Content Syndication - Exemplar Documentation

Issue 1

28th March 2003

1)

Copyright

Table of Contents

1 Overview

1.1 Scope and Objectives

This document is consists of summary documentation and pointers to open source technology to enable external organisations to deploy their own aggregator/publisher components and participate in syndication.

It will consist of the following documents:

- Java HOW TO
- Perl HOW TO
- PHP HOW TO
- ASP HOW TO

The Exemplar documentation is designed to be a short (approx. 2 pages) document which provides an overview to sourcing and deploying RSS components.. The audience for the Exemplar documentation is IT managers who will be new adopters of it.

1.2 Exemplar Documentation

Short documents (approx. 2 pages) will be provided on sourcing and deploying RSS components using the following technologies:

- Java/JSP
- Perl
- ASP
- PHP

The document audience is IT managers, principally new adoptors of RSS content syndication, seeking to source and deploy RSS components. The aim is produce a useful resource and, if appropriate, hints for implementators of RSS content syndication.

Hints (i.e. low level technical detail such as configuration settings) should only be included if
a) these are not easily determined AND
b) external documentation does not appear to be readily available AND
c) they are verified by one of the trial participants

The documentation will supplement existing information, including URLs to external resources (e.g. community resources). "Under the radar" sources like early adopter weblogs and OSS projects will be referenced.

These documents will be similar in function to the HOWTO documentation used within the OSS community, although less technically rigorous SBS wants to save other organisations time in adopting an open syndication model by making these materials available.

Each Exemplar document will consist of the following sections:

- Introduction
- Specific Technology Overview (Publisher and Aggregator components)
- Further resources

1.2.1 Additional Assumptions for Exemplar Documentation

- The audience will be IT literate.
- Open Source Software options where applicable will be highlighted.
- The documents are not intended to act as a complete package for "selling" RSS. Business drivers for syndication will be familiar to the audience.
- SBS partners involved in the syndication trials will cooperate with SBS to provide information on their infrastructure and verify what is produced by SBS reflects and supplements their experience and existing documentation.
- SBS may include information compiled from other trial partners without independently verifying the components involved.

1.3 Use of Internet Resources

One of the major benefits of using the widely supported RSS standard is that many example programs and supporting library functions are available on The Internet. As a convenience, this document references a sample of these. All third party information is provided on an "AS IS" basis. Use of the Internet is at the syndication partner's risk and subject to all applicable laws. SBS has no responsibility for any information, software, services or other materials obtained using the Internet.

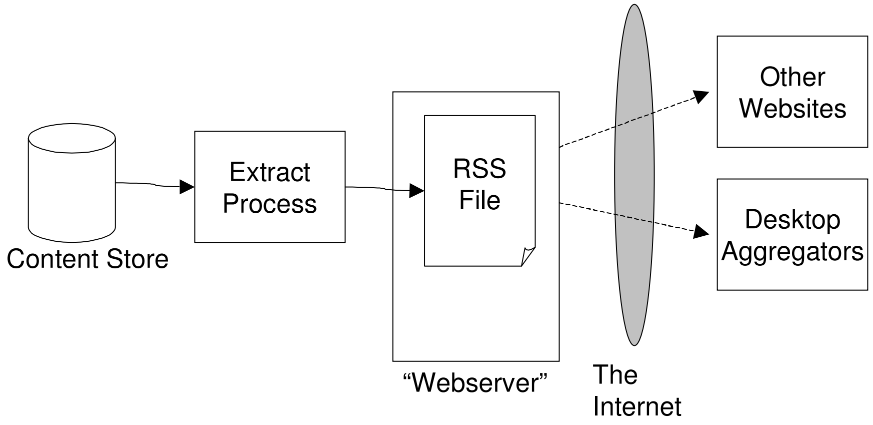
2 Overview

This document considers content syndication from two, complementary, perspectives.

