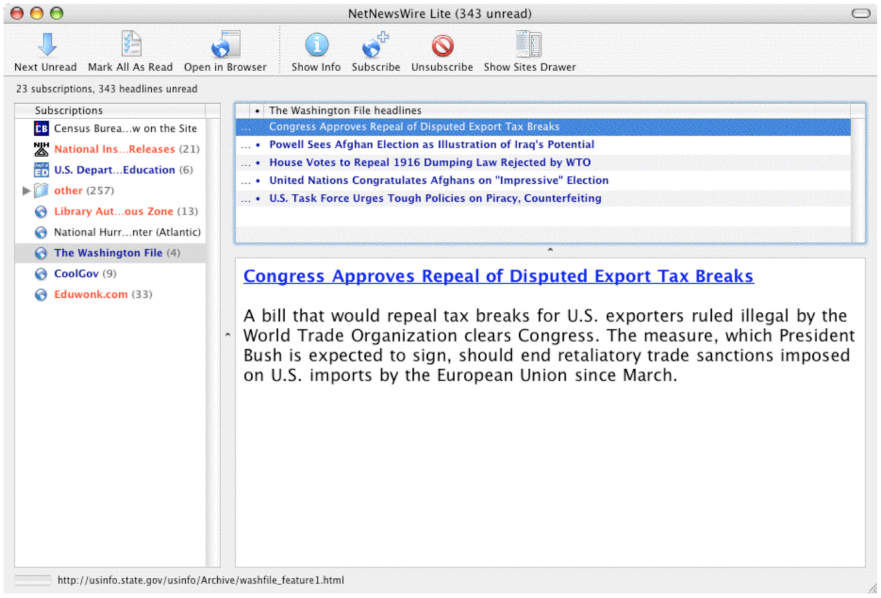
Note the different colors of headlines indicating whether I've viewed them here in the newsreader or the complete item in my web browser.



Here, a headline is highlighted and a summary of the headline appears in the 3rd pane, bottom right of the rss reader window.

Note url of the item at bottom of window. Clicking on the headline opens the complete article in a browser window

Here's how the web page with the same headline we just saw appears

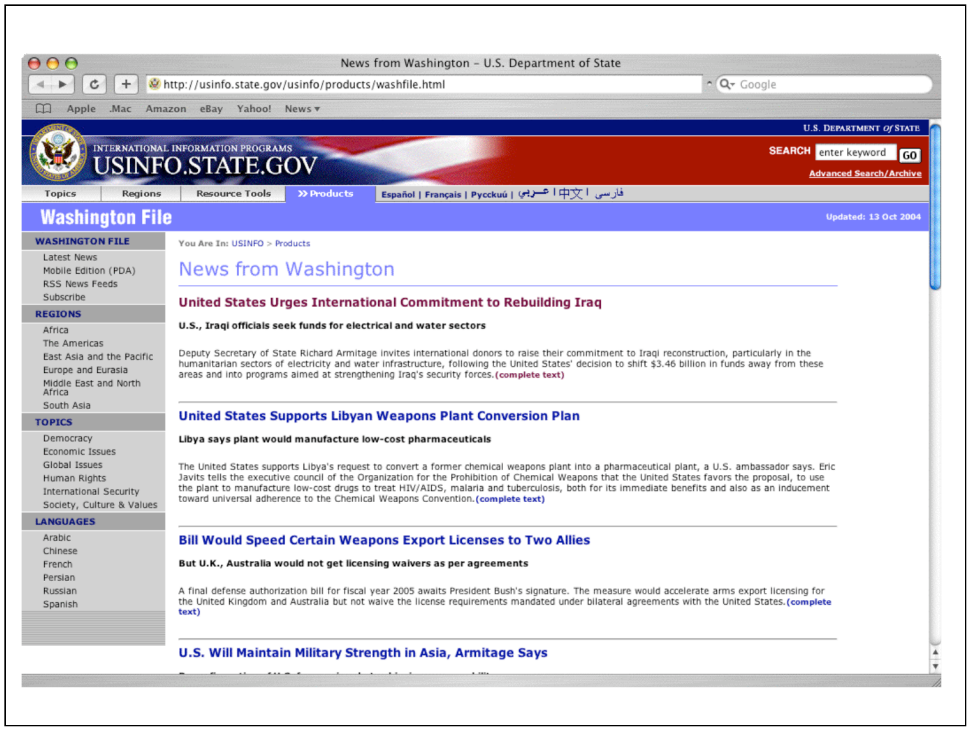


Another example. This from a state dept. rss feed “washington file.”

