Creating Really Simple RSS from a Database

Do you have news that you want to spread around the Web on a regular basis (for example, press releases, product reviews, or other such information)? If so, then maybe RSS is for you.

What is RSS? RSS stands for "Really Simple Syndication" or "RDF Site Summary" depending on the precise version being used. Both names are appropriate.

The first name, "Really Simple Syndication" applies primarily to versions referred to as 0.9x (for example, 0.91, 0.92, etc.). It takes its name from the fact that it uses a fairly basic XML file to provide a list of headlines, descriptions, and links (the "Really Simple" part) and places this XML file (called an RSS feed) on a Web site where it can be accessed by other Web sites or dedicated RSS readers (the "Syndication" part). These sites and readers often collect feeds on a regular basis (for example, once every five minutes) from various sources and present the collected information to users. This lets users keep up-to-date with new information on multiple Web sites without having to visit each one individually to see if anything's new.

The second name, "RDF Site Summary" applies primarily to version 1.0, which is a formal standard and uses the RDF standard to help define it and hence, the different name. In truth, though, RSS 1.0 is just an enhanced version of the 0.9x versions which allows for the addition of RSS modules which can define additional tags to let feeds carry extra information (for example, dates and authors). As such, the "Really Simple Syndication" label could be just as easily applied to it. And to show how simple it is to create an RSS feed, we're going to use the MKS Toolkit utilities to read information from a database and turn it into a basic RSS 1.0 feed.

Why are we using RSS 1.0 instead of RSS 0.9x? First of all, 1.0 is a W3C (World Wide Web Consortium) standard. Second, 1.0 feeds are similar to 0.9x feeds, but have slightly more complexity and thus, by showing you the techniques to create a 1.0 feed, it should be easy enough for you to apply those same techniques to create the simpler 0.9x feed.

ACCESSING THE DATABASE

So what's in this database that we're going to turn into a RSS feed? For this example, we'll be using a Microsoft Access database named `press_releases.mdb`. It contains information on actual MKS Inc. press releases. To be precise, it contains a table named `Releases` which, for each press release, has five fields: `Date`, `Title`, `Description`, `Link`, and `Live`. These fields, respectively, indicate the date of the press release, its headline, its first paragraph, the link to where it can be found on the MKS Web site, and whether or not the release has gone live (that is, released to various media sources).

Our overall task is to retrieve the information for the three most recent live press releases from this database and put that information into an RSS feed.

The first step in this task is to retrieve the press release information. Fortunately, MKS Toolkit provides the **db** utility. This utility is designed specifically to get information from a database using SQL (Structured Query Language). Before **db** can access a database, a data source needs to be created for that database. This data source provides the drivers needed for SQL to properly access the desired database. For our purposes, we are going to assume that such a data source (named `press_releases.dsn`) has already been created for `press_releases.mdb`. The basic format of the **db** command for accessing this database is:

```
db -d press_releases.dsn sql_command
```

where *sql_command* is the SQL command to be used on the database.

But what should this SQL command be? Since RSS feeds use a headline, description, and link for each item in the feed, we need to retrieve this information for each release we want to include. Also, we only want the three most recent live releases. Thanks to the power of SQL, we can do this with a single command:

```
SELECT TOP 3 Title, Description, Link FROM Releases WHERE Live=1 ORDER BY Date DESC
```

Basically, this command sorts the `Releases` table in the database in descending order by the `Date` field (that is, the most recent press releases will appear first) and selects first three live press releases (`Live` is a Yes/No field in the database which returns 1 for Yes and 0 for No) and returns the `Title`, `Description`, and `Link` fields from those releases. We can now use this SQL command with the **db** utility:

```
db -d press_releases.dns " SELECT TOP 3 Title, Description, Link FROM Releases WHERE
Live=1 ORDER BY Date DESC"
```

