

TEXTPIPE RESTRICTIONS -

WHAT ARE THEY AND HOW DO I USE THEM?

What is a restriction?	1
Creating Subfilters.....	1
Combining Restrictions.....	2
Restrict to line and column ranges	3
Restrict to CSV and Tab-delimited field ranges.....	3
HTML and XML restrictions.....	3
Restricting to sections of a .INI file	3
Restrict to lines matching/not matching a pattern	4
Restrict to the filename.....	4
Restrict to each line in turn.....	4
Restrict to lines with X Tab fields/CSV fields	4
Restrict to blocks of length XXX	5
Splitting log files	5
Feedback and Questions	7
More White Papers and Documentation Like This	7
TextPipe Pro Evaluation.....	7
Contact Details.....	7

What is a restriction?

TextPipe Restrictions are one of its most powerful features. Put simply, they allow you to control exactly **where** a text modification occurs. An example of a restriction could be lines 1 to 10, or columns 3 to 17, or only lines matching a pattern. For example, if you need to perform a search and replace in only the very first line of a file (e.g. the header line of a CSV or tab-delimited file), you could do it like this:

```
|--Restrict lines:Line 1 .. line 1
|
|   |--Replace [old value] with [new value]
|       [ ] Match case
|       [ ] Whole words only
|       [ ] Case sensitive replace
|       [ ] Prompt on replace
|       [ ] Skip prompt if identical
|       [ ] First only
|       [ ] Extract matches
```

2 By the way, if you've never seen filters displayed as above, check out the File Menu\Export Filter to Clipboard option)

Creating Subfilters

A subfilter is where one filter appears inside another, just like subfolders do in Windows Explorer. In TextPipe, a subfilter indicates a constraint or restriction of some form. You can drop filters inside a comment filter so that both the documentation and the filters are kept together.

To create a subfilter:

1. First add both filters from the Filter Menu
2. Drag and drop the lower filter to the RIGHT HAND HALF of the upper filter in the filter view.
3. When you drop it, it will be added as a subfilter, inside the upper filter.

You can check the status line (at the bottom of the screen) to see where TextPipe is going to add the dropped filter. In general, dropping a filter in the right hand half of the filter view drops it as a subfilter (if the filter you are dropping it on supports subfilters), and dropping it in the left hand half of the filter view drops it underneath *but at the same level* as the filter being dropped on.

You can read more about subfilters in *Subfilters - What are they and how do I use them?*

www.datamystic.com/docs

Combining Restrictions

You can combine restrictions by putting one restriction inside another. In this example, we only want to reformat those lines of a report that contain the word 'Reference' and *also* contain the word 'Page':

```
|--Restrict to lines matching [REFERENCE]
|  [ ] Include line numbers
|  [ ] Include filename
|  [ ] Match case
|  [ ] Count matches
|  Pattern type: 0
|  Context before: 0
|  Context after: 0
|
|+--Restrict to lines matching [PAGE]
|  [ ] Include line numbers
|  [ ] Include filename
|  [ ] Match case
|  [ ] Count matches
|  Pattern type: 0
|  Context before: 0
|  Context after: 0
|
|  --Remove blanks from Start of Line
|
|  --Remove multiple whitespace
|
|+--Replace [REFERENCE] with [Reference]
|  [ ] Match case
|  [ ] Whole words only
|  [ ] Case sensitive replace
|  [ ] Prompt on replace
|  [ ] Skip prompt if identical
|  [ ] First only
|  [ ] Extract matches
```

Restrict to line and column ranges

Another example - you can restrict to Columns 2-20 of lines 1-100 like this:

```
|--Restrict lines:Line 1 .. line 100
|
|  |--Restrict columns:Column 2 .. column 20
|  |
|  |  |--Do something here...
```

2 Having a Restrict Lines inside a Restrict Columns **doesn't work** because a column range only consists of one line at a time.

Restrict to CSV and Tab-delimited field ranges

You can also restrict to CSV or Tab-delimited field ranges, to modify field values inside one or more fields -

```
|--Restrict CSV fields:CSV field 1 .. field 1
|
|  |--Convert to lowercase
```

HTML and XML restrictions

Let's say you wanted to rename an HTML FORM INPUT field from 'name' to 'surname', without affecting anything else:

```
|--Restrict to between tags <form>...</form>
|  [ ] Include text
|  [ ] Match case
|
|  |--Restrict to tag <input...>
|  |  [ ] Include text
|  |  [ ] Match case
|  |
|  |  |--Replace [name] with [surname]
|  |  |  [ ] Match case
|  |  |  [ ] Whole words only
|  |  |  [ ] Case sensitive replace
|  |  |  [ ] Prompt on replace
|  |  |  [ ] Skip prompt if identical
|  |  |  [ ] First only
|  |  |  [ ] Extract matches
```

Although we've omitted it for clarity, you could also add a 'Restrict to attribute' filter to ensure the replacement only occurred inside the NAME attribute of the INPUT tag.