Note the presentation in the newsreader makes it easy to browse lots of information quickly!

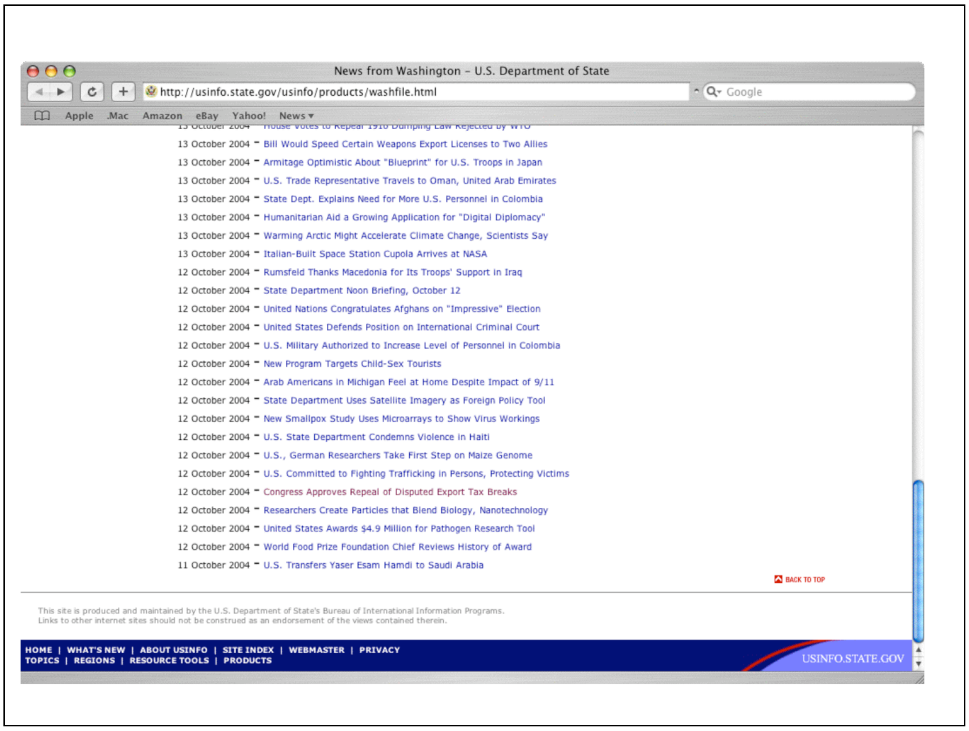
This item is presented in exactly the same way as the item from NIH.

Very different way of viewing news from browsing web pages with their very different presentations!

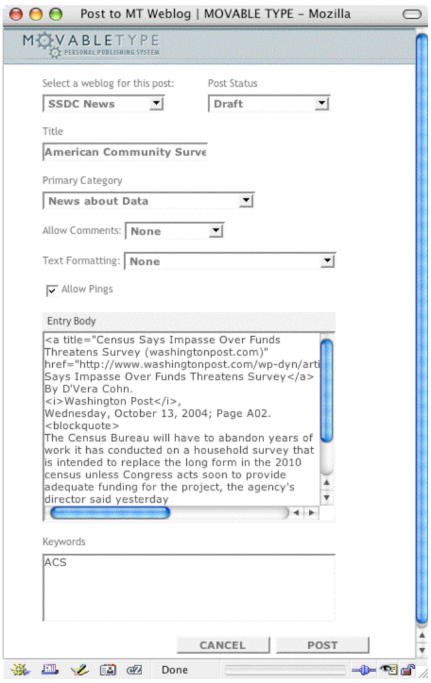


You can see this by looking at the state dept. web page and compare it to the rss-reader view.

Here is the top of a long page…



Here, near the bottom is the headline we saw in our rss reader.



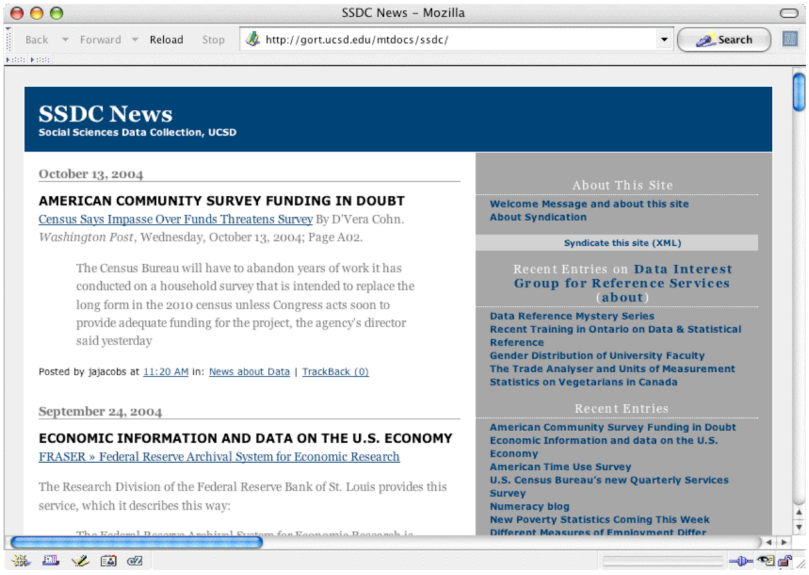
RSS – THE PRODUCER VIEW

“blogging software” allows you to do a lot quickly.

This is a screen shot of a typical form provided by the Movable Type blogging software.

The blogger simply fills in the form, concentrating on describing a resource, not a technical details or formatting issues.

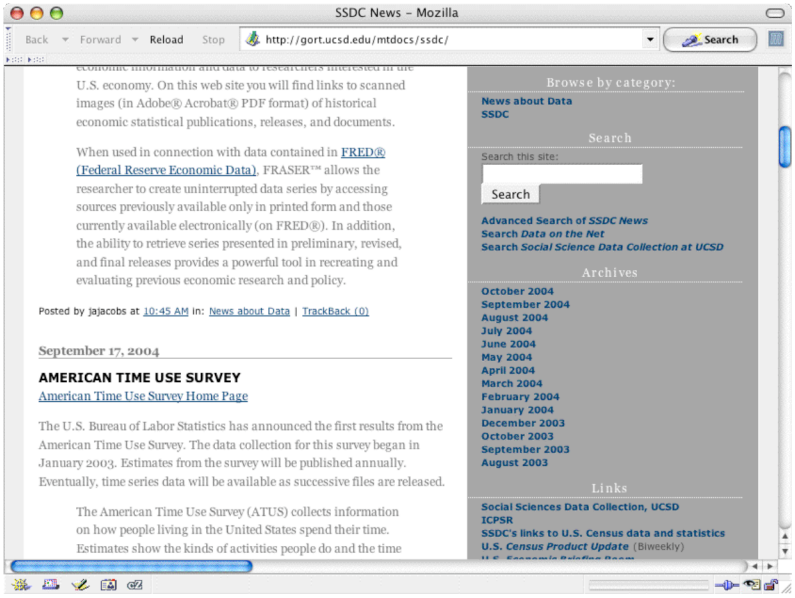
When done, you just click “Post” button at the bottom of the form.



Immediately (like 2 or 3 seconds!) several things happen.

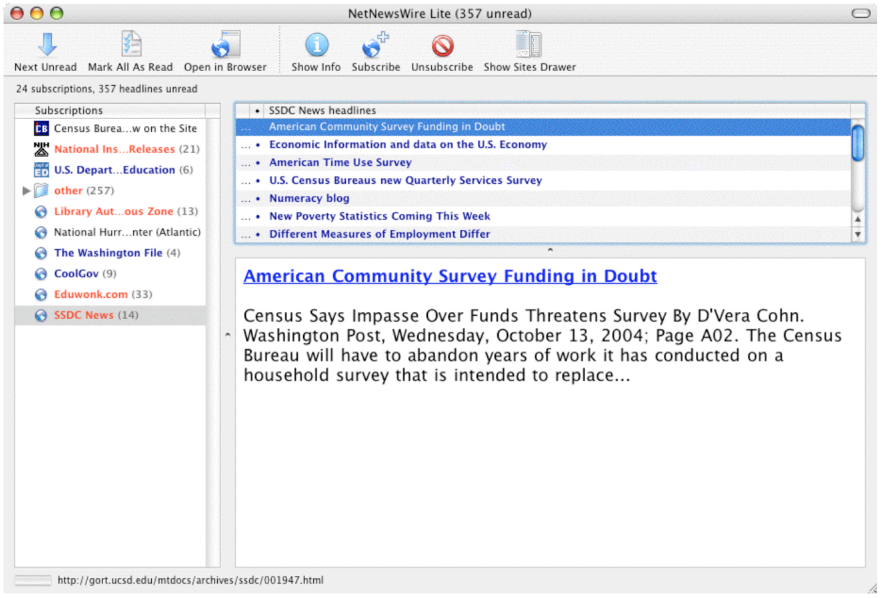
Html is created and your blog is updated.

Note here that the item I just described in the form appears at the top of my blog's home page.



In addition to the main page, the “archives” pages are updated for browsing by month.

And an index is updated so the item you just created is immediately findable through searching.



And, of course, your RSS feed is updated and users see your posting in their rss readers.

RSS -- The Technical View

RSS is just an XML file on a web server.

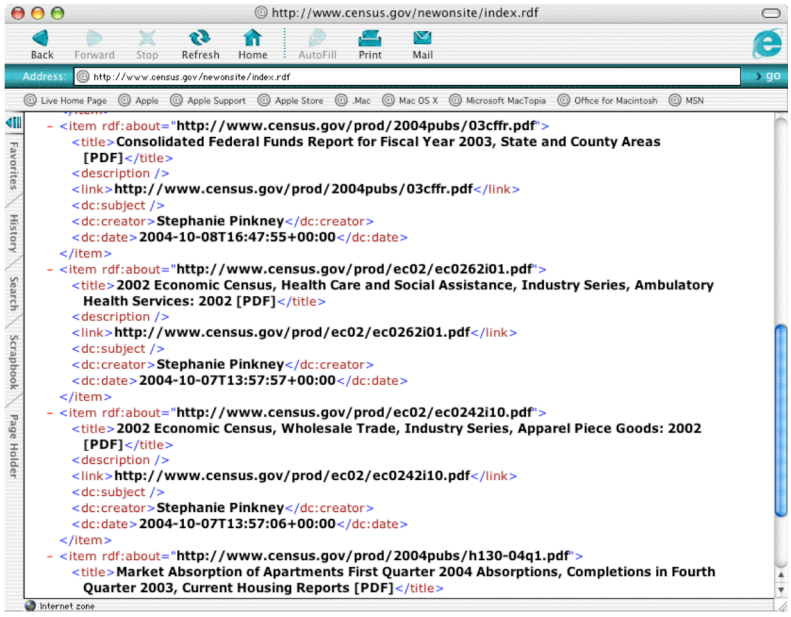
Now for the technical view.

This will be short…. Don't panic.

Rss is just an xml file on a web server.

There, we're done!

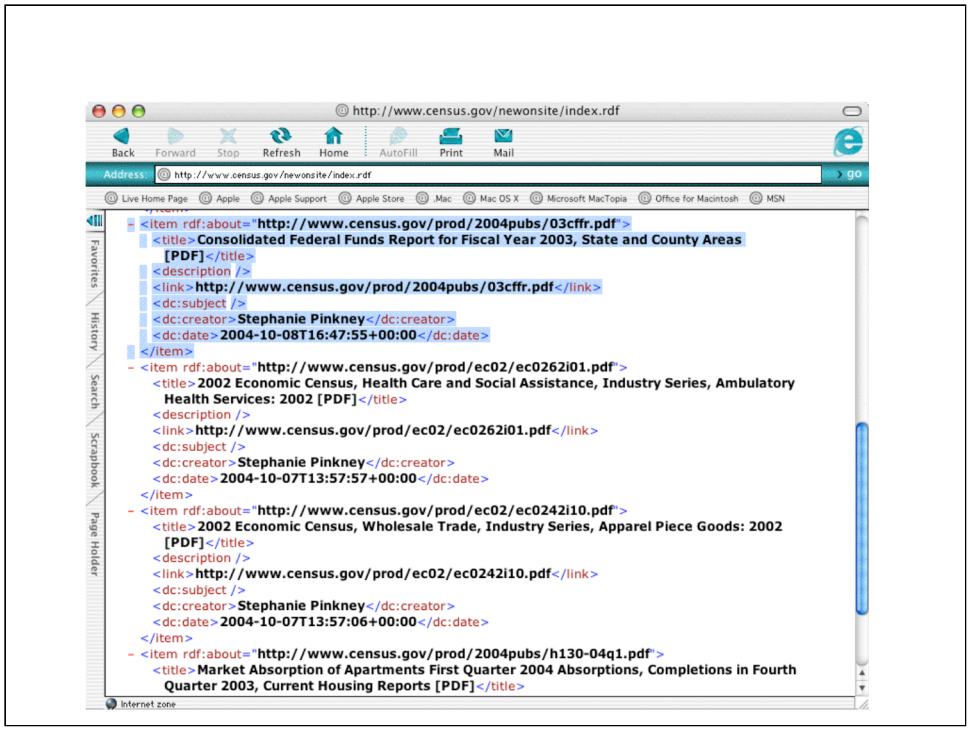
http://www.census.gov/newonsite/index.rdf



This is a screenshot of IT viewing an RSS feed from census bureau. Note the url at the top.

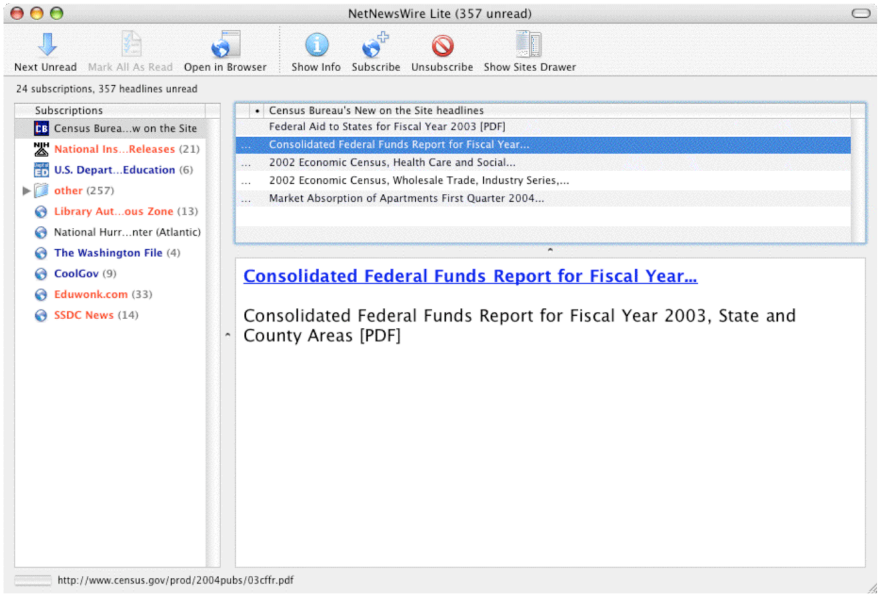
IE displays this file using its XML viewer so you can actually see the structure of the XML.

If you pointed IE at any xml file on a web server, IE would do the same thing: display the structure of the xml.



Highlighted here in blue is one entry, one headline.
“consolidated federal funds report…” pdf

NOTE the “dc:subject” and “dc:creator” tags. These are Dublin Core.



Point an RSS reader at the same file and the RSS reader parses the XML into headlines and summaries and displays the information in the way you have configured it to display these.

Here is how the rss reader presents that same file and headline.

HTML / XML

< >stuff</>

Now for a bit more technical stuff…

A quick review of the differences and similarities of HTML and XML

They are both markup languages. (ML = Markup Language)

We mark up our “stuff” with angle brackets.

HTML / XML

The Sun Also Rises

If we have, for instance a title…

HTML / XML

The Sun Also Rises

We can mark it with HTML by using to put it in italics. And our web browsers dutifully display the text in italics.

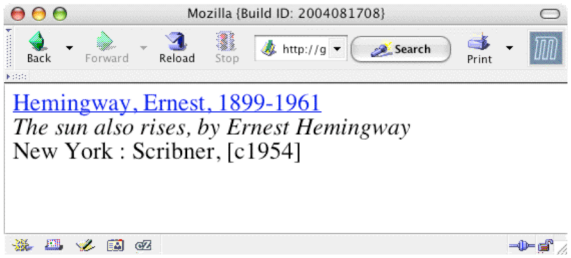
HTML / XML

The Sun Also Rises

<title>The Sun Also Rises</title>

We use XML tags to describe content rather than specify display properties.

HTML

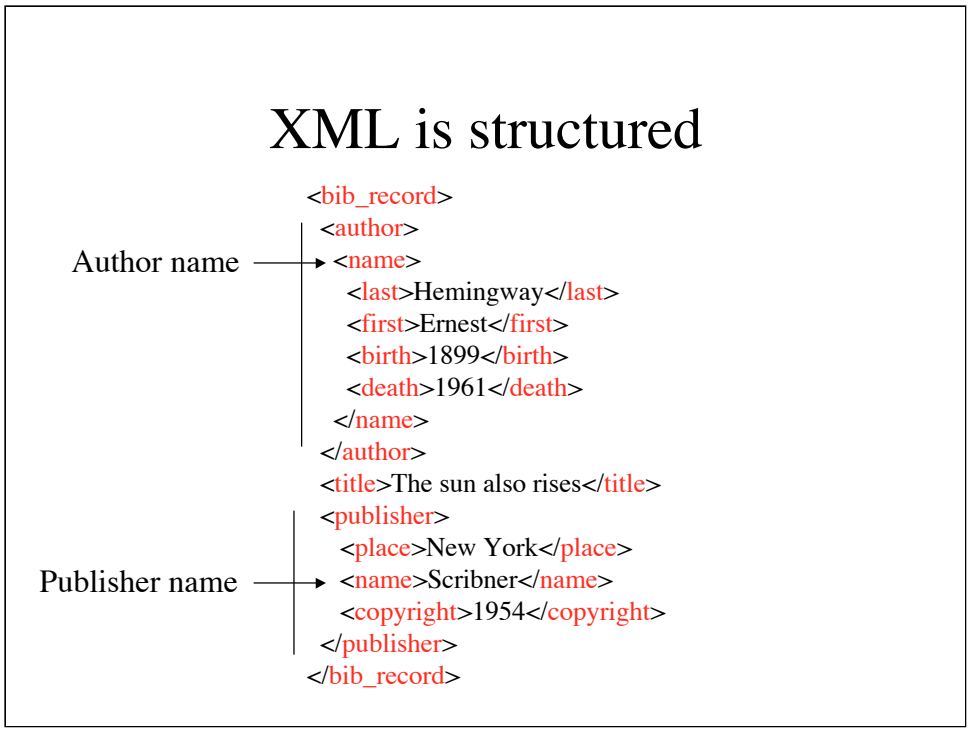



```
<a href="hemingway">Hemingway, Ernest, 1899-1961</a>
<br />
<i> The sun also rises, by Ernest Hemingway</i>
<br />
New York : Scribner, [c1954]
```

At the bottom of this slide is valid html.

Note that the tags tell us nothing about what this content is.

Note that there is text (the publisher information) that is not even inside any tag.



But xml is structured!

And hierarchical.

This example is not RSS, but, like RSS it is XML and serves as a simple easy to understand illustration of the hierarchical nature of XML.

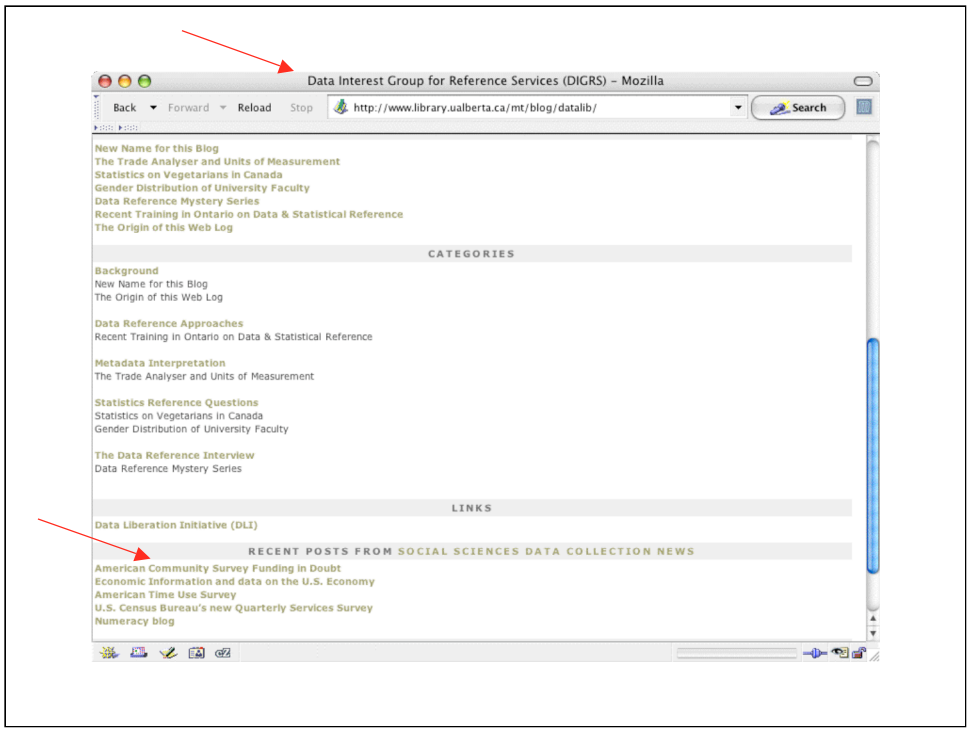
A tag “name” means one thing when it is inside an “author” tag and another inside a “publisher” tag.

Computers find this very easy to “parse”

In other words, machines (software) can “understand” xml and do more or less “intelligent” things with xml.

For example,, Give this file to a computer and and you could ask the computer for the birth date of the author of ‘the sun also rises” and the computer would be able to tell you.

Because it is easy to parse, it is easy to make use of XML in more than one way…



This means that It is easy to use and re-use xml

Here is an example of reusing and rss/xml "feed."

the blog entry we just saw appears on another blog, in canada, automagically! Xml makes this easy because it is easy to parse and re-use.

All this technical stuff (RSS is XML, XML is structured information) leads us to a couple of interesting points…

XML

- A lot of what we do is describe resources: in our OPACS, on our Web pages, in handouts, emails, databases, etc.
- XML is very well suited to describing resources in a structured way that enables using and reusing those descriptions.
- RDF = Resource Description Framework

Here are 3 points..

…

RDF is one very flexible way of describing resources and can be expressed in XML.

If we want to put most of our effort into describing resources and making it easy for users to make use of what we find,

And if we want to put less of our effort into designing web pages, writing and updating html, re-keying the same information in more than one place, etc..

Then one thing we might consider is using XML as a way to store information that describe resources.

Now… one more technical point…

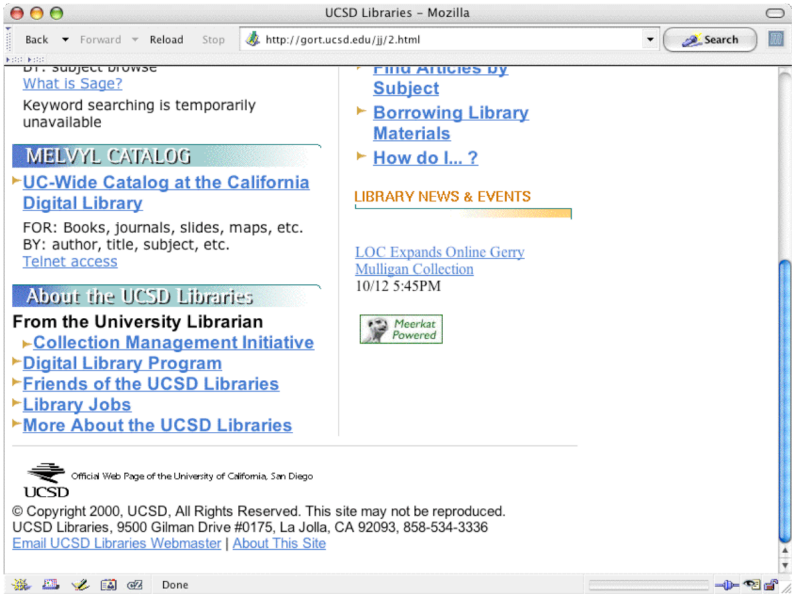
You don’t need a blog to create RSS.

You don’t need an RSS-reader to read RSS.

That’s right. Since RSS is just an XML file on a web server, all you need to create and “RSS FEED” is a way of creating XML and a web server.

And, since XML is easy for software to parse, the information can be used and reused in different ways, not just by news readers.

Here is what an actual, static library home page might look like with an rss feed adding dynamic content.



Here is a closeup of the feed from “meercat” displaying a headline from oct 12 on this page that I last edited in 2003.

This is an example of an RSS feed from LOC being re-used w/o an RSS reader. Here some simple javascript in an otherwise static HTML page inserts dynamic content (from RSS file at LOC) into the web page.

Create RSS/XML/RDF from any structured-data source

- OPAC / MARC records
- Database
- XML files
- OAI (Open Archives Initiative) Files
- Dublin Core
- Queries that return structured results

Now, if you start thinking this way, you might realize that any source of structured information can be exported to XML.

We already have lots of structured information and create new information with existing tools all the time.

XML is the markup language

There are many vocabularies and syntaxes for different purposes

- RSS
- RDF
- RDF / Dublin Core
- EAD
- MARC
- MODS
- ONIX
- PRISM
- DocBook
- PhilML

And RSS is only one of many XML formats.

Each of these is useful for a different kind of structured information or useful for describing a different kind of resource.

XML can be easily converted

- Many tools exist to read/convert XML. (Java, javascript, perl, PHP, etc.)
- XSL and XSLT were created explicitly for converting XML. With them XML can be converted to HTML, PDF, other XML, etc.
- XML is highly structured so it can be predictably converted.

And since xml can be easily converted or “transformed” …

One application can create XML and the XML can then be used and re-used in different applications for different purposes in different ways!

We can use and re-use our descriptions of resources.

Describe once, use over and over!

Imagine what GPO could do..

- Imagine GPO creating a single XML record for each and every document they catalog.

Imagine what GPO could do…

- An RSS feed for each FDLP library listing new resources
- An RSS feed for each agency
- An RSS feed for each Item number

So… imagine using our existing sources of structured information and using our existing tools to create XML from that structured information that is now locked up in applications (OPACs, databases) and in single-function formats (MARC, proprietary database formats)…

Imagine what we might do…

We could get those descriptions of resources and use and reuse them in many ways.

Imagine how this would empower FDLP libraries…

- Your library converting GPO's XML to:
  - A blog
  - Your own RSS feed
  - MARC/OPAC records
  - Static web pages
  - Records in databases like Infomine.

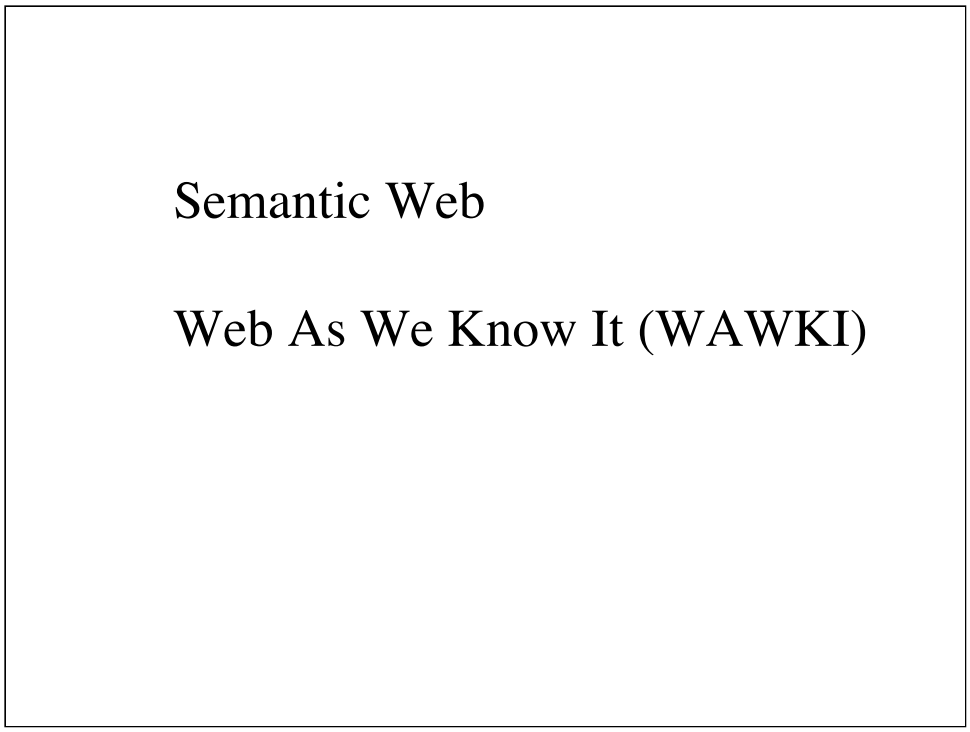
We could convert the gpo xml into our own local blog or blogs, one or more RSS feeds.

We could convert directly to MARC to load in our OPACs. We could create static web pages or dynamic web pages.

We could populate specialized databases.

And imagine, once we do this, we can also do other things. Imagine one or more rss feeds of new books generated from an OPAC. Not just government info, but gov. info mixed with other items in our OPAC.

Winding up now…



Two (possibly) new terms for you …

The Semantic Web is about defining data in a consistent, accurate way, so that it can be shared by machines and by humans.

The WAWKI is about moving human-friendly representations of resources from one place to another…

"Semantic Web site" …doesn't really exist [yet].

…[W]e're aiming to build a useful knowledge base of information about a specific domain, to publish that knowledge base on the Web, so that agents, both human and machine, can use the data in ways that aids them in accomplishing their goals and plans.

-- “Stuck in the Senate.” by Paul Ford, *XML.com*, October 13, 2004. http://www.xml.com/pub/a/2004/10/13/hackcongress.html

If you remember what the internet was like before HTML, you’ll remember usenet, ftp, and email that worked most of the time.

HTML revolutionized the internet and turned it into The Web.

In a similar way, XML will revolutionize and transform the web the way HTML revolutionized and transformed the internet.

We can be part of this by starting to use the tools (like blogs and rss) we have available to us now.