When issued from the command line, this command displays the three fields for each selected release. The information is displayed as one line per release with the fields separated by a Tab character. *Figure 1: db Output* shows the results when we apply this command to our database of MKS Inc. press releases. Note that this output also includes a line at the top identifying the selected fields.

```
Title <<<Tab>>> Description <<<Tab>>> Link

TEVA Pharmaceuticals USA to Standardize on MKS Integrity Solution to Meet
FDA Audit Requirements <<<Tab>>> MKS Inc. (TSX:MKX), a leading provider of
enterprise software configuration management (SCM) solutions for the Global
1000, today announced that TEVA Pharmaceuticals USA, the wholly-owned
American subsidiary of TEVA Pharmaceutical Industries Ltd. (NASDAQ: TEVA)
and the leading manufacturer and distributor of generic drugs in the U.S.,
has selected the MKS Integrity Solution for enterprise software
configuration management (SCM). [more] <<<Tab>>>
#http://www.mks.com/press/index.jsp?action=readarticle&article_id=6875#

MKS Guarantees Success on Mission-critical UNIX-to-Windows Migration
Projects Using MKS Toolkit for Enterprise Developers <<<Tab>>> MKS Inc.
(MKS) (TSE:MKX), the leading provider of tools for porting UNIX and Linux
applications to Windows, today launched the Guaranteed Success program which
provides a free, hour-long, scheduled porting consultation for any MKS
Toolkit for Enterprise Developers Evaluator, as well as 3 additional free
hour-long, scheduled consultations with MKS porting experts after license
purchase. In addition, MKS is so confident customers will succeed through
this program that they are now offering a partial refund on all MKS porting
tools licensed if the migration is not successful within 90 days. [more]
<<<Tab>>>
#http://www.mks.com/press/index.jsp?action=readarticle&article_id=5855#

Magellan Health Services to Standardize on MKS for Organizational IT Change
and Sarbanes-Oxley Compliance <<<Tab>>> MKS Inc. (MKS) (TSX:MKX), a leading
provider of enterprise software configuration management (SCM) solutions for
the Global 1000, today announced that Magellan Health Services Inc., the
United States leading managed behavioral health care organization, has
signed an enterprise agreement to purchase MKS's software configuration
management solution. Magellan will implement MKS's enterprise SCM solution
to provide greater control, visibility, and auditability across its
information technology (IT) organization, aiding the company in its business
goal to establish a foundation for secure organizational IT change
management and Sarbanes-Oxley and HIPAA compliance. [more] <<<Tab>>>
#http://www.mks.com/press/index.jsp?action=readarticle&article_id=5601#
```

Note: Tab characters are shown as **<<<Tab>>>** for clarity.

Figure 1: db Output

Now that we know how to retrieve the desired information, we can move on to processing the information we have retrieved and turning it into a RSS feed.

GENERATING THE RSS FEED

As we saw in the previous section, the output from the **db** command consists of a heading line followed by one line for each of the selected releases with individual fields separated by tabs. Fortunately, MKS Toolkit includes the perfect utility for processing data in this form: **awk**. The **awk** utility is actually a scripting language that processes input one line at a time, automatically breaking that line down into individual fields based on a definable separator. So, if we create an **awk** script that uses tabs as the field separator, we can just feed in the output from the **db** utility and **awk** will automatically understand that each line has three separate fields. It can then store these individual fields in arrays and once all the output has been processed, it can use these arrays to generate the actual RSS feed.

Figure 2: MKS KornShell Script for Generating RSS Feed shows the final script with the lines numbered for easy reference. The script basically uses the **db** command described earlier and pipes the output from that command into the **awk** script which generates the RSS feed which is stored in the file `mks_press_releases.rss`.