Restricting to sections of a .INI file

You can restrict a replacement or set of changes to a single section of a .INI file. The trick is to use a search/replace as the restriction. The example below converts a .INI file section starting with [TmainForm] to lowercase. This can be easily changed to perform a search/replace, or to remove matching lines, or to add new lines with an Add Footer filter.

```

EasyPattern [[ mustBeginWith( '[TmainForm]', cr, lf, ), 1+ char,
mustEndWith( cr, lf, '[' ) ]] with [$0]
| [ ] Match case
| [ ] Whole words only
| [ ] Case sensitive replace
| [X] Prompt on replace
| [ ] Skip prompt if identical
| [ ] First only
| [ ] Extract matches
| Maximum text buffer size 4096
|
+--Convert to lowercase

```

If the .INI file section will be longer than 4Kbytes, increase the maximum text buffer size to suit.

Restrict to lines matching/not matching a pattern

There are two restriction filters that can be used to limit to lines matching or not matching a perl or EasyPattern pattern. In the example below, we remove columns 1 to 15 of lines starting with the word 'HD2'

```

|--Restrict to lines matching [^HD2]
| [ ] Include line numbers
| [ ] Include filename
| [ ] Match case
| [ ] Count matches
| Pattern type: 0
| Context before: 0
| Context after: 0
|
+--Remove column:Column 1 .. column 15

```

Restrict to the filename

You can perform search and replace, numbering and other functions inside the filename -

```

|--Restrict to file name
|
+--Convert to Title Case

```

Restrict to each line in turn

This filter is handy if you want to perform a search and replace against each line without the possibility of a new line being included in the found text.

Restrict to lines with X Tab fields/CSV fields

Using EasyPatterns, you can easily setup a restriction that will only match lines with a given number of CSV or Tab fields.

Using a Filters Menu\RestrictMatching Lines filter, use one of the following patterns:

Description	EasyPattern (*)
Match lines with exactly 4 Tab fields	[lineStart, 3 (TabField, Tab), TabField, lineEnd]
Match lines with 6 or more Tab fields	[lineStart, 5+ (TabField, Tab), TabField, lineEnd]
Match lines with 3 or less Tab fields	[lineStart, 0 to 2 (TabField, Tab), TabField, lineEnd]

Match lines with exactly 4 CSV fields	[lineStart, 3 (CSVField, Comma), CSVField, lineEnd]
Match lines with 6 or more CSV fields	[lineStart, 5+ (CSVField, Comma), CSVField, lineEnd]
Match lines with 3 or less CSV fields	[lineStart, 0 to 2 (CSVField, Comma), CSVField, lineEnd]

* Note how the number is generally set to one less than the number of fields required, because with n fields there are (n-1) fields each followed by a Tab/Comma, then one more field without a Tab/Comma.

Restrict to blocks of length XXX

This filter is handy for processing mainframe fixed block length reports, which don't have end of line characters.

The example below is for processing a mainframe report with blocks of length 764. Starting at column 164, there are 8 consecutive fields containing packed decimal values that must be expanded. Each field is 8 characters long.

```

|--Restrict to blocks of length 764
|
|  |--Restrict columns:64 columns starting at column 164
|  |
|  |  |--Restrict to blocks of length 8
|  |  |
|  |  |  |--VBScript script Timeout: 10000 milliseconds
|  |  |  |'Expand Mainframe EBCDIC Packed Decimal
|  |  |  |'ie PIC S9(15) COMP-3 (an 8-byte field)

```

You can read more about handling mainframe reports in *Working with Mainframe Reports*

www.datamystic.com/docs

Splitting log files

Let's say you have a log file where you need to split entries into new files based on various columns. In the example below, column 2 contains the event (EVENT FIRED or URL LAUNCH), and column 3 contains the job number. We want all data from job 11 to be split into two files, event_fired_11.log and url_launch_11.log.

```

18:33:39,EVENT FIRED,11,c:\program files\textpipe\textpipe.exe 1
18:33:40,URL LAUNCH,9,http://www.datamystic.com
18:33:39,EVENT FIRED,11,c:\program files\textpipe\textpipe.exe 2
18:33:40,URL LAUNCH,11,http://www.datamystic.com
18:33:39,EVENT FIRED,7,c:\program files\textpipe\textpipe.exe
18:33:40,URL LAUNCH,11,http://www.datamystic.com/1
18:33:39,EVENT FIRED,6,c:\program files\textpipe\textpipe.exe
18:33:40,URL LAUNCH,5,http://www.datamystic.com
18:33:39,EVENT FIRED,3,c:\program files\textpipe\textpipe.exe
18:33:40,URL LAUNCH,11,http://www.datamystic.com/2
18:33:39,EVENT FIRED,11,c:\program files\textpipe\textpipe.exe 3
18:33:40,URL LAUNCH,11,http://www.datamystic.com/3

```

Let's see how we would do this:

```

|--Restrict to lines matching [^[^,\r\n]*,[^,\r\n]*,11,]
|   | [ ] Include line numbers
|   | [ ] Include filename
|   | [ ] Match case
|   | [ ] Count matