

CHAPTER 4

Working with Feeds, RSS, and Atom

A fundamental enabling technology for mashups is syndication feeds, especially those packaged in XML. *Feeds* are documents used to transfer frequently updated digital content to users. This chapter introduces feeds, focusing on the specific examples of RSS and Atom. RSS and Atom are arguably the most widely used XML formats in the world. Indeed, there's a good chance that any given web site provides some RSS or Atom feed—even if there is no XML-based API for the web site. Although RSS and Atom are the dominant feed format, other formats are also used to create feeds: JSON, PHP serialization, and CSV. I will also cover those formats in this chapter.

So, why do feeds matter? Feeds give you structured information from applications that is easy to parse and reuse. Not only are feeds readily available, but there are many applications that use those feeds—all requiring no or very little programming effort from you. Indeed, there is an entire ecology of web feeds (the data formats, applications, producers, and consumers) that provides great potential for the remix and mashup of information—some of which is starting to be realized today.

This chapter covers the following:

- * What feeds are and how they are used
- * The semantics and syntax of feeds, with a focus on RSS 2.0, RSS 1.0, and Atom 1.0
- * The extension mechanism of RSS 2.0 and Atom 1.0
- * How to get feeds from Flickr and other feed-producing applications and web sites
- * Feed formats other than RSS and Atom in the context of Flickr feeds
- * How feed autodiscovery can be used to find feeds
- * News aggregators for reading feeds and tools for validating and scraping feeds
- * How to remix and mashup feeds with Feedburner and Yahoo! Pipes

Note In this chapter, I assume you have an understanding of the basics of XML, including XML namespaces and XML schemas. A decent tutorial on XML is available at <http://www.w3schools.com/xml/>. If you are new to the world of XML, working with RSS and Atom is an excellent way to get started with the XML family of technology.

What Are Feeds, and Why Are They Important?

Feeds are documents used to transfer frequently updated digital content to users. This content ranges from news items, weblog entries, installments of podcasts, and virtually any content that can be parceled out in discrete units. In keeping with this functionality, there is some commonly used terminology associated with feeds:

- * You *syndicate*, or *publish*, content by producing a feed to distribute it.
- * You *subscribe* to a feed by reading it and using it.
- * You *aggregate* feeds by combining feeds from multiple sources.

Although feeds come in many data formats, I focus in the following sections on three formats that you are likely to see in current web sites: RSS 2.0, Atom 1.0, and RSS 1.0. (Later in the chapter, I will mention other feed formats.) The formats have fundamental conceptual and structural similarities but also are different in fundamental ways. In addition, they have a complicated, interdependent, and contested history—which I do not untangle here.

The examples of the three feed formats are adapted from the RSS 2.0 feed of new books from Apress (<http://www.apress.com/rss/whatsnew.xml>). They are meant to be (as much as possible) the same data packaged in different formats. They are minimalist, though not the absolute minimal, example to illustrate the core of each format. For instance, the description elements have embedded HTML. Also, I show two items to illustrate that channels (feeds) can contain more than one item (entries). I discuss extensions to RSS and Atom later in the chapter.

RSS 2.0

There are two main branches of formats in the RSS family. RSS 2.0 is the current inheritor of the line of XML formats that includes RSS versions 0.91, 0.92, 0.93, and 0.94. The “RSS 1.0” section covers the other branch. You can find the specification for RSS 2.0 here, from which you can get the details of required and optional elements and attributes:

<http://cyber.law.harvard.edu/rss/rss.html>

Here are some key aspects of RSS 2.0:

- * The root element is `<rss>` (with the `version="2.0"` attribute).
- * The `<rss>` element must contain a single `<channel>` element, which represents the source of the feed.
- * A `<channel>` contains any number of `<item>` elements.
- * A `<channel>` is described by three mandatory elements (`<title>`, `<link>`, and `<description>`) contained within `<channel>`.
- * An `<item>` element is described by such optional elements, such as `<title>`, `<description>`, and `<link>`. An item must contain at least a `<title>` or `<description>` element.

- * The tags in RSS 2.0 are not placed in any XML namespaces to retain backward compatibility with 0.91–0.94.

Here is an example of an RSS 2.0 feed with two `<item>` elements, each representing a new book. Each description contains entity-encoded HTML.¹

```
<?xml version="1.0" ?>
<rss version="2.0">
  <channel>
    <title>Apress :: The Expert's Voice</title>
    <link>http://www.apress.com/</link>
    <description>
      Welcome to Apress.com. Books for Professionals, by Professionals(TM)...
      with what the professional needs to know(TM)</description>
    <item>
      <title>Excel 2007: Beyond the Manual</title>
      <link>http://www.apress.com/book/bookDisplay.html?bID=10232</link>
      <description>
        &lt;p&gt;&lt;i&gt;Excel 2007: Beyond the Manual&lt;/i&gt;
        will introduce those who are already familiar with Excel basics to more
        advanced features, like consolidation, what-if analysis, PivotTables,
        sorting and filtering, and some commonly used functions. You'll learn
        how to
        maximize your efficiency at producing professional-looking spreadsheets
        and
        charts and become competent at analyzing data using a variety of tools.
        The
        book includes practical examples to illustrate advanced
        features.&lt;/p&gt;
      </description>
    </item>
    <item>
      <title>Word 2007: Beyond the Manual</title>
      <link>http://www.apress.com/book/bookDisplay.html?bID=10249</link>
      <description>
        &lt;p&gt;&lt;i&gt;Word 2007: Beyond the Manual&lt;/i&gt; focuses on new
        features of Word 2007 as well as older features that were once less
        accessible than they are now. This book also makes a point to include
        examples of practical applications for all the new features. The book
        assumes familiarity with Word 2003 or earlier versions, so you can focus
        on
        becoming a confident 2007 user.&lt;/p&gt;
      </description>
    </item>
  </channel>
</rss>
```

1. http://examples.mashupguide.net/ch04/RSS2.0_Apress_simple_example.xml
RSS 1.0

As a data model, RSS 1.0 is similar to RSS 2.0, since both are designed to be represent feeds. In contrast to RSS 2.0, however, RSS 1.0 is expressed using the W3C RDF specification (<http://www.w3.org/TR/REC-rdf-syntax/>). Consequently, RSS 1.0 feeds are part of the Semantic Web, an ambitious effort of the W3C to build a “common

framework that allows data to be shared and reused across application, enterprise, and community boundaries...based on the Resource Description Framework (RDF).”²

Note Other than this description of RSS 1.0 and a brief analysis of RDFa in Chapter 18, the Semantic Web is beyond the scope of this book. Although I urge any serious student of mashups to track the Semantic Web for its long-term promise to transform the world of mashups, it has yet to make such an impact. Nonetheless, because RSS 1.0 is a concrete way to get started with RDF, I mention it here.