Let's take a look at the guts of the **awk** script. Like all **awk** scripts, it has three sections: the `BEGIN` section which is run before it starts processing input, the `END` section which is run after all the lines in the input have been processed, and the

```
1  db -d press_releases.dsn "SELECT TOP 3 Title, Description, Link FROM Releases WHERE
   Live=1 ORDER BY Date DESC" |
2  awk '
3    BEGIN {
4      FS="\t"
5      n=0
6    }
7    NR > 1 {
8       n++
9       Title[n]=$1
10      Description[n]=$2
11      Link[n]=$3
12      sub("^#","",Link[n])
13      sub("#$","",Link[n])
14    }
15    END {
16      print "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>"
17      print "<rdf:RDF xmlns:rdf=\"http://www.w3.org/1999/02/22-rdf-syntax-ns#\""
18      print "           xmlns=\"http://purl.org/rss/1.0/\">"
19      print "<channel rdf:about=\"http://www.mks.com/press/index.jsp?action=history\">"
20      print "<title>MKS Press Releases</title>"
21      print "<link>http://www.mks.com/press/index.jsp?action=history</link>"
22      print "<description>The Latest News from MKS Inc.</description>"
23      print "<items>"
24      print "  <rdf:Seq>"
25      for (i=1;i<=n;i++) {
26        print "  <rdf:li resource=\"" Link[i] "\" />"
27      }
28      print "  </rdf:Seq>"
29      print "</items>"
30      print "</channel>"
31      for (i=1;i<=n;i++) {
32        print "<item rdf:about=\"" Link[i] "\">"
33        print "  <title>" Title[i] "</title>"
34        print "  <link>" Link[i] "</link>"
35        print "  <description>" Description[i] "</description>"
36        print "</item>"
37      }
38      print "</rdf:RDF>"
39    }' > mks_press_releases.rss
```

Figure 2: MKS KornShell Script for Generating RSS Feed

main body which is executed for each line in the input. Here, the `BEGIN` section (lines 3-6) simply sets the field separator for the input to the tab character in line 4 (`\t` is how you specify a tab character in **`awk`**) and sets the counter for the number of releases read (n) to 0. Why do we need such a counter if we have specifically asked for three releases to be included? First, it makes easier to change the number of desired release because we would only need to change the **db** command in line 1. Second, it allows for the possibility that there might be less than three live press releases in the database; in which case, the **db** command simply returns all of the live press releases.

The main body of the script (lines 7-14) would normally be run for each line of input; however, in this case, the opening curly brace for the section is preceded by the `NR>1` conditional. The internal **`awk`** variable `NR` is equal to the number of lines read thus far (counting the current one). Attaching this conditional to the main body means that the first line of input (in this case, that's the heading with the field names) is effectively ignored. For all other lines in the input, the release counter (n) is incremented, and the three fields of that line, known to **awk** as $1, $2, and $3, are added to the arrays `Title[]`, `Description[]`, and `Link[]`. But that's not all that needs to be done here. If you look carefully at *Figure 1: db Output*, you'll notice that the Link fields retrieved from the database have a # character placed at the beginning and end of the link. We neither need nor want these characters in our RSS feed, so lines 13 and 14 remove them.

When the last line of input has been read and processed, the `END` section is executed. This is where the actual RSS feed is produced and output. Why is it done here and not as we process each line of input? The reason is quite simple. The RSS 1.0 standard calls for the `<channel>` tag that precedes the individual items to include a list of all the links used specified in those items. As a result, before we can actually output the first part of the RSS feed, we need to know exactly what links the feed contains. By storing the contents of the input lines in arrays and saving the output operation until the END section, we can easily accomplish this.

Getting back to the script itself, lines 18-20 output the information needed for all RSS 1.0 feeds. If we were using any RSS modules to extend the data contained in the feed, we would specify them here. You should also note the use of `\"` to include a double quote within the **`awk print`** statement. Lines 21-34 print out the `<channel>` tag and its contents which describe the feed itself. Within this chunk code is lines 28-30 which loop through our `link[]` array and output the list of links discussed above.

The next portion of the script, lines 36-42, loops through our three arrays and outputs the `<item>` tag and its contents for each item in the feed. Finally, line 44 outputs the `</rdf:RDF>` tag to close the `<rdf:RDF>` tag opened back in line 19 and line 45 ends our **awk** script and redirects the output it generated into the file `mks_press_releases.rss`. *Figure 3: The RSS Feed* shows the file generated by this script when applied to the **db** output shown in *Figure 1*.

```
<?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.mks.com/press/index.jsp?action=history">
<title>MKS Press Releases</title>
<link>http://www.mks.com/press/index.jsp?action=history</link>
<description>The Latest News from MKS Inc.</description>
<items>
  <rdf:Seq>
  <rdf:li resource="http://www.mks.com/press/index.jsp?action=readarticle&article_id=6875" />
  <rdf:li resource="http://www.mks.com/press/index.jsp?action=readarticle&article_id=5855" />
  <rdf:li resource="http://www.mks.com/press/index.jsp?action=readarticle&article_id=5601" />
  </rdf:Seq>
</items>
</channel>
<item rdf:about="http://www.mks.com/press/index.jsp?action=readarticle&article_id=6875">
  <title>TEVA Pharmaceuticals USA to Standardize on MKS Integrity Solution to Meet FDA Audit
Requirements</title>
  <link>http://www.mks.com/press/index.jsp?action=readarticle&article_id=6875</link>
  <description>MKS Inc. (TSX:MKX), a leading provider of enterprise software configuration management (SCM)
solutions for the Global 1000, today announced that TEVA Pharmaceuticals USA, the wholly-owned American
subsidiary of TEVA Pharmaceutical Industries Ltd. (NASDAQ: TEVA) and the leading manufacturer and distributor of
generic drugs in the U.S., has selected the MKS Integrity Solution for enterprise software configuration
management (SCM). [more]</description>
</item>
<item rdf:about="http://www.mks.com/press/index.jsp?action=readarticle&article_id=5855">
  <title>MKS Guarantees Success on Mission-critical UNIX-to-Windows Migration Projects Using MKS Toolkit for
Enterprise Developers</title>
  <link>http://www.mks.com/press/index.jsp?action=readarticle&article_id=5855</link>
  <description>MKS Inc. (MKS) (TSE:MKX), the leading provider of tools for porting UNIX and Linux applications
to Windows, today launched the Guaranteed Success program which provides a free, hour-long, scheduled porting
consultation for any MKS Toolkit for Enterprise Developers Evaluator, as well as 3 additional free hour-long,
scheduled consultations with MKS porting experts after license purchase. In addition, MKS is so confident
customers will succeed through this program that they are now offering a partial refund on all MKS porting tools
licensed if the migration is not successful within 90 days. [more]</description>
</item>
<item rdf:about="http://www.mks.com/press/index.jsp?action=readarticle&article_id=5601">
  <title>Magellan Health Services to Standardize on MKS for Organizational IT Change and Sarbanes-Oxley
Compliance</title>
  <link>http://www.mks.com/press/index.jsp?action=readarticle&article_id=5601</link>
  <description>MKS Inc. (MKS) (TSX:MKX), a leading provider of enterprise software configuration management
(SCM) solutions for the Global 1000, today announced that Magellan Health Services Inc., the United States
leading managed behavioral health care organization, has signed an enterprise agreement to purchase MKS's
software configuration management solution. Magellan will implement MKS's enterprise SCM solution to provide
greater control, visibility, and auditability across its information technology (IT) organization, aiding the
company in its business goal to establish a foundation for secure organizational IT change management and
Sarbanes-Oxley and HIPAA compliance. [more]</description>
</item>
</rdf:RDF>
```

Figure 3: The RSS Feed

CONCLUSION

Now that we have generated the RSS feed, we can copy the `mks_press_releases.rss` file to an appropriate Web site. Now any application with RSS-reading capability or a dedicated RSS reader can just point to the file's URL and display our RSS feed. To make sure that the feed stays up-to-date, we can use the MKS Toolkit Scheduler to run the MKS KornShell Script shown in *Figure 2* along with the operation of placing the feed file generated on the Web site on a regular basis (for example, every 10 minutes).

So there it is. With just a few tools from MKS Toolkit, you can create an RSS feed from information in a database that lets you share this information with the rest of the World Wide Web. And by scheduling the feed to be regenerated on a regular basis, it is easy for readers of this feed to see when new information is available.

For more information about the MKS Toolkit products please visit http://www.mkssoftware.com and to view the full reference pages for the commands mentioned in this document visit http://www.mkssoftware.com/docs/cmd_index.asp.