

メタデータが既存ウェブサイトの付加価値を生む

# いま話題の RSSを解明する!

最近、RSSという単語を目にする機会が増えてきた。とんでもない事件が続発して人々が常にニュースを求めるようになる中、ウェブ上の最新情報を手軽に提供する標準フォーマットとして、RSSが注目を集めているのだ。

## RDFをベースにしたRSSによるサイト更新情報の仕組み

text: 神崎正英

### サイトの更新情報を伝える

RSSはもともとネットスケープが自社のポータルMy Netscapeに「チャンネル」を登録する手段として1999年に提案したものだ。その機能は一部では早くから利用されてきたが、ウェブログ用のツールがサイトの更新情報をRSSとして自動的に生成したり、ニューヨークタイムズやBBCといった海外の大手ニュースサイトが最新のヘッドラインをRSSで提供し始めたりして、この1年ほどで急速に脚光を浴びるようになってきた。

RSSは、サイトの概要とコンテンツのタイトルやURI(URIはURLの一般型)といったメタデータをXMLの形で提供する。利用者はRSSフィードリーダーと呼ばれるツールを使って、サイトから取りだしたRSSデータを表示するのが一般的だ。

RSSのフォーマットはシンプルで標準化されているため、多様な表示ツールが生まれている。メールソフト的な画面構成でたくさんのチャンネルを一括管理できるものもあれば、最新ヘッドラインを電光掲示板のように流すティッカーのようなものも

ある。自分のサイトのページにニュースサイトなどのRSSを取り込んで、好みのスタイルで表示するのも簡単だ。こうしたカスタマイズの容易さ、自由度の高さも、RSS人気の要因の1つだろう。

### チャンネルを記述する channel要素

RSSはRDF Site Summaryの略だ。実はほかの名称で呼ばれることもあるのだが、それについては135ページを参照)。その名のとおり、サイトの要約情報をRDF(Resource Description Framework)に則って記述する。ここでは、もっとも標準的なバージョン1.0の構成要素を見ていくことにしよう。

RSSは、以下のような2つの基本パーツを持つ。

サイトに関する情報やそのコンテンツのリストを提供するchannel要素

個別のコンテンツについて記述するitem要素

channel要素は「チャンネル」リソースを記述するRDFの「主語」にあたり、

rdf:about属性でチャンネル自身のURIを示す(以下、図1を参照)。要素内には、サイトのタイトル、URI、概要をそれぞれtitle、link、description要素として記述する。そのうえで、チャンネルとして取り上げるコンテンツのリストを、items要素の中に列記する(itemではなく複数形であることに注意)。

items要素内では、RDFの列記リスト構文であるrdf:Seq要素とrdf:li要素を用いて、コンテンツのURIを示す。これはXHTMLのul/li要素とよく似ているが、URIはrdf:li要素のrdf:resource属性値として記述するのが特徴だ(空要素となるので、'/>'でタグを閉じる)。

以上は必須要素で省略できない。このほかchannel要素には、オプションとしてimage要素を置き、サイトのロゴを示すこともできる。これもXHTMLのimg要素に似ているが、src属性ではなくrdf:resource属性を使って画像のURIを記述する。

### コンテンツの説明item要素

channel要素で列記したコンテンツの

具体的な紹介には、item要素を使う(こちらは単数形)、これはchannelの子要素ではなく、兄弟要素として記述する(以下、図2を参照)。

item要素の開始タグは、コンテンツのURIをrdf:about属性として持つ。このURIは、channel要素の一覧で示したrdf:resource属性の値と一致しており、チャンネルとコンテンツの対応関係がはっきりするわけだ。

item要素内でコンテンツを説明するには、channel要素の場合と同様、title、link、description要素を使う。RSSフィードリーダーなどのRSSアプリケーションは、通常これらの情報を使ってヘッドラインやハイパーリンク、要約を表示する。link要素の内容はitem要素のrdf:about属性と重複して無駄なのだが、過去との互換性のために必ず記述することになっている。

channel要素内にオプションのimage要素を置いた場合は、それに対応するimage要素を、やはり兄弟要素として記述し、説明を加えておく(図3)。rdf:about属性にリソースのURIを書き、title要素を加えるのはitem要素と同じだ。link要素も用いるが、ここでは画像のURIではなく、画像をクリック可能にしたときのリンク先となるURI(通常はサイトのURI)を書く。画像のURIは、やはり互換性のためにurl要素としても記述する。

## RDFとしてのRSS

channel、itemなどの要素は、全体としてrdf:RDF要素の中に収められる。そしてこの開始タグでは、RDFやRSSの名前空間とその接頭辞を宣言する。ここまで断りなしで使ってきたが、この記事ではRSSをデフォルト名前空間とし、RDFの方をrdf:という接頭辞で表してきたわけだ。必要に応じて冒頭にXML宣言を置き、これで1つのRSSファイルが完成する(図4)。

ここで、RSSのベースとなっているRDFについて振り返っておくことにしよう。RDFでは、リソース(主語)の情報をプロパティ(述語)とその値(目的語)という形

図1 RSSのchannel要素

```
<channel rdf:about="http://internet.impress.co.jp/im/rss.rdf">
  <title>iNTERNET magazine+</title>
  <link>http://internet.impress.co.jp/im/</link>
  <description>インターネットは2ndステージへ。インプレスがお送りする、
  インターネット情報誌の草分け「インターネットマガジン」のウェブサイト。
</description>
  <items>
    <rdf:Seq>
      <rdf:li rdf:resource="http://internet.impress.co.jp/im/bn/">
      <rdf:li rdf:resource="http://internet.impress.co.jp/winavi/">
      ...
    </rdf:Seq>
  </items>
  <image rdf:resource="http://internet.impress.co.jp/header/img02/logo.gif"/>
</channel>
```

channel要素内には、要約するサイトのタイトル、URI、概要などを記述する。channel要素のrdf:about属性には、このRSS自身のURLを記述する。

チャンネル(RSS自身)のURI

要約する対象サイトのURI

description要素に続けて、items要素で紹介するコンテンツを列挙する。rdf:Seq、rdf:li要素はそれぞれHTMLのul、li要素と同等だが、コンテンツのURIはrdf:li要素の内容ではなく、rdf:resource属性の値として記述する。

紹介するコンテンツのURIを列挙する

channel要素内には、オプションとしてサイトのロゴ画像を記述してもいい。画像のURIをrdf:resource属性値として示す。

図2 RSSのitem要素

```
<item rdf:about="http://internet.impress.co.jp/winavi">
  <title>Winavi+</title>
  <link>http://internet.impress.co.jp/winavi/</link>
  <description>モバイルer必読のワイヤレス情報満載メディア。ワイヤレス関連ニュース、無線LAN関連製品や無線アクセススポットの紹介など、無線LANのことがいろいろ分かります!
</description>
</item>
```

channel要素内のitems要素で列挙したコンテンツに対応して、それぞれのitem要素を記述する。rdf:about属性値とlink要素は同じURIで無駄なのだが、過去との互換性のためにこうなっている。

このコンテンツのURI

図3 RSSのimage要素

```
<image rdf:about="http://internet.impress.co.jp/header/img02/logo.gif">
  <title>iNTERNET magazine+</title>
  <link>http://internet.impress.co.jp/im/</link>
  <url>http://internet.impress.co.jp/header/img02/logo.gif</url>
</image>
```

channel要素内にimage要素を記述した場合は、それに対応するimage要素をchannelの兄弟要素として加える。titleは画像のalt属性、linkはクリック可能にした場合のリンク先となる。

ロゴ画像をクリック可能にした場合のリンク先URI

ロゴ画像のURI

で「文」として記述する。これをXMLで表現する場合は、主語リソースをrdf:about属性で示し、その子要素にプロパティを、さらにその内容として目的語を持つというサンドイッチ構造が基本だ。

channel要素を例に取れば、ここでの主語はrdf:aboutの値であるチャンネル自身。そのプロパティとしてtitle、link、descriptionなどがあり、その値はそれぞれの要素の内容になる。さらに主語要素がchannelなど一般の語彙で記述される場合は、リソースはそのタグ名の「型」を持つ(図5)。これを普通の言葉で表現すれば、「channel型のリソース」<http://internet.impress.co.jp>は、そのtitleとして「iNTERNET magazine+」を持つ」といった具合だ。

図4 RSSの構造

```
<rdf:RDF xmlns="http://purl.org/rss/1.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <channel
  rdf:about="http://internet.impress.co.jp/im/rss.rdf">
  ...
  </channel>
  <item ...>
  ...
  </item>
  <item ...>
  ...
  </item>
  <image ...>
  ...
  </image>
</rdf:RDF>
```

channel、item、image要素を兄弟要素として並べ、全体をrdf:RDF要素とする。そして、その開始タグでRDFやRSSの名前空間を宣言する。channel要素などを接頭辞なしで用いるため、RSSの名前空間をデフォルトにしている。

RDFの目的語が文字列ではなくリソースであるときは、rdf:resource属性を使ってそのURIを書く。チャンネルのコンテンツ一覧としてrdf:li要素に書き込んだものがそれにあたる。

目的語がURIを持つリソースならば、そのリソースを主語として別の「文」を続けることができる。item要素のrdf:about属性がrdf:li要素のrdf:resource属性と同じということは、図6のようにRDFの文が連結されることを意味しているのだ。このように、RSSはチャンネルをリソースのツリーとして表現しているのである。

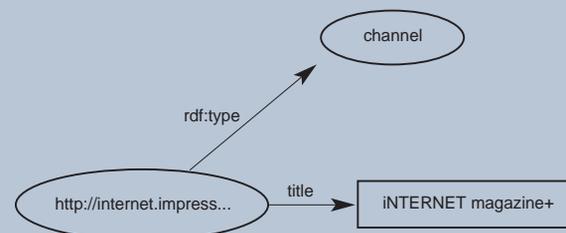
## 名前空間を使ったRSSの拡張

RSSをRDFに基づいて定義するメリットは、その汎用性と拡張性だ。RDFはもちろんRSS専用のフォーマットではなく、本誌3月号で取り上げたクリエイティブコモンズ(CC)のライセンス表現(UR)など、さまざまなメタデータ記述の場面で使われている。ということは、RDF用のツール(たとえばデータベースなど)でRSSをそのまま処理できることになるし、また名前空間宣言を加えて、CCのライセンスをRSSに含めることもできるわけだ。

特に有益なのは、コンテンツの作者や日付などを表すDublin Coreの語彙だろう。この名前空間をdc:という接頭辞と結びつけて宣言しておけば、たとえばchannel要素内にdc:creator要素を記述して作者を知らせたり、item要素にdc:date要素を加えてコンテンツごとの更新日時を明示したりできる(図7)。ニュースヘッドラインなどでは情報の鮮度に関するデータは重要だから、これらの標準的なプロパティが使えるメリットはとて大きい。

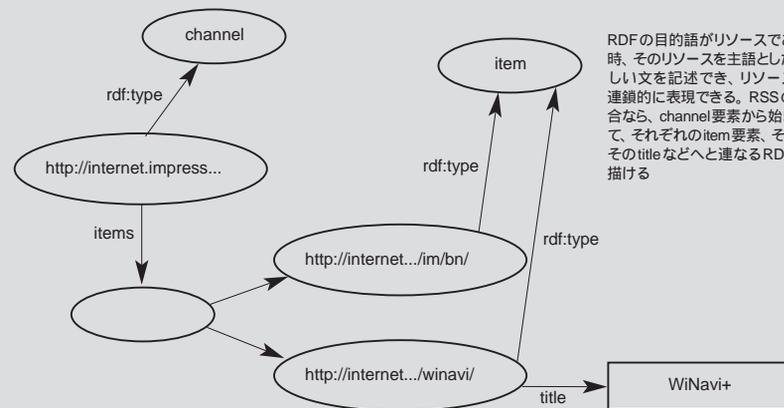
ウェブログツールMovableTypeのRSSテンプレートでは、このほかに配信(更新)頻度を示すsy:updatePeriod、XHTMLのタグも含めたコンテンツ本文を配信するためのcontent:encoded、エラーの報告先を示す

図5 RDFとしてのRSSを考える



channel要素は「rdf:about属性のURIを持つ「channel」型のリソースを主語とし、titleというプロパティの値として「iINTERNET magazine+」というリテラル文字列を持つ」というRDFのモデルで示される。

図6 連続されるRDFの構造



RDFの目的語がリソースである時、そのリソースを主語とした新しい文を記述でき、リソースを連鎖的に表現できる。RSSの場合なら、channel要素から始まって、それぞれのitem要素、そしてそのtitleなどへと連なるRDFが描ける

図7 Dublin Coreの名前空間が使える

```
<rdf:RDF
  xmlns="http://purl.org/rss/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <channel rdf:about="http://internet.impress.co.jp/im/rss.rdf">
    <title>iINTERNET magazine+</title>
    <dc:date>2003-04-25</dc:date>
    <link>http://internet.impress.co.jp/im/</link>
    ...
```

名前空間を追加することで、RSSのコア部分を全まったく変更することなく、日付などの新しい語彙を利用することが可能になる。

Dublin CoreのXML名前空間URIを追加

admin:errorReportsToといった語彙が取り込まれているので、これらも今後一般的になっていくだろう。

名前空間による語彙の拡張は、RSS自身の構文や語彙定義をまったく変更する必要がない。HTMLの独自拡張のような混乱に陥ることなく、安心して新機能を利用できるわけだ。

もちろんシンプルな情報を提供するの

なら、RSS 1.0の基本語彙だけで十分。新しい語彙を使わないとダメということはない。RSSがおもしろそうだと感じたら、まず手軽な形で自サイトの更新情報を公開してみよう。

クリエイティブコモンズとは

<http://internet.impress.co.jp/im/pdf/cc.pdf>

# RSSの歴史をたどってバージョンの複雑さを読み解く

RSS 1.0と2.0は名前も違えば仕様も違う

## RSSの誕生

RSSはサイトのメタデータを記述するフォーマットとして1999年に登場したが、そのルーツは1997年頃にまでさかのぼる。

当時アップルは、ニュースのチャンネルやサイトマップを記述するためのフォーマット MCF ( Meta Content Framework ) を公開していた。この技術をネットスケープが買い取り、XMLベースに改めてW3Cに提案する。これはその後、W3CでのRDF開発の基礎となるのだが、ネットスケープ自身もこれを新しいブラウザのインターフェイス定義に用いたり、ポータルサイトの情報提供に用いたりしようと考えた。こうして1999年3月に誕生したのが「チャンネル」を公開するためのRSSだったわけだ。この仕様は「RDF Site Summary ( RSS )」と呼ばれ、まだ発展途上であることを示すためにバージョンは0.9とされた。

RSS 0.9はごくシンプルにサイトの見出し一覧を記述するもので、提供される語彙はRSS 1.0の基本要素とほぼ同じだった。この仕組みはコンパクトで扱い

やすかったため、広く受け入れられて「チャンネル」記述のスタンダードに躍り出る。名前のとおりRDFの形を取り、名前空間も定義していた(ただし、rdf:about属性もなければchannel要素とitem要素の関係も不明で、RDFとしては中途半端だったのだが)。

## RSSの拡張と混乱、そして未来へ

ところで1997年と言えば、ポイントキャストというプッシュコンテンツの仕組みが注目を集めたり、Internet Explorerの4.0が「チャンネルパー」を携えて登場したりした時期でもある。このチャンネルで使われたCDF ( Channel Definition Format ) という規格を覚えている人もいるだろう。CDFは「プッシュ」が廃れるとともに次第に忘れられてしまったが、そこに盛り込まれていた更新時刻、配信頻度、著作権といった語彙は、チャンネル記述には重要なものでもあった。そこでUserLandというシンジケーションサービスのScriptingNewsがこの語彙を規格に適用し、さらにRSSもその影響を受けて、1999年7月にはバージョン0.91が誕生する。

RSS 0.91は、語彙が豊富になっただけでなく、RDFや名前空間は複雑すぎるという一部の主張を受け入れ、異なる構文の独自XMLとして再定義された。RDFではないので名称もRich Site Summaryと改められ、少々話がややこしくなってくる。

その後、ネットスケープはRSSのメンテナンスから手を引いてしまうのだが、チャンネル情報を提供する枠組みとしてのRSSへの要望はむしろ高まり、これに独自の要素型を加味して用いるユーザーも出現するなど、混乱の様相を見せてきた。そこで、RSS 0.9と同等のコア部分をもう一度RDFとして定義し直し、必要に応じて名前空間で語彙を拡張する(モジュールを組み込む)ようにしたRSSが、2000年12月に提案される。これが、現在のRSS 1.0だ。

長い目で見れば、ウェブ上のメタデータはRDFとして記述する方が応用の範囲が広く、セマンティックウェブにもつながっていく。だが、名前空間やRDFを受け入れたくない面々は、RSS 0.92、0.93などを提案して別の路線を歩んだ。さらに2002年8月には「RSS 2.0」を打ち出し、名前もReally Simple Syndicationだと主張するにいたっている(図8)。

もっとも、結局RSS 2.0は名前空間を導入して、外部の語彙の利用を可能にした(ただし後方互換性を理由にRSS 2.0のデフォルト名前空間は定義されていない)。したがって、両者の違いはRDFの枠組みを用いるかどうかということに絞られてきている。

最新のウェブログツールなどは、RDFを採用したRSS 1.0とRSS 0.91/2.0のどちらの書き出しにも対応しているようだ。汎用性と将来性を考えれば、RSS 1.0を選択しておくのが賢明だろう。

図8 RSS 2.0の記述

```
<rss version="2.0">
<channel>
<title>INTERNET magazine+</title>
<link>http://internet.impress.co.jp/im/</link>
<description>インターネットは2ndステージへ。...</description>
<language>ja-JP</language>
<pubDate>2003-04-25</pubDate>
<item>
<title>WiNavi+</title>
<link>http://internet.impress.co.jp/winavi/</link>
<description>モバイル必読の...</description>
</item>
<item>
...
</item>
</channel>
</rss>
```

サンプルのRSSをバージョン2.0で記述してみたもの。ルート要素がrssであること、channel要素の子要素としてitem要素を用いる点が大きく異なる。コンテンツの発行日や言語を独自の要素として記述できるが、これらの代わりにDublin Coreの名前空間を宣言してdc:date、dc:languageなどを用いることもできる。

# 独自アプリケーションを大活用! RSSはこう使おう

text: 増田(maskin)真樹 / フリージャーナリスト・情報デザイナー

URL <http://metamix.com/>

## RSSを RSSフィードリーダーで読む

RSSフィードで提供されるXMLファイル(.rssや.rdf、.xml)は、ウェブブラウザでは意図したようには表示されない。そこで専用のアプリケーションのRSSフィードリーダーを使えばRSSというメタデータを活用できる。

RSSフィードリーダーの最大の利点は、複数のニュースサイトの情報を一度に閲覧できるところだ。操作はどのアプリケーションも簡単で、RSSが提供されているURL(RSSフィード)を指定するだけでいい。記事本文を読みたい場合は、RSSフィードリーダーから簡単にウェブサイトにジャンプできる。ほとんどのアプリケーションが、RSS 0.91と1.0に対応しているが、それ以外のバージョンには未対応のことが多い。

日本語に対応したビューアーで人気の

のはウィンドウズならその名も「FeedReader」で、Mac OS Xなら「NetNewsWire」だ。特にNetNewsWireは、最初からかなりの数のRSSフィードへのインデックスが用意してあったり、ウェブログに書き込みができたりするなど機能が豊富な。

このほかJavaで記述された「HotSheet」というアプリケーションもあり、これならプラットフォームを問わずにRSSフィードを見られる。現在は英語しか表示できないのが残念だ。

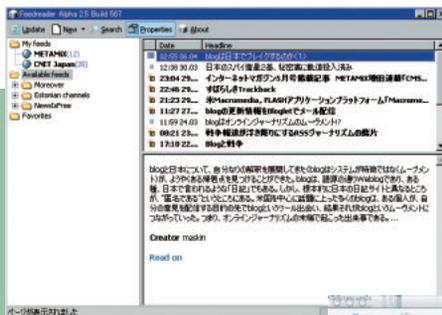
## 日本語で読める RSSフィードを探す

RSSフィードは、すでに英語を中心とした世界中のメジャーなニュースサイトで行われている。たとえば英国の「BBCニュース」[URL01](#)や米国の人気コラムサイト「Salon」[URL02](#)などを筆頭に数多くのサイ

トがRSSを採用している。

しかし、日本国内のニュースメディアによるRSSフィードは、“これから立ち上がるかもしれない”という段階で、正式にRSSを提供しているところは「CNET JAPAN」くらいで、実質的に皆無に等しいのが現状だ。しかし独自にニュースをRSSに変換して配信する実験を行っている「rss-jp.net」や「Bulknews」からのRSSフィードを受信すれば、さまざまな日本語のニュースをRSSフィードリーダーで見られるようになる。

一方、これだけRSSが話題になっている理由として、ウェブログが自動的にRSSを生成しているということが挙げられる。たとえばウェブログツールのMovableTypeでは標準でRSS 0.91と1.0が生成される。したがってMovableTypeを使っている日本語サイトはほとんどRSSを提供していると言っていいだろう。



FeedReader  
URL <http://www.feedreader.com/>  
ご覧のとおり日本語にも対応。フォルダを作ってカテゴリーごとに管理できる。ウィンドウズ95/98/Me/2000/XPに対応。フリーウェア。



NetNewsWire  
URL <http://ranchero.com/>  
筆者いち押し RSSフィードリーダー。ウェブログへの書き込み(ポスト)が可能 XML-RPC対応。残念ながら日本語の入力はIMとの相性が悪い。Mac OS Xに対応。ドネーションウェア。そのほかメモなど機能は豊富。



HotSheet  
URL <http://www.johnmunsch.com/projects/HotSheet/>  
Javaで作られたRSSフィードリーダー。ウィンドウズ、マッキントッシュ、UNIXで動作する。フリーウェア。

## アグリゲーションサービスで 利用される

最後に、RSS アグリゲーションサービスを紹介しよう。「NewsIsFree」[URL03](http://www.newsisfree.com/)は自分のアカウントを作成して、好きなRSSフィードのURLを登録しておけば、それをあ

たかもポータルサイトのようにまとめて1つのウェブページに表示してくれるサービスだ。ウェブブラウザさえあれば、どこからでも自分専用にカスタマイズしたニュースのリストページが表示できる。またNewsIsFreeは世界中のRSSフィードのデータベースを持っているので、新しいニュー

ースを探すのにも手がかからない。残念なのは現在のところ日本語のサイトだとUTF-8でも文字化けをしてしまうという点だろう。今後の対応が期待される。

[URL01](http://www.bbc.co.uk/syndication/feeds/news/) http://www.bbc.co.uk/syndication/feeds/news/