You can find the RDF 1.0 specification here:

<http://web.resource.org/rss/1.0/spec>

The RDF 1.0 format is associated with an RDF schema (<http://www.w3.org/TR/rdf-schema/>):

<http://web.resource.org/rss/1.0/schema.rdf>

Here I rewrite the RSS 2.0 feed to represent the same information as RSS 1.0 to give you a feel for the syntax of RSS 1.0:³

2. <http://www.w3.org/2001/sw/>
3. http://examples.mashupguide.net/ch04/RSS1.0_Apress.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://purl.org/rss/1.0/">
  <channel rdf:about="http://www.apress.com/rss/whatsnew.xml">
    <title>Apress :: The Expert's Voice</title>
    <link>http://www.apress.com/</link>
    <description>
      Welcome to Apress.com. Books for Professionals, by Professionals(TM)...
      with what the professional needs to know(TM)
    </description>
    <items>
      <rdf:Seq>
        <rdf:li rdf:resource="http://www.apress.com/book/bookDisplay.html?
bID=10232" />
        <rdf:li rdf:resource="http://www.apress.com/book/bookDisplay.html?
bID=10249" />
      </rdf:Seq>
    </items>
  </channel>
  <item rdf:about="http://www.apress.com/book/bookDisplay.html?bID=10232">
    <title>Excel 2007: Beyond the Manual</title>
    <link>http://www.apress.com/book/bookDisplay.html?bID=10232</link>
    <description>
      &lt;p&gt;&lt;i&gt;Excel 2007: Beyond the Manual&lt;/i&gt; will introduce
those
who are already familiar with Excel basics to more advanced features, like
consolidation, what-if analysis, PivotTables, sorting and filtering, and
some
commonly used functions. You'll learn how to maximize your efficiency at
```

```

        producing professional-looking spreadsheets and charts and become
competent at
        analyzing data using a variety of tools. The book includes practical
examples
        to illustrate advanced features.&lt;/p&gt;
        </description>
    </item>
    <item rdf:about="http://www.apress.com/book/bookDisplay.html?bID=10249">
        <title>Word 2007: Beyond the Manual</title>
        <link>http://www.apress.com/book/bookDisplay.html?bID=10249</link>
        <description>
            &lt;p&gt;&lt;i&gt;Word 2007: Beyond the Manual&lt;/i&gt; focuses on new
            features of Word 2007 as well as older features that were once less
accessible
            than they are now. This book also makes a point to include examples of
            practical applications for all the new features. The book assumes
familiarity
            with Word 2003 or earlier versions, so you can focus on becoming a
confident
            2007 user.&lt;/p&gt;
        </description>
    </item>
</rdf:RDF>

```

Consider the following aspects of RSS 1.0:

- * Note the commonality in data structure between RSS 1.0 and RSS 2.0 in the use of such elements as `<channel>`, `<item>`, `<title>`, and `<description>`.
- * It uses an XML namespace associated with RDF (<http://www.w3.org/1999/02/22-rdf-syntax-ns#>) and a default namespace related to RSS 1.0 (<http://purl.org/rss/1.0/>) to place all elements such as `<channel>`, `<item>`, and `<title>` into that namespace.
- * It uses an enclosing `<rdf:RDF>` root element.
- * Note the sequencing of `<rdf:resources>` contained by an `<items>` element.
- * Note the placement of the `<item>` elements outside the `<channel>` element.

Since RSS 1.0 feeds are harder to find than RSS 2.0 and Atom 1.0 feeds, the following are some examples of RSS 1.0 feeds:

- * <http://rss.slashdot.org/Slashdot/slashdot>
- * http://www.nature.com/nature/current_issue/rss/index.html (drawn from a list at <http://www.nature.com/webfeeds/index.html>)
- * <http://www.w3.org/2000/08/w3c-synd/home.rss>
- * <http://simile.mit.edu/blog/?feed=rdf>

Note There are efforts to update RSS. The RSS Advisory Board (<http://www.rssboard.org/>) has

been designing updates to RSS 2.0, whereas RSS 1.1 (<http://inamidst.com/rss1.1/>) has been created by a small number of developers to enhance RSS 1.0. RSS 2.0 and RSS 1.0 remain the most important versions of the two major families of RSS specifications.

Atom 1.0

The name Atom applies to two related proposed standards: the Atom Syndication Format (whose current version is also known as Atom 1.0) and the Atom Publication Protocol (APP). Here, I discuss Atom 1.0 and return to APP later in this book in the context of various Google web services that use GData, an extension of APP.

Designed to overcome perceived shortcomings of the various RSS formats, Atom 1.0 is currently a proposed IETF standard:

<http://tools.ietf.org/html/rfc4287>

Atom 1.0, constructed to syndicate web content, has a similar semantics to RSS but a different naming scheme. In an Atom document, a `<feed>` element is composed of one or more `<entry>` elements, each described by a set of tags such as `<title>`, `<link>`, `<id>`, and `<summary>`.

Let me now rewrite the sample Apress “new books” feed into Atom 1.0:⁴

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Apress :: The Expert's Voice</title>
  <subtitle>
    Welcome to Apress.com. Books for Professionals, by Professionals(TM)...
    with what the professional needs to know(TM)
  </subtitle>
  <link rel="alternate" type="text/html" href="http://www.apress.com/" />
  <link rel="self"
href="http://examples.mashupguide.net/ch04/Atom1.0_Apress.xml" />
  <updated>2007-07-25T12:57:02Z</updated>
  <author>
    <name>Apress, Inc.</name>
    <email>support@apress.com</email>
  </author>
  <id>http://apress.com/</id>
  <entry>
    <title>Excel 2007: Beyond the Manual</title>
    <link href="http://www.apress.com/book/bookDisplay.html?bID=10232" />
    <id>http://www.apress.com/book/bookDisplay.html?bID=10232</id>
    <updated>2007-07-25T12:57:02Z</updated>
    <summary type="html">
      &lt;p&gt;&lt;i&gt;Excel 2007: Beyond the Manual&lt;/i&gt; will introduce
those
who are already familiar with Excel basics to more advanced features, like
consolidation, what-if analysis, PivotTables, sorting and filtering, and
some
commonly used functions. You'll learn how to maximize your efficiency at
producing professional-looking spreadsheets and charts and become
competent at
```

```

    analyzing data using a variety of tools. The book includes practical
examples
    to illustrate advanced features.</p>
  </summary>
</entry>
<entry>
  <title>Word 2007: Beyond the Manual</title>
  <link href="http://www.apress.com/book/bookDisplay.html?bID=10249"/>
  <id>http://www.apress.com/book/bookDisplay.html?bID=10249</id>
  <updated>2007-07-25T12:57:10Z</updated>
  <summary type="html">
    &lt;p>&lt;i>Word 2007: Beyond the Manual&lt;/i> focuses on new
    features of Word 2007 as well as older features that were once less
accessible
    than they are now. This book also makes a point to include examples of
    practical applications for all the new features. The book assumes
familiarity
    with Word 2003 or earlier versions, so you can focus on becoming a
confident
    2007 user.</p>
  </summary>
</entry>
</feed>

```

4. http://examples.mashupguide.net/ch04/Atom1.0_Apress.xml

Note the following about this example:

- * Note the use of a default Atom-related XML namespace (<http://www.w3.org/2005/Atom>).
- * The `<subtitle>` element instead of `<description>` (in RSS) describes the feed.
- * The `<feed>` and `<entry>` elements must both include an `<updated>` element.
- * `<link rel="alternate" type="text/html" href="http://www.apress.com/" />` indicates that the document is an “alternate” representation (that is, a feed) of the web page <http://www.apress.com/>.
- * `<link rel="self" href="http://examples.mashupguide.net/ch04/Atom1.0_Apress.xml" />` indicates the location of this feed document.
- * The attribute `type="html"` in the `<summary>` elements indicates the use of entity-encoded HTML.

Writing a simple feed as RSS 2.0 and Atom 1.0 sheds some light on how the two formats compare. For a more detailed analysis, see the following:

http://en.wikipedia.org/wiki/Atom_%28standard%29#Atom_Compared_to_RSS_2.0

Finally, Atom 1.0 has an official RNG schema, defined in the appendix of RFC 4287:

<http://atompub.org/rfc4287.html#schema>

Extensions to RSS 2.0 and Atom 1.0

Extensions to RSS 2.0 and Atom 1.0 enable you to take advantage of the popular feed formats to be able to move information within the whole feed ecology while adding more information than is allowed in the simple base RSS or Atom vocabulary.

You can insert foreign XML elements (ones that are not defined in the respective specifications) into RSS 2.0 and Atom 1.0 by using XML namespaces. That is, with a few exceptions in Atom 1.0,⁵ foreign tags are allowed as long as they are qualified in a namespace that is different from that of the base format. For RSS 2.0, that would mean the foreign tag would have to be placed in some namespace instead of having no namespace such as the core elements in RSS 2.0.

5. Foreign markup is permitted unless explicitly forbidden for specific contexts. See <http://tools.ietf.org/html/rfc4287#section-6> for more details.

Note I do not cover RSS 1.0 extensibility here other than to refer readers to RSS 1.0 Modules (<http://web.resource.org/rss/1.0/modules/>) and to note the standard modules: (<http://web.resource.org/rss/1.0/modules/dc/> (Dublin Core), <http://web.resource.org/rss/1.0/modules/syndication/> (Syndication), and <http://web.resource.org/rss/1.0/modules/content/> (Content)).

Let's look at a simple example by adding a tag to the Atom 1.0 feed listed previously. Suppose you want to add a tag called `<isbn>` for each of the `<entry>` elements. You can do so by associating the `<isbn>` tag with a namespace (say, <http://mashupguide.net>):⁶

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:mg="http://mashupguide.net">
  [...]
  <entry>
    <title>Excel 2007: Beyond the Manual</title>
    <link href="http://www.apress.com/book/bookDisplay.html?bID=10232"/>
    <id>http://www.apress.com/book/bookDisplay.html?bID=10232</id>
    <updated>2007-07-25T12:57:02Z</updated>
    <mg:isbn>1590597982</mg:isbn>
  [...]
  </entry>
</feed>
```

This example is not meant to show the best way to encode an ISBN but to show how to extend Atom 1.0. Although inserting your own custom vocabulary results in a completely valid document, it doesn't necessarily help in terms of interoperability. How much software out there is set to interpret an `<isbn>` element in the <http://mashupguide.net> namespace—other than to ignore it? If you use a widely used extension, the better your chances that there is software that acts on those extensions. Some prominent RSS 2.0 extensions are as follows:

- * Media RSS (<http://search.yahoo.com/mrss>), used in Flickr

- * iTunes (<http://www.apple.com/itunes/store/podcaststechspecs.html>), used for podcasting
- * OpenSearch (<http://www.opensearch.org/Specifications/OpenSearch/1.1>)

Tip Other XML dialects use XML namespaces in a similar fashion to enable extensions, so it's useful to understand how extensions work in feeds to get a handle of how it works elsewhere.

6. http://examples.mashupguide.net/ch04/Atom1.0_Apress_ISBN.xml

There are few widely used Atom 1.0 extensions at this point. If you want to follow that topic, I suggest the series of articles by James Snell on Atom 1.0 and various proposed extensions.⁷

Now that you have studied three important formats for feeds (RSS 2.0, RSS 1.0, and Atom 1.0), you'll learn how feeds are implemented in Flickr and then in other web sites.

Feeds from Flickr

You can find feeds in Flickr in several ways. First, you can look throughout the Flickr UI for the orange feed icon and the text *Subscribe To*, as shown in Figure 4-1. You can then find out the feed's URL from the *Feed* link. Once you have that URL, you can subscribe to the feed and read the data it contains. For example, if you go to the following address:

<http://flickr.com/groups/central/>

you will find a the feed icon and link pointing here:

http://api.flickr.com/services/feeds/groups_discuss.gne?id=34427469792@N01&lang=en-us&format=rss_200

Insert 858Xf0401.tif

Figure 4-1. Icon for subscribing to a feed in Flickr. (Reproduced with permission of Yahoo! Inc. © 2007 by Yahoo! Inc. YAHOO! and the YAHOO! logo are trademarks of Yahoo! Inc.)

A second way of finding feeds is to consult Flickr's documentation of its feeds:

<http://www.flickr.com/services/feeds/>

In a moment, I'll list the Flickr feeds that are available following the same convention of using URI templates as I did in Chapter 2. All the feeds share two common optional parameters: `format` and `lang`.

Flickr Feed Parameters

Let's look first at `format`, which can be one of the values listed in Table 4-1.

Table 4-1. Values for the `format` Parameter in Flickr Feeds

Format Definition

rss_200 or **rss2** RSS 2.0

atom_1 or **atom** Atom 1.0

rss_091 RSS 0.91

rss_092 or **rss** RSS 0.92

rss_100 or **rdf** RSS 1.0

rss_200_enc RSS 2.0 with enclosures (but without enclosure sizes)

php Code to represent feed as a PHP array

php_serial Input to the PHP unserialize function

csv Comma-separated value

json JavaScript Object Notation (<http://www.json.org/>)

sql Statements to store the feed data into a SQL database

yaml YAML (<http://en.wikipedia.org/wiki/YAML>)

cdf Channel Definition Format
(http://en.wikipedia.org/wiki/Channel_Definition_Format)

7. <http://www-128.ibm.com/developerworks/xml/library/x-atom10.html>, <http://www-128.ibm.com/developerworks/xml/library/x-extatom1/>, and <http://www-128.ibm.com/developerworks/xml/library/x-extatom2.html>

If **format** is not specified, Atom 1.0 is assumed. Note that RSS 2.0, RSS 1.0, and Atom 1.0 (along with RSS 0.92 and RSS 0.91) are included among the formats. I'll present some samples of various formats later in this chapter, after I cover the rest of the URL language for Flickr feeds.

The second pervasive and optional parameter is **lang**, which represents the language you can use to query Flickr. Table 4-2 lists the values. The default language is **en-us** (English).

*Table 4-2. Values for the **lang** Parameter in Flickr Feeds*

Format Definition

de-de German

en-us English

es-us Spanish

fr-fr French

it-it Italian

ko-kr Korean

pt-br Portuguese (Brazilian)

zh-hk Traditional Chinese (Hong Kong)

I'll now list the feeds available. Remember that all the feed URLs can include the optional `format` and `lang` parameters.

Examining the Flickr Feeds

Feeds for public photos in Flickr are available here:

`http://api.flickr.com/services/feeds/photos_public.gne`

with the following optional parameters:

- * `id={user_nsid}`
- * `ids={comma_delimited_user_nsids}`
- * `tags={comma_delimited_tags}`
- * `tagmode={mode}` where `mode` is `all` (the default value) or `any`

For example, the following:

`http://api.flickr.com/services/feeds/photos_public.gne?tags=flower%2CBerkeley&format=rss2&lang=fr-fr`

returns a RSS 2.0 feed, annotated in French, of recent public photos tagged with both `flower` and `Berkeley` tags.

You can get a feed of recent photos of a user's friends here:

`http://api.flickr.com/services/feeds/photos_friends.gne?user_id={user-nsid}`

where `user-nsid` is the NSID of the user whose friends' photos you want to access. There are also optional parameters:

- * `display_all` can be 1 to show multiple photos per friend, instead of the default value of one photo per friend.
- * `friends` can be set to 1 to limit photos to only the family and friends for the requested user.

For example, the following:

`http://api.flickr.com/services/feeds/photos_friends.gne?user_id=48600101146@N01&friends=0&display_all=0&lang=en-us&format=atom_1`

is an Atom 1.0 feed of up to one photo each from my Flickr friends, family, and contacts.

You can get feeds of a group discussion here:

`http://api.flickr.com/services/feeds/groups_discuss.gne?id={group-nsid}`

Feeds for the group photo pools are accessible here:

`http://api.flickr.com/services/feeds/groups_pool.gne?{group-nsid}`

Discussion feeds from the Help forum (<http://www.flickr.com/help/forum/en-us/>) are here:

`http://api.flickr.com/services/feeds/forums.gne`

You can track recent activity on a user's photos through feeds here:

<http://api.flickr.com/services/feeds/activity.gne?user-id={user-nsid}>

Feeds of recent comments made by a user are here:

http://api.flickr.com/services/feeds/photos_comments.gne?user-id={user-nsid}

You can track Flickr news bulletins here:

<http://api.flickr.com/services/feeds/news.gne>

Now that you know how to access the various feeds available in Flickr, you'll look at some examples of feeds to understand how various formats are implemented, including the role of extensions. Consider the following excerpt of a sample RSS 2.0 feed of recent public photos with the tag **tree**:

http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=rss2

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:media="http://search.yahoo.com/mrss/"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
  <channel>
    <title>tree - Everyone's Tagged Photos</title>
    <link>http://www.flickr.com/photos/tags/tree/</link>
    <description>A feed of tree - Everyone's Tagged Photos</description>
    <pubDate>Mon, 29 Jan 2007 06:40:42 -0800</pubDate>
    <lastBuildDate>Mon, 29 Jan 2007 06:40:42 -0800</lastBuildDate>
    <generator>http://www.flickr.com/</generator>
    <image>
      <url>http://www.flickr.com/images/buddyicon.jpg</url>
      <title>tree - Everyone's Tagged Photos</title>
      <link>http://www.flickr.com/photos/tags/tree/</link>
    </image>
    <item>
      <title>Odd Tree</title>
      <link>http://www.flickr.com/photos/davidleong/373343287/</link>
      <description>[...]</description>
      <pubDate>Mon, 29 Jan 2007 06:40:42 -0800</pubDate>
      <dc:date.Taken>2007-01-28T11:31:31-08:00</dc:date.Taken>
      <author>nobody@flickr.com (mountainhiker)</author>
      <guid isPermaLink="false">tag:flickr.com,2004:/photo/373343287</guid>
      <media:content
        url="http://farm1.static.flickr.com/127/373343287_df43da61f7_m.jpg"
        type="image/jpeg" height="160" width="240"/>
      <media:title>Odd Tree</media:title>
      <media:text type="html">[...]</media:text>
      <media:thumbnail
        url="http://farm1.static.flickr.com/127/373343287_df43da61f7_s.jpg"
        height="75" width="75"/>
      <media:credit role="photographer">mountainhiker</media:credit>
      <media:category scheme="urn:flickr:tags">snow tree vancouver
northvancouver
      grousemountain</media:category>
    </item>
  </channel>
  [...]
```

```
</item>
</channel>
</rss>
```

Note the following about this XML fragment:

- * Within an `<item>` element are some standard elements that you will find in RSS 2.0 feeds, namely, `<title>`, `<link>`, `<description>`, `<pubDate>`, and `<guid>`, as well as elements that are extensions of RSS 2.0, including `<dc.date.Taken>` and the tags in the `media` namespace such as `<media:thumbnail>`.
- * Remember that RSS is basically a flat structure. That is, a `<channel>` consists of `<item>` elements—but an `<item>` doesn't consist of other `<item>` elements. This contrasts with other XML formats that you will learn about in later chapters of the book (such as the OpenDocument format).

Let's take a closer look at the extensions used in the RSS 2.0 Flickr feed:

- * The `dc` prefix (for example, in `<dc:date.Taken>`), corresponding to the <http://purl.org/dc/elements/1.1/> namespace, denotes the Dublin Core (DC) metadata standard. DC is a vocabulary for a core set of metadata designed to be applicable to a wide range of digital content.
- * Media RSS (<http://search.yahoo.com/mrss>), which supplements the `<enclosure>` element in RSS 2.0, is used by Flickr to store metadata of the photos, such as the following:
 - * `<media:content>`, with the attributes `url`, `type`, `height`, and `width`
 - * `<media:title>` to hold the photo's title
 - * `<media:text>` with HTML to embed that picture into a web page
 - * `<media:thumbnail>` to hold a URL to the thumbnail in the `url` attribute
 - * `<media:credit>` to indicate the owner of the photo
 - * `<media:category>` to holds tags

It's interesting to look at how all the Flickr metadata is expressed in the other feed formats. The following:

```
http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=atom
```

shows that Atom 1.0 is natively able to encode much of the information for which Media RSS is being used in RSS 2.0. The Atom feed does, however, use a Dublin Core extension to mark up a date.

