Report Mining with TextPipe Pro

Background

This document is a guide to the general principles of mining data from reports using TextPipe Pro, our report mining solution.

Report mining is the process of extracting useful data from unstructured reports. Typically these reports are designed for humans to read, and have headers, footers and page numbers, column titles, totals etc. If your data is already in CSV, tab-delimited, fixed width, HTML or XML form, see our other online documents (http://www.datamystic.com/docs) for data mining web sites and working with restrictions.

Report mining allows you to

- **Take advantage of existing reporting systems already in place. You can make use of reusable data from reports on hand such as monthly, weekly or daily reports on topics such as collected receivables or bills paid**
- **Avoid the complex programming required to generate a report**
- **Avoid re-applying the same complex business rules as was used to generate the report**
- **Extract data from inaccessible systems. Because plain ASCII "line printer" text is a "lowest common denominator" standard of output offered by virtually every operational system, report mining tools can be utilized in virtually any computing environment. The report mining tool does not need to "know" what databases or systems the reports came from.**

General Principles

There are four key stages to report mining

1. **Obtain the report**
2. **Throw away unnecessary data**
3. **Reformat and extract desired data**
4. **Load into a database**

1. Obtain the report

The first stage to report mining is to obtain a copy of the report.

This may involve the use of a screen scraper in a terminal emulator such as SecureTelnet, saving data sent to a line printer, transferring files from a mainframe or unix server etc.

If the report is from a mainframe or unix server, you may need to first convert it to a useable form. See our working with mainframe reports white paper (http://www.datamystic.com/docs).

2. Throw Away Unnecessary Data

A typical report consists of many things we don't need – headers and footers, page numbers, column titles, totals etc. The first stage is to remove all these things.

In general, we use the Remove Matching Lines filter, to remove lines matching a pattern. Common patterns include:

To Remove lines containing these...	Use this EasyPattern...
Page numbers	Page [1 or more digits]
---------- lines of dashes	[20+ '-']
Report header "my report"	my report
A date formatted as mm/dd	[month, '/', day]

For more examples see the EasyPattern reference in the TextPipe help file.

We also tend to

1. **Remove blanks from the start of each line**
2. **Remove blanks from the end of each line**
3. **Remove blank lines**
4. **Remove multiple whitespaces**

You can link to our standard filter in report extraction\generic data miner.fll:

1. **Open TextPipe**
2. **On the File Menu, click Link to filter...**
3. **Enter C:\Program Files\DataMystic\TextPipe\report extraction\generic data miner.fll**
4. **A link to the existing filter is created. You can open the linked filter by clicking the Open button beside it.**

3. Reformat and Extract Desired Data

In general, a pattern match is used to extract the content and then reformat to the desired output format, generally a comma- or tab—delimited output file.

Let's say we had data like this:

TI: Quarantine host range studies with Lophyrotoma zonalis, an Australian sawfly of interest for

biological control of melaleuca, Melaleuca quinquenervia, in Florida

AU: Buckingham, GR

JN: Biocontrol

PD: 2001

VO: 46

NO: 3

PG: 363-386

This can be matched with the EasyPattern:

TI: [capture(1+ not CR or LF)]

AU: [capture(1+ not CR or LF)]

JN: [capture(1+ not CR or LF)]

PD: [capture(1+ not CR or LF)]

VO: [capture(1+ not CR or LF)]

NO: [capture(1+ not CR or LF)]

PG: [capture(1+ not CR or LF)]

Each captured field (numbered 1 through 7) can be output in the replacement result like this:

Num[0] Date[$4]

Author[$2]

Title[$1]

Citation[$3 $5($6):$7]

Comment[]

Here we have reformatted and changed the order of the original data.

More report mining examples can be found in the report extraction folder.

3b. An Alternative Approach

If the data is not regular, first tag each item you want to extract, and then throw away everything else.

1. **Use one or more patterns to match the data you need. Output (replace) the found data with the 'tag' characters ### at the start of the line. E.g.**

   Find EasyPattern: Price: $[capture(1 + digits, '.', 2 digits)]
   Replace with: \r\n###$1\r\n

2. **Throw away all lines not starting with ###, ie *Filters\Remove\Remove lines\Remove non-matching lines*, with a Pattern:**

   ^###

3. **If the fields do not always occur in the same order, instead of ### use ###1, ###2, ###3 etc, and use *Filters\Special\Sort* to sort based on the first 4 characters.**

4. **Finally, we remove the leading ### characters (or ###1, ###2, ###3 etc) using *Filters\Remove\Columns* to remove columns 1 to 3 (or 4).**

Now we have all the data, and we just need to convert it to CSV or XML output. We can do this with one whopper search/replace like this:

Find EasyPattern:
[capture(0+ not cr or lf), cr, lf,
capture(0+ not cr or lf), cr, lf,
capture(0+ not cr or lf), cr, lf,
capture(0+ not cr or lf), cr, lf,
...
capture(0+ not cr or lf), cr, lf]

Replace with:
"$1","$2","$3","$4"..."$36"
or
<row value1="$1" value2="$2" value3="$3" ... value36="$36" />

4. Load into a Database

Once you have extracted and manipulated the data into a Comma Separated Value format, you can easily load it into a database or into Excel. If the output file extension is .csv, then double-clicking it will automatically open it in Excel.

To load the data into a database, you need to first prepare a database with a table in the right format. To load the data into a database, you need to first prepare a database with a table in the right format

(alternatively, you can use a freeform database such as www.asksam.com to avoid having to define a precise database structure, as well as getting very powerful searching and reporting functions for free).

Then you need to manipulate the data into the form

Insert into tablename (field1,field2,field3...) values (value1,value2,value3...);

See the example filters in the C:\Program Files\DataMystic\TextPipe\database folder.

Feedback and Questions

If you have feedback or questions about this documentation, please contact us.

We can also send you updated sample filters from this article, or sample filters tailored to your data processing needs.

More White Papers and Documentation like This

Available from:

www.datamystic.com/docs

TextPipe Pro Evaluation

You can download a free 30 day trial of TextPipe Pro from

www.datamystic.com/textpipe-wp.exe

You can also access our other downloads from

www.datamystic.com/freetrials.html

Please contact us if you have any questions, difficulties or queries.

Contact Details

DataMystic

5 Bond Street

Mt Waverley

Victoria 3149

Australia

Web site: www.datamystic.com

Phone: +61-3 9913-0595

Fax: +61-3 8610-1234