- Outbound syndication where existing content on a website is made available for other sites to take.
- Inbound syndication where a site takes content from one or more other websites

2.1 Generic Outbound Syndication

Creating RSS Feeds allows other websites to take your content, either as summaries with links back to your site for the full content (RSS 0.91 [Ref.3], RSS 1.0 [Ref.4]), or as full articles (RSS 2.0 [Ref.5]). The diagram below illustrates the process of publishing (or "syndicating") RSS content onto the Internet.



The diagram above shows a 'generic' implementation consisting of:

Content Store This is where the existing content (e.g. stories, editorials, news etc) of the website is stored

Extract Process A process to extract content from the content store.

"Webserver" This is some form of web server (or web application server) technology

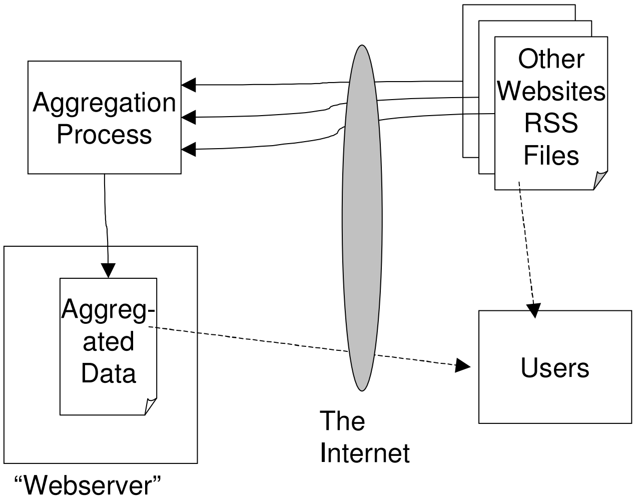
RSS An XML file of syndicatable content produced by the site in an RSS format

The following sections refine this with specific architecture components that could be used in several popular web architectures.

2.2 Generic Inbound Syndication

Inbound syndication is the converse of the above. The websites takes RSS data from the internet, processes it and re-displays it in summary and/or aggregated form. A typical example of this would be a site that scans three other sites for news headlines and then displays a selection of these in summary form. Typically, once a user clicks one of the summary links, the users browser is redirected to the originating site which provides the body of the story. This generates 'pull through' traffic for the originating site, so providing mutual benefit to both sites. (Note that RSS 2.0 allows syndication feeds to provide the body of the content, as well as the summary).

This is illustrated diagrammatically below:



Other Websites RSS Files These are RSS files published elsewhere on the Internet

Aggregation Process This process goes out to the Internet and pulls in one or more RSS Feeds.

Aggregated Data Typically this would be text containing a summary of each selected content item and a link to where the full story is stored on the internet. This may be stored in HTML, database or some other form depending on the technology used on the site.

"Webserver" Some form of website technology

Users Users browsing the aggregated feed.

3 Java HOW TO

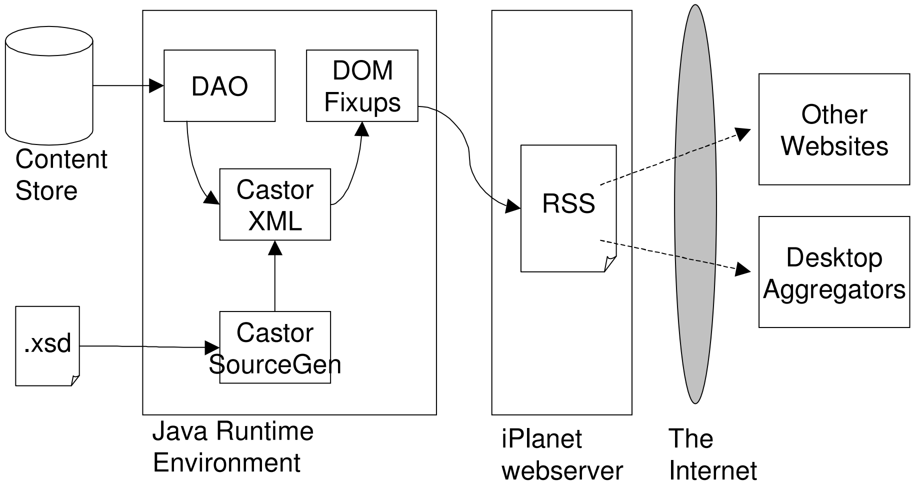
3.1 Outbound Syndication on a Java Based Website

There are several potential ways of achieving this. The method documented below is the current method used by the businesslink.org site and uses Open Source Software to marshall and unmarshall Java objects to and from XML.

In overview the process consists of the following steps:

- Read the data from the database via an appropriate Data Access Object
- Convert the internal object representation into XML using Castor XML
- Potentially add extra fields to the basic transformation using DOM

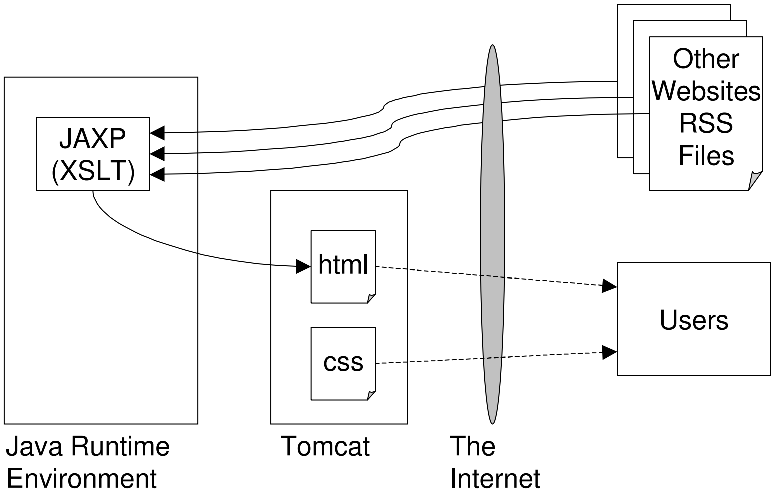
The following diagram illustrates these steps:



The Castor XML Parser and Source Generator are open source tools documented at http://castor.exolab.org/ which includes example code. (Note that Sun have now released a newer API for marshalling/unmarshalling Java objects to/from XML (JAXB). This is documented at http://java.sun.com/xml/jaxb/index.html. this is likely to be the preferred solution in future.)

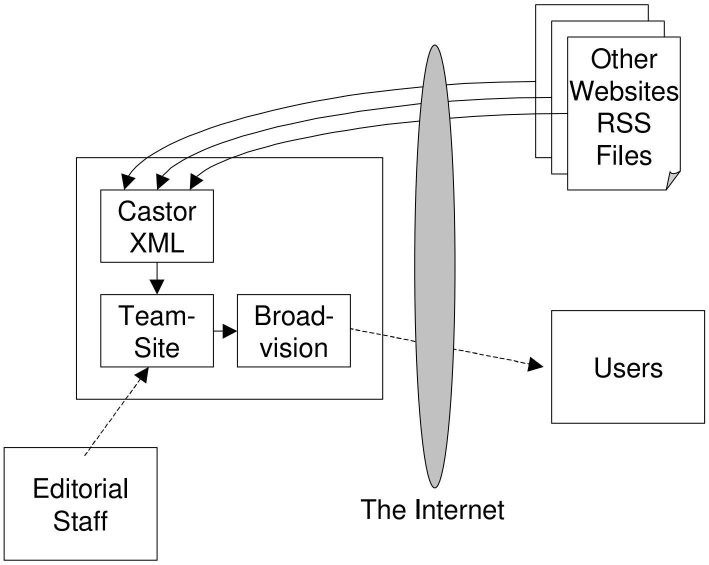
3.2 Inbound Syndication on a Java Based Website

A simple implementation is shown below. In this implementation an XSLT component is used to render other sites news headlines into a single aggregated HTML file. This is styled according to the sites default and included in certain pages when they are rendered.



The businesslink.org site implements a more complex implementation where incoming RSS items are read into the Teamsite content management system. Here the syndicated content may be edited, located on the website and then approved before being published to the live site. This is a more labour intensive implementation more suitable to formal syndication of content between sites.

This is shown diagrammatically below:

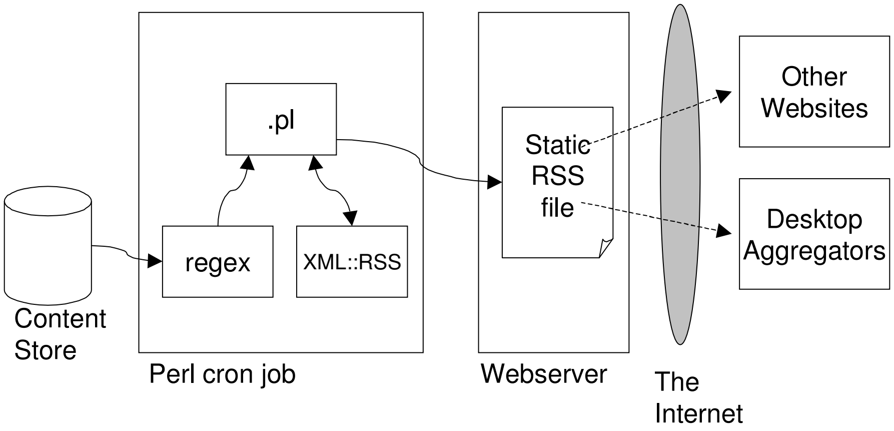


In this case the sites content management system (Teamsite) inserts the syndicated content into a workflow. This allows the content to be reviewed and possibly edited before deploying to the live Broadvision website. The specific details of this implementation are beyond the scope of the document, however it illustrates a more complex use of syndication including integration with a content management system.

4 Perl HOW TO

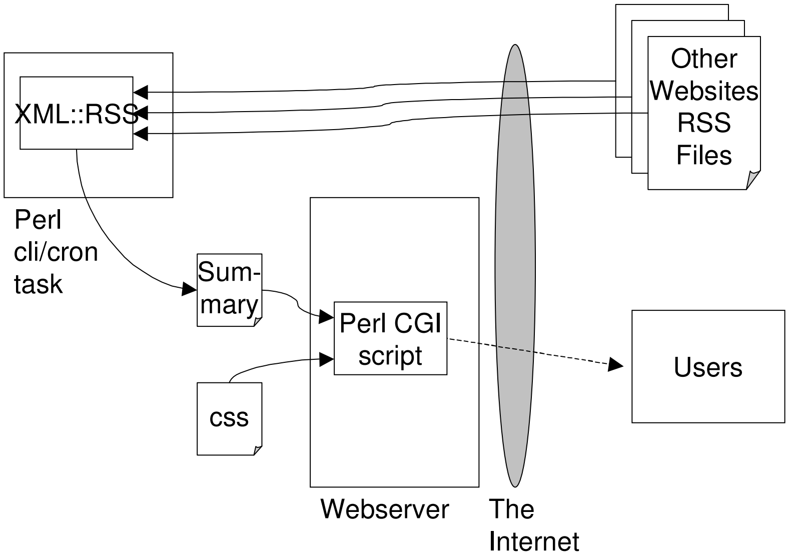
4.1 Outbound Syndication on a Perl Based Website

In this example a periodic task reads the content store (using regex in this case). A new RSS object is built with XML::RSS and populated, field by field from fixed site values and this content.



4.2 Inbound Syndication on a Perl Based Website

In this approach the other websites are read using the Perl get method, with the output parsed into an XML::RSS object.. Individual fields can then be extracted, merged, sorted and filtered. Finally the summary list can be rendered on the Website.



Perl Examples:

Channel editor program allowing retrieval, editing and replacement of RSS feeds (GNU)

http://www.webreference.com/perl/tools/

Displaying other sites RSS feeds on your website

http://www.webreference.com/perl/tutorial/8/index.html

General info from the same site

http://www.webreference.com/authoring/languages/xml/rss/

An example perl portal based on XML:RSS

http://www.perl.com/pub/a/2000/01/rss.html

XML:RSS info

http://search.cpan.org/author/KELLAN/XML-RSS-1.02/lib/RSS.pm#METHODS

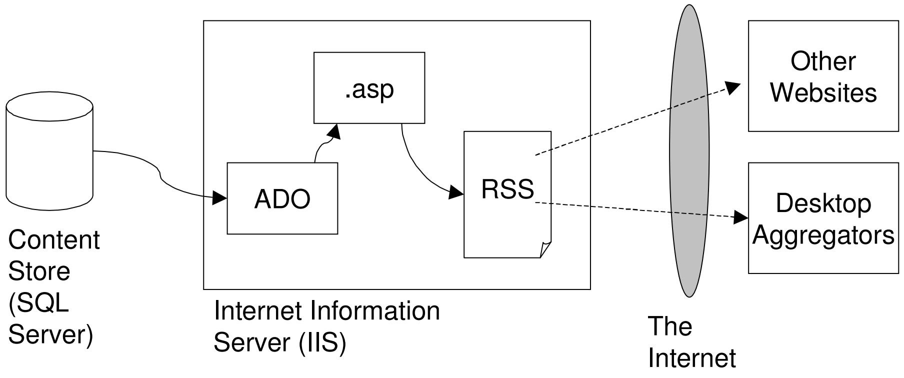
5 ASP HOW TO

5.1 Outbound Syndication on an ASP Based Website

The simplest way of doing this is to use a .asp file to

- Output fixed RSS header tags
- Open a database connection
- For each appropriate record read write out the corresponding rss item
- Output the RSS trailer tags.

The following diagram illustrates these steps:



Although very simple this technique has some weaknesses:

XML output is not checked as “well formed”. A simple coding glitch could result in badly formed XML being published.

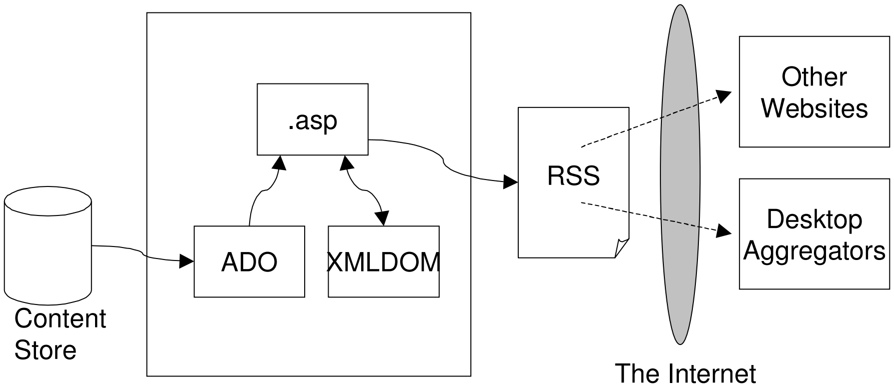
Examples of this approach are readily available on the Internet for example.

http://asprss.com/CreateRSS.asp

http://www.purplepages.ie/site/articles/article.asp?faq=6&fldAuto=76

The first example dynamically generates the RSS file on each page hit (somewhat inefficient,, but always up to date). The second example more closely reflects the diagram above in that it writes the output to an xml file which is picked up and served by ISS (more efficient but must be done more regularly). The approach is broadly similar though.

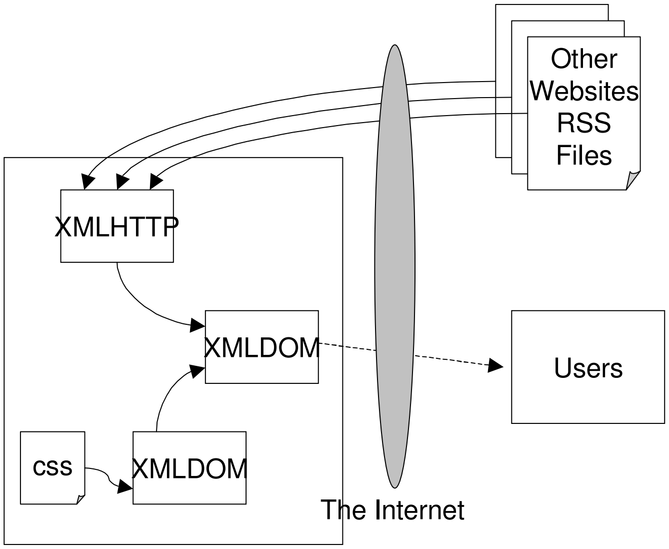
Although the approach above is a very simple solution to get an rss feed generated, it is probably better long term to use an XML parser (Microsoft XMLDOM) as shown below:



Using this method an RSS document is created as an XMLDOM object. Fixed fields are then set within it. The dynamic content is extracted from the database and added using DOM methods. Finally the whole XML document is written out to a static rss file on the site.

5.2 Inbound Syndication on an ASP Based Website

In this approach the other websites are read using the Microsoft XMLHTTP object. The response text from this is fed into the Microsoft XMLDOM object which parses the XML. A stylesheet is also parsed into an XMLDOM object. This is used to render the output.



Examples:

http://home.att.net/~codeLibrary/XML/rss.htm
http://www.edazzle.net/#grabnews

For .Net architecture sites Microsoft has defined a class XmlDocument which is analogous to XMLDOM above. In fact there is a whole range of XML support classes. See the Technet article below:

http://support.microsoft.com/default.aspx?scid=kb;EN-US;313651

Other .net web references:

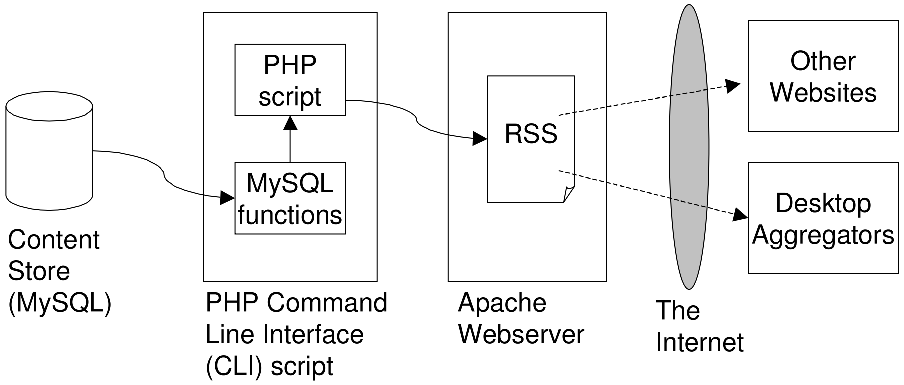
http://www.c-sharpcorner.com/Code/2002/Aug/RssFeedProject.asp (C#)
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnexxml/html/xml02172003.asp (although this is a desktop news aggregator this demonstrates some of the Microsoft technologies)

Summary of XML classes and methods in .net
http://www.fawcette.com/xmlmag/2002_04/online/online_eprods/xml_dwahlin04_29/default_pf.asp

6 PHP HOW TO

6.1 Outbound Syndication on a PHP Based Website

The simplest method is to retrieve the content from store and write directly to an rss file, without using an XML parser. This has the advantage of getting things going quickly.

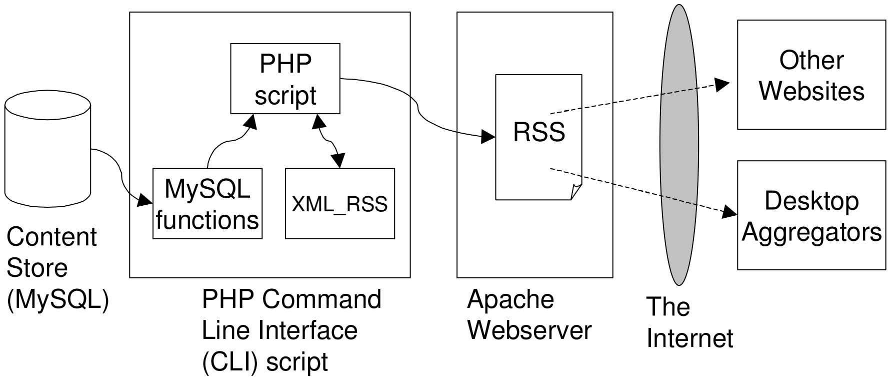


The PHP script simply reads data from the database, using the MySQL library commands and then directly builds up the output as PHP strings, finally writing the output to a rss file for the webserver to serve.

Example:

http://www.devarticles.com/art/1/238

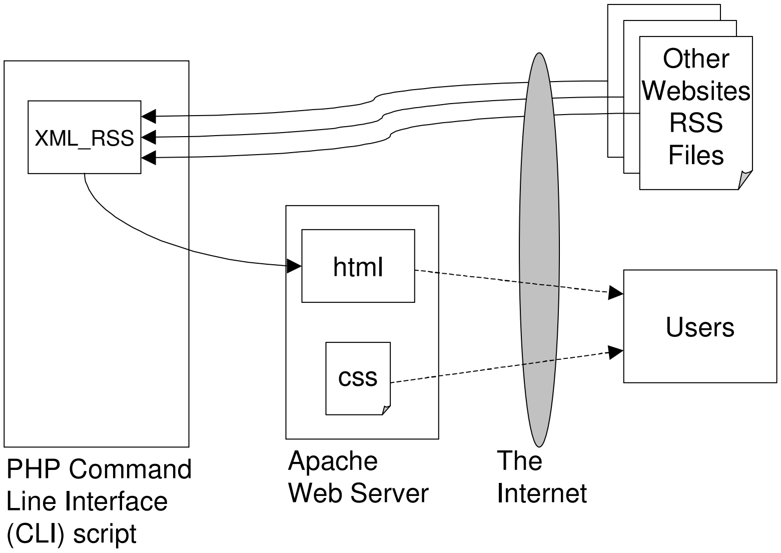
A more elegant solution would be to use the XML_RSS parser libraries as shown below:



Here the XML_RSS library ensures valid XML is produced.

6.2 Inbound Syndication on a PHP Based Website

In this approach the other websites are read using the Pear XML_RSS parser.



Example of above:

GPL newsfeed aggregator

http://www.cgiexpo.com/?qi=6780

Pear XML_RSS library used above

http://pear.php.net/package-info.php?pacid=22

PHP How to examples

http://magpierss.sourceforge.net/cookbook.html

rss_utils.inc

http://phpxref.sourceforge.net/sourceforge/export/rss_utils.inc.html

7 Other Options

Some other options for adding RSS capability to your site are discussed here.

7.1 Blogification

A very simple option, that could be used for testing or a simple demonstrator, is to submit your own (non-RSS) pages, suitably tagged to a 3rd party site offering a 'blogify' service e.g.

http://logicerror.com/blogifyYourPage

Your RSS service would be entirely dependent on the 3rd party and the terms of use of this site would need to be adhered to.

7.2 Coldfusion

For Coldfusion users here is an example page showing how to generate an RSS feed.

http://www.devx.com/xml/Article/10790/0/page/1

7.3 Web Based Newsreaders

These allow online reading of a newsfeed, via a 3rd party site. This is a way of testing your RSS feed by submitting it to a 3rd party site, and seeing if it is readable.

http://soapclient.com/rss/rss.html

7.4 Web Based Validators

These allow your feed to be checked for correctness

http://aggregator.userland.com/validator

http://asprss.com/RSSform.asp

http://feeds.archive.org/validator/check?url=<your RSS URL here>

8 References

Ref.	Description	Version or Doc. No.
1.	RSS 0.91 Jun 4, 2000 http://backend.userland.com/rss091	0.91

2.	RSS 1.0 2000-12-06. http://www.purl.org/rss/1.0/	1.0
3.	RSS 2.0 Aug 19, 2002 http://backend.userland.com/rss	2.0

9 Glossary

DMZ	De-Militarised Zone (where web servers are located)
RSS	An XML standard for syndication links and articles between sites. Originally stood for “Rich Site Summary”. Now taken to mean “Really Simple Syndication.”