Exchange Formats Other Than RSS and Atom

Let's now return to the feed formats supported by Flickr other than RSS 2.0, RSS 1.0, and Atom 1.0. To get a start at understanding the various formats, load a Flickr feed, which will by default be in Atom 1.0, and replace the `format` parameter with the alternatives. For instance, if you are curious about CDF, change the following:

```
http://api.flickr.com/services/feeds/photos_public.gne?tags=tree
```

to the following:

http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=cdf

In the following sections, I'll mention some but not all of formats supported in Flickr feeds.

RSS 0.91 and RSS 0.92

Although RSS 0.91 and RSS 0.92 have largely been superseded by RSS 2.0, you can still look at examples such as this:

http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=rss_091

and this:

http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=rss_092

to see how those older versions do not contain any of the Media RSS or Dublin Core extensions you find in the RSS 2.0 feeds.

JSON

The JSON format facilitates the consumption of Flickr feeds in JavaScript. I discuss JSON at greater length in Chapter 8. However, if you look at the following:

http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=json

you can see that Flickr returns JavaScript code, specifically a call to a `jsonFlickrFeed()` function with one parameter. This parameter is a JavaScript object that holds the feed data. As a JavaScript programmer, you can use this JSON-formatted feed by supplying a `jsonFlickrFeed()` function to do something with the feed, such as displaying it on a web page.

Here's an example of some code that reads the Flickr JSON feed and renders the feed as HTML in the browser:⁸

8. <http://examples.mashupguide.net/ch04/Flickr.JSON.html>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Flickr JSON feed</title>
    <script type="text/javascript">
      //

      function jsonFlickrFeed (feed) {
        var feed_div = document.getElementById("feed");
        var feed_html = '&lt;p&gt;' + '&lt;a href="' + feed.link + '"'&gt;' +
          feed.title + '&lt;/a&gt;' + '&lt;/p&gt;';
        for (x=0; x&lt;feed.items.length; x++) {
          feed_html += '&lt;a href="' + feed.items[x].link + '"'&gt;' +
            '&lt;img ' + 'src="' + feed.items[x].media["m"] + '"' + ' alt="' +
            feed.items[x].title + '"' + '&gt;' + '&lt;/a&gt;' + '&lt;br/&gt;';
        }
        feed_div.innerHTML = feed_html;
      }
    ]]&gt;
  &lt;/script&gt;
&lt;/html&gt;</pre></div>
```

```

    }

    function load() {
        var head_element = document.getElementsByTagName("head")[0];
        var newScript = document.createElement('script');
        newScript.type = 'text/javascript';
        newScript.src = 'http://api.flickr.com/services/feeds/photos_public.gne?
tags=tree&format=json';
        head_element.appendChild(newScript);
    }

    //]]>
</script>
</head>
<body onload="load()">
    <div id="feed" />
</body>
</html>

```

In Chapter 8, I'll return to how to use JSON in mashups.

Other Feed Formats

The other available formats are less commonly used to represent feeds in web sites other than Flickr but come in handy depending on your specific needs. For example:

- * You can use the `php` or `php_serial` format to generate a convenient representation of the feed for PHP programming. (These formats have roughly the relationship to PHP that the `json` format has to JavaScript.)
- * You can use the `sql` format to quickly generate SQL code to get the Flickr feed into your SQL database.
- * The `csv` format comes in handy for importing your feed into a spreadsheet.
- * The `rss_200_enc` format is used to insert a reference to the original photo in an RSS 2.0 `<enclosure>` tag.

Feeds from Other Web Sites

Feeds are extremely helpful in creating mashups because feeds are packaged in formats designed to be accurately and automatically parsed by software. Not only do they not require programming to use—they are widely available, much more so than web APIs.

Nonetheless, feeds are still sometimes difficult to find. I first revisit the question of how to find feeds and the topic of autodiscovery. I then provide examples of feeds that are available from some specific web sites: a selection of blogs, Wikipedia, Google, and Yahoo! News. You will see how web sites other than Flickr use feeds. Moreover, I have focused in my examples on news-oriented web sites because I draw upon such sites in the feed mashups I create with Yahoo! Pipes later in the chapter.

Finding Feeds and Feed Autodiscovery

In the context of Flickr, I mention two ways of finding feeds that are applicable to other web sites:

- * Looking in the user interface for features such as the common orange icon or the words *feed*, *RSS*, *subscribe*, and so on
- * Finding documentation for a web site's feeds

Let's explore some other approaches to finding feeds. There are specialized feed directories and search engines such as the following, which also has an API (in case you find it useful):

<http://www.syndic8.com/>

Some of this feed search functionality has been incorporated into feed aggregators (which I describe more in a moment). For instance, you can browse and search for feeds from within Google Reader. This search functionality is also available from the Google AJAX Feed API.⁹ Some have used general-purpose search engines to search for feeds, but it's hard to say how reliable such self-described hacks are.¹⁰

It seems sensible that if you know the URL of a web page, you should be able to easily figure out the URL for any feeds that are associated with it. Indeed, a mechanism called *RSS autodiscovery* (or more generally, *feed autodiscovery*) has become a de facto standard in associating web pages with feeds. To connect a web page to a feed, you add `<link>` elements to the `<head>` element, making appropriate use of the `rel`, `href`, and `type` (and optionally `title`) attributes of `<link>`:

- * `rel` is set to the value `alternate`.
- * `href` is the URL of the feed.
- * `type` is set to the MIME type of the feed (either `application/rss+xml` or `application/atom+xml`).
- * `title` is optionally set to be a title of the feed.

For example, in the following `<head>` element:

<http://news.yahoo.com>

you find the following `<link>`, which points to a corresponding RSS feed at <http://rss.news.yahoo.com/rss/topstories>:

```
<link rel="alternate" type="application/rss+xml" title="Yahoo! News - Top
Stories"
href="http://rss.news.yahoo.com/rss/topstories" />
```

Many of the modern browsers support feed autodiscovery. If you use any of those browsers to go to a web page with a link to its feeds, you'll see an icon that leads to those feeds.

Autodiscovery is similarly useful for creators of mashups. For example, if your program is fed the URL of a web page, you could look for the presence of associated feeds that might give you the data you need by using feed autodiscovery.

9. <http://code.google.com/apis/ajaxfeeds/>

10. <http://www.xml.com/pub/a/2004/02/11/googlexml.html>

OFFICIAL STANDARDIZATION OF FEED AUTODISCOVERY?

Even though feed autodiscovery has been widely implemented, there is currently no de jure standard for this practice. Autodiscovery started as a collaboration carried out through weblogs (such as http://diveintomark.org/archives/2002/06/02/important_change_to_the_link_tag), progressed to being discussed as an IETF draft (whose last expired version was <http://www.ietf.org/internet-drafts/draft-snell-atompub-autodiscovery-00.txt>), and now is being considered in the context of standardization as part of HTML 5 (<http://www.whatwg.org/specs/web-apps/current-work/#alternate>).

In the meantime, some of the current practice around feed autodiscovery is documented in places such as the wiki at <http://www.feedautodiscovery.org/doku.php>.

Feeds from Weblogs

Weblogs are a major source of feeds because almost all modern weblog software produces feeds, which are often turned on by default. For example:

- * Blogspot weblogs have Atom feeds¹¹ (for example, <http://googleblog.blogspot.com/atom.xml> and <http://googleblog.blogspot.com/feeds/posts/default>).
- * WordPress blogs¹² (for example, <http://blog.mashupguide.net/feed/> and <http://blog.mashupguide.net/feed/atom/>).
- * TypePad blogs support feeds.¹³

Wikipedia Feeds

Let's look at what Wikipedia has in the way of feeds to supplement Flickr as an example and to be of use in the following case studies. Wikipedia is a great source of information about the news and publishes RSS feeds. Here's some documentation for the feeds:

- * <http://meta.wikimedia.org/wiki/RSS>
- * <http://en.wikipedia.org/wiki/Wikipedia:Syndication>.

You can get a feed for the history of any regular page here:

<http://en.wikipedia.org/w/index.php?title={page-name}&action=history&feed={format}>

11. <http://help.blogger.com/bin/topic.py?topic=8927>

12. http://codex.wordpress.org/WordPress_Feeds

13. http://support.typepad.com/cgi-bin/typepad.cfg/php/enduser/std_adp.php?p_faqid=86

For example:

http://en.wikipedia.org/w/index.php?title=Hurricane_Katrina&action=history&feed=atom

http://en.wikipedia.org/w/index.php?title=Mashup_%28web_application_hybrid%29&action=history&feed=atom

Two of Wikipedia's special pages also have feeds. The first is of all recent changes to Wikipedia (which tends to have way too much data because Wikipedia is extremely active):

<http://en.wikipedia.org/wiki/Special:Recentchanges?feed={format}>

and the other lets you track the creation of new pages:

<http://en.wikipedia.org/wiki/?Special:Newpagesfeed={format}>

If you want to track news using Wikipedia, you might want to use Wikinews (http://en.wikinews.org/wiki/Main_Page), which has an RSS feed:

<http://feeds.feedburner.com/WikinewsLatestNews>

Finally, you can get at your Wikipedia watch list (when logged in) here:

<http://en.wikipedia.org/w/api.php?action=feedwatchlist&feedformat={format}>

where **format** is **rss** or **atom**.

Google and Yahoo! News

The feeds for Google News are documented here:

http://news.google.com/intl/en_us/news_feed_terms.html

You can access a variety of U.S.-oriented feeds here:

<http://news.google.com/news?ned=us&topic={topic}&output={format}>

where **output** is **rss** or **atom** and where **topic** is one of the values listed in Table 4-3.

*Table 4-3. Possible Values for **topic** in Google News Feeds*

Topic	Coverage
h	Top news
w	World
n	United States
b	Business
t	Science/technology
m	Health
s	Sports
e	Entertainment

For example, you can get the top news in RSS here:

<http://news.google.com/news?ned=us&topic=h&output=rss>

You can also get international news here:

<http://news.google.com/news?ned={region}&topic=n&output={format}>

where **region** is one of the values listed in Table 4-4.

Table 4-4. Possible Values for **region** in Google News Feeds

Region	Country
au	Australia
ca	Canada
in	India
ie	Ireland
nz	New Zealand
en_za	South Africa
uk	United Kingdom

In addition to feeds for general topics, you can generate a feed for a specific search term in Google News (an extremely useful feature you will use when constructing targeted feeds later in the chapter):

```
http://news.google.com/news?q={query}&output={output}
```

For example, to follow news on mashups, use this:

```
http://news.google.com/news?q=mashup&output=rss
```

Yahoo! News has some similarities to Google News. In addition to getting feeds by large categories, listed here:

```
http://news.yahoo.com/rss
```

you can also get feeds by keywords via

```
http://news.search.yahoo.com/news/rss?p={search-term}. For example:
```

```
http://news.search.yahoo.com/news/rss?p=Hurricane+Katrina
```

News Aggregators: Showing Flickr Feeds Elsewhere

A primary use of feeds is to allow you as an end user to keep up with lots of information from many different sources—all in one place. News aggregators (also known as *feed readers*) gather items from the feeds you subscribe to and present them to you to read in a single interface.

Subscribing to feeds has become such a sufficiently mainstream activity for web users that modern web browsers now provide options for doing so when the user arrives at an RSS or Atom feed in the browser. For example, in Firefox 2.0+, you see options for how to subscribe to that feed, as shown in Figure 4-2.

Insert 858Xf0402.tif

Figure 4-2. Choosing a news aggregator with which to subscribe to a feed in Firefox

There are different news/feed aggregators of note:

- * Firefox Live Bookmarks. You can track feeds within the context of Firefox bookmarks.¹⁴ There seem to be similar features in other browsers such as Safari¹⁵ and Internet Explorer 7.¹⁶
- * Bloglines (<http://www.bloglines.com/>).
- * SharpReader, a desktop RSS aggregator/news reader for Windows (<http://www.sharpreader.net/>).
- * NetNewsWire, a desktop news reader for the Mac. (<http://www.newsgator.com/NGOLProduct.aspx?ProdID=NetNewsWire>).
- * Google Reader (<http://www.google.com/reader/view/>).
- * My Yahoo! You can add an RSS feed to <http://my.yahoo.com>. You can, for instance, add an RSS feed with this URL: <http://e.my.yahoo.com/config/cstore?.opt=rss&.page=p1>. For more information, see the following:

- * <http://my.yahoo.com/s/faq/rss/>
- * <http://publisher.yahoo.com/rssguide>

14. <http://www.mozilla.com/en-US/firefox/livebookmarks.html>
15. <http://www.apple.com/macosx/features/safari/>
16. <http://www.microsoft.com/windows/rss/default.mspx>

Validating Feeds

In addition to consuming feeds, you may want to create feeds as part of your mashups. It's certainly helpful to read and understand the specifications for the various feed formats. I have found the following feed validators to be invaluable in helping me to spot and correct errors in feeds that I create or read from others:

- * <http://feedvalidator.org> is an online service, whose software you can also run locally.¹⁷
- * <http://validator.w3.org/feed/> is a W3C service built from the software available at <http://feedvalidator.org>. The syntax checking is available also as a SOAP web service.¹⁸
- * For dealing with RSS 1.0, you may find the W3C RDF Validation Service useful (<http://www.w3.org/RDF/Validator/>).

Scraping Feeds Using GUI Tools

Feeds are available for many applications—but by no means for all applications. Because feeds are so useful, some services have arisen to generate feeds out of unstructured web sites. The goal of these services is to enable you to construct feeds more easily than you could screen-scrape the pages yourself—which, as I discuss in Chapter 2, is an option absent of APIs and feeds. Let's briefly consider one usage scenario to which we will apply two services. (I return to this topic of feed-scraping in Chapter 11.)

As I mention elsewhere in this book, perhaps the single most useful site on the Web for tracking web APIs is Programmableweb.com. Currently, it does not have an API and does not have a feed to represent all the APIs tracked by the site, but there is a feed for the latest changes in the list of APIs. The scenario I explore here is creating an RSS or Atom feed out of the list of APIs here:

<http://programmableweb.com/apis/directory>

Here I apply two services to this problem. The first is a specialized feed-creation web site:

<http://www.feedity.com/>

You can use Feedity to generate an RSS feed:

<http://feedity.com/?http://programmableweb.com/apis/directory%40%40%40CAT%40%40%406>

The feed is a perfectly fine feed except for the ads embedded in the feed. You need to use Pro (for-fee) level to get rid of the ads.

I used Openkapow.com's RoboMaker as a second approach to generate a feed. RoboMaker is a desktop visual tool to create bots hosted on Openkapow.com to generate feeds and APIs for web sites. In Chapter 11, I analyze RoboMaker and other tools that simplify mashup making. Here, I simply point out the end product of the Openkapow.com bot that converts the list of APIs into an RSS 2.0 feed:

<http://service.openkapow.com/rdhyee/programmablewebapis.rss>

There is a small image for Openkapow.com in the feed but no advertisements buried in the items themselves.

As you will see in the next section, being able to generate feeds for sites that don't have the feeds you want enables you to use the many tools that accept feeds as input.

17. http://feedvalidator.org/docs/howto/install_and_run.html

18. <http://validator.w3.org/feed/docs/soap>

Remixing Feeds with Feedburner

Feedburner (<http://feedburner.com>) lets users remix feeds and offers intermediary services based on feeds (such as tracking usage and advertising). It thus provides a useful illustration of the ways some users and companies are reusing and repackaging feeds.

The best way to understand Feedburner is to study the effect various options have on the feed you create with the service. Here's what happened when I created a Feedburner feed:

1. I signed up for an account and went to <http://www.feedburner.com/fb/a/myfeeds>. I entered the URL of my weblog <http://blog.mashupguide.net>, instead of the URL of a feed.
2. Feedburner prompted me to choose a feed from among the five feeds associated with my weblog via the feed autodiscovery mechanism (described earlier in this chapter). I chose the Mashup Guide Atom Feed (<http://blog.mashupguide.net/feed/atom/>).

3. I accepted the defaults for the title ([Mashup Guide](#)) and address (<http://feeds.feedburner.com/MashupGuide>).

Feedburner has various features for customizing your feed:

- * You can customize the appearance of your feed in the browser. Feedburner attaches an XSLT style sheet to perform client-side transformation of the feed to HTML for a cleaner display of the feed in most browsers. For an example feed, you can explicitly see the HTML output using the W3C online XSLT service (http://www.w3.org/2005/08/online_xslt/) to generate this:

```
http://www.w3.org/2005/08/online_xslt/xslt?xslfile=
http%3A%2F%2Ffeeds.feedburner.com%2F%7Ed%2Fstyles%2
Fatom10full.xsl&xmlfile=http%3A%2F%2Ffeeds.feedburner.com%2FMashup
Guide+&content-type=&submit=transform
```

- * You can get traffic statistics for the feeds you create.
- * You can add tags from the iTunes or Media RSS extensions to your feeds to support podcasts.
- * You can splice your feed with your links from various social-bookmarking sites (including del.icio.us) or your photos from various photo-sharing sites (such as Flickr).
- * You can georeference your feed by having Feedburner attach the latitude and longitude of a given location to it.
- * You can convert your feed to one of RSS 2.0, RSS 1.0, Atom 0.3, or Atom 1.0.

I list these features here not to advertise Feedburner (it seems to do well enough for itself given its acquisition by Google) but rather to present it as a model so you can study the many ways in which others are remixing and mashing up feeds. In fact, Feedburner provides an API (<http://www.feedburner.com/fb/a/developers>), which suggests the high level of automation in place (or at least anticipated) for feeds.

Remixing Feeds with Yahoo! Pipes

Yahoo! Pipes (<http://pipes.yahoo.com/pipes/>) is a “an interactive data aggregator and manipulator that lets you mash up your favorite online data sources.” Yahoo! Pipes is focused on enabling end users to filter and combine feeds into new feeds. You construct pipes through dragging and dropping graphical widgets (called *modules*), entering parameters, and describing data flows through wiring these widgets together. Yahoo! Pipes is arguably more accessible to nonprogrammers because it does not involve typing code in a text editor. You’ll see in practice whether the masses will be making mashups with Yahoo! Pipes.

Note I will say that as a programmer, Yahoo! Pipes does make it easier to remix feeds in many instances and got me to create feeds that I could have created programmatically but was not inspired to do so without the Yahoo! Pipes environment.

In this section, I describe how I built a series of pipes to solve a specific problem. In doing so, I hope to shed light on how to think about Yahoo! Pipes, specifically how to construct increasingly more complicated structures. For the basics of Yahoo! Pipes, please consult the official documentation:

<http://pipes.yahoo.com/pipes/docs>

especially the documentation of the modules available in Yahoo! Pipes:

<http://pipes.yahoo.com/pipes/docs?doc=modules>

The problem I address with Yahoo! Pipes is creating a single feed from diverse news sources, unified around a single topic or search term. In constructing my pipes, I had a concrete scenario in mind. I wanted a feed that enables one to follow the latest news about the aftermath of Hurricane Katrina. Though I generalized my Yahoo! Pipes where I could easily do so, I am not attempting here to develop a comprehensive solution.

The solution I devised was to synthesize a feed out of the following four sources:

- * Yahoo! News
- * Google News
- * Wikinews
- * The national section of *The New York Times*

This range of new sources enables me to illustrate how to overcome some of the challenges you'll likely face when using Yahoo! Pipes.

A Simple First Pipe with Yahoo! News

The first step I took was to build a pipe to handle the first source, Yahoo! News. I exploited the fact that you can generate an RSS Yahoo! News for a search term with this:

<http://news.search.yahoo.com/news/rss?p={search-term}>

I built two versions of a pipe to return a feed for a given search term. The first version—called “Yahoo! News by Search Term (First Version)”—is here:

http://pipes.yahoo.com/pipes/pipe.info?_id=Rg_rh3NA3BGdECIe17okhQ

You can run it and view the source if you are logged in. You can run a pipe to get an RSS 2.0 feed for a given search term here:

http://pipes.yahoo.com/pipes/pipe.run?_id={pipe-id}&_render=rss&search_term={search-term}

For example, to search for Hurricane Katrina, go here:

http://pipes.yahoo.com/pipes/pipe.run?_id=Rg_rh3NA3BGdECIe17okhQ&_render=rss&search_term=Hurricane+Katrina

The pipe uses three widgets to enable a user to pass in a search term and return a feed of Yahoo! News (see Figure 4-3):

- * A **Text Input** module that takes the search term from the user and is wired to feed this term to the **URL Builder** described next
- * A **URL Builder** module that has a **Base** parameter of `http://news.search.yahoo.com/news/rss` and a query parameter of `p`
- * A **Fetch Feed** that fetches the feed at the URL coming from the **URL Builder**

Insert 858Xf0403.tif

Figure 4-3. Pipe for “Yahoo! News by Search Term (First Version)”. (Reproduced with permission of Yahoo! Inc. © 2007 by Yahoo! Inc. YAHOO! and the YAHOO! logo are trademarks of Yahoo! Inc.)

Google News and Refactoring Pipes

The second news source that I need to pull in is Google News, which returns an Atom feed for a given search term here:

`http://news.google.com/news?q={search-term}&output=atom`

I use `output=atom` instead of `output=rss` to show that Yahoo! Pipes can handle Atom feeds.

One way to build a module to handle Google News is to clone the one for Yahoo! News and change the parameters in the **URL Builder** module. Instead, because I figured that there are plenty of feeds with URLs that consist of a single parameter and search term, I decided to build a utility pipe that would return feeds at URLs in the following form:

`{base-URL}?{parameter_name}={parameter_value}{URL_suffix}`

Constructing such a pipe is equivalent to writing a reusable function. By contrast, cloning a pipe is analogous to copying and pasting code. When you use pipes seriously, you begin to see patterns that can be captured and reused in a pipe.

The pipe I constructed for that purpose (called “Feed from a URL Constructed from One Variable Parameter”) is located here:

`http://pipes.yahoo.com/pipes/pipe.info?_id=VolceXZA3BGkqcJZJx0y0Q`

Note the intermixing of the **URL Builder** and **String Builder** modules to concatenate parameters and build a URL that can actually be fed to **Fetch Feed** module.

I used that pipe and the fact you can write the URLs to retrieve feeds from Yahoo! News and Google News in terms of these four parameters (see Table 4-5).

Table 4-5. Parameters for Pipe Called “Feed from a URL Constructed from One Variable Parameter”

parameter_	parameter_	URL_base-URL	name	value	suffix
<code>http://news.google.com/news</code>	<code>q</code>	<code>{search-term}</code>	<code>&output=atom</code>		
<code>http://news.search.yahoo.com/news/rss</code>	<code>p</code>	<code>{search-term}</code>			

The pipe that accesses Yahoo! News using the utility pipe (entitled “Yahoo! News by Search Term”) is as follows:

http://pipes.yahoo.com/pipes/pipe.info?_id=5NhmMndA3BGg5zQ5n0artA

The “Google News by Search Term” pipe is here:

http://pipes.yahoo.com/pipes/pipe.info?_id=OKWv6nNA3BGkPtA8qWIyXQ

Wikinews and NY Times: Filtering Feeds

Now we come to the third and fourth sources: Wikinews, which has a single feed:

<http://feeds.feedburner.com/WikinewsLatestNews>

and the New York Times National News (I select the National News feed because of the focus on Hurricane Katrina):

<http://www.nytimes.com/services/xml/rss/nyt/National.xml>

In contrast to the Yahoo! News and Google News for which I can generate a feed for a given search term by constructing the appropriate URL, I use **search-term** to filter the feed. To that end, I use the **Filter** module in the Filter for Given Term in Feed Description and Title pipe here:

http://pipes.yahoo.com/pipes/pipe.info?_id=KIYSv3pA3BGg1oVbCB2yXQ

to pass along only those items in the feed whose description or title contains the search term.

With that pipe as a foundation, I construct the “Wikinews Filtered by Search Term” pipe here:

http://pipes.yahoo.com/pipes/pipe.info?_id=PA7iqHpA3BGb0AiVX0sBXw

and the NY Times National News Filtered by a Search Term pipe here:

http://pipes.yahoo.com/pipes/pipe.info?_id=yhBh7HxA3BGU_YRj1vC6Jw

Pulling the Feeds Together

With a pipe each for my four news sources, each of which takes a search term as input, I then create a pipe here:

http://pipes.yahoo.com/pipes/pipe.info?_id=q1Ukcn1A3BGew1NQjknRlg

that does the following:

- * Takes a search term as input and passes it to the individual news source pipes
- * Concatenates the individual feeds with the **Union** module
- * Gets rid of feed items that have the same link with the **Unique** module

You’re done. You can search for Hurricane Katrina in the four news sources here:

http://pipes.yahoo.com/pipes/pipe.info?Search_term=Hurricane+Katrina&_cmd=Run+Pipe&_id=q1Ukcn1A3BGew1NQjknRlg&_run=1

The corresponding feed is available here:

http://pipes.yahoo.com/pipes/pipe.run?Search_term=Hurricane+Katrina&

`_id=q1Ukcn1A3BGewlNQjknRlg&_render=rss`

Obviously, the pipes I created could be refined. Indeed, you can do so right now by going to the list of pipes I created for this chapter and cloning and modifying them for your own use:

<http://pipes.yahoo.com/pipes/tag.info?namespace=user&tag=mashupguide>

This section used feeds that are relatively easy to access. If you start to use tools such as Openkapow.com to screen-scrape new feeds, you have even more combinatorial possibilities for your mashups. Note that other tools that I discuss later in the book (such as the Google Mashup Editor) are able to consume feeds, such as those generated by Yahoo! Pipes and Feedburner.

Summary

Feeds provide the backbone of data exchange for mashups. Many web sites—including Flickr—generate feeds, particularly in the RSS and Atom formats. In this chapter, you learned about the ecosystem that has arisen around this plethora of feeds: news aggregators that gather feeds for reading, validators that help you produce good feeds, scrapers that let you generate feeds when a web site doesn't provide one, and remixing tools that let you generate elaborate mashups of feeds. In an extended example of mashups based on feeds, I showed how to use Yahoo! Pipes to generate a single topical news feed that is a mashup of several other appropriately filtered feeds.