[URL02](http://www.salon.com/) http://www.salon.com/

[URL03](http://www.newsisfree.com/) http://www.newsisfree.com/

### 日本国内でRSSを提供しているサイト

名前	URL	内容
CNET JAPAN	<a href="http://japan.cnet.com/">http://japan.cnet.com/</a>	国内唯一(4/8時点)正式にニュースをRSSでフィードするサイト。
Slashdot Japan	<a href="http://slashdot.jp/code.shtml">http://slashdot.jp/code.shtml</a>	テクノロジー系コミュニティの最新テーマをRSSでフィードしている。
jss-jp.net	<a href="http://www.rss-jp.net/">http://www.rss-jp.net/</a>	実験的にさまざまなニュースサイトやコミュニティサイトのRSSフィードを行うサイト。2ちゃんねるや開心空間のフィードも行う。
Bulknews	<a href="http://www.bulknews.com/">http://www.bulknews.com/</a>	30分ごとに沢山のニュースサイトを巡回しRSSフィードを行っている。
eグループ	<a href="http://www.egroups.co.jp/">http://www.egroups.co.jp/</a> <a href="http://www.egroups.co.jp/messages/グループ名/messages?rss">http://www.egroups.co.jp/messages/グループ名/messages?rss</a>	公開グループのみ投稿メッセージをRSSで受信できる。

## COLUMN

# 自分のサイトのウェブページにRSSフィードを表示する

アプリケーションを使ってサイトの最新情報を盛り込む

RSSを自分のウェブページから表示させれば「NewsDigest」のような、自分だけのニュースポータルサイトを作ることだって可能だ。それにはパーサーとよばれるRSS解読プログラムが必要だが、導入は簡単なのでぜひためしてみよう。

NewsDigest

[URL](http://www.d-sys.co.jp/news/) <http://www.d-sys.co.jp/news/>

## PHPが使える ホスティングの場合

PHPが使えるサイトなら「MagpieRSS」というパーサーが無償で提供されているのでこれを使うのがいちばん手っ取り早いし、いろいろと加工もできる。インストール後、右のようなプログラムで表示できる。

MagpieRSS

[URL](http://magpierss.sourceforge.net/) <http://magpierss.sourceforge.net/>

## ウェブログの場合

ウェブログツールのMovableTypeならプラグインで自分のウェブログにRSSフィードを表示できる。プラグインをインストールしたあとRSSフィードごとにテンプレートを作ればOKだ。

mt-rssfeed

[URL](http://www.mplode.com/tima/files/mt-plugins/#mt-rssfeed) <http://www.mplode.com/tima/files/mt-plugins/#mt-rssfeed>

## それ以外の場合

なんとFLASHで、RSSを電光掲示板風に表示できるツールもある。「C-TICKER」はJavaとFlashでRSSフィードを自分のページに表示できる。HTMLに組み込んでアップロードすれば、フリーのウェブサービスでも利用できる。

C-TICKER

[URL](http://www.nk.rim.or.jp/~t_kimata/cgi/soft/c-ticker/) [http://www.nk.rim.or.jp/~t\\_kimata/cgi/soft/c-ticker/](http://www.nk.rim.or.jp/~t_kimata/cgi/soft/c-ticker/)

### PHPで記述されたRSSパーサー「MapieRSS」

```
<?php
require_once 'rss_fetch.inc';           MapieRSSのライブラリーを指定する
$url = 'http://www.metamix.com/index.rdf'; 取得するRSSのURLを指定する
$rss = fetch_rss($url);
$title = $rss->channel['title'];
$title = mb_convert_encoding($title, "utf-8", "auto");  サイト名を取得して表示する
echo "title n";
foreach ($rss->items as $item) {
    $title = $item['title'];
    $title = mb_convert_encoding($title, "utf-8", "auto");
    $url = $item['link'];
    echo "<a href= \"$url\">$title</a><br> n";  記事のタイトルとURLを取得して表示する
}
?>
```

備考: 文中の utf-8 は、rss の文字コードに合わせて SJIS や EUC-JP に書き換える必要があります。



## [インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

**株式会社インプレスR&D**

All-in-One INTERNET magazine 編集部

[im-info@impress.co.jp](mailto:im-info@impress.co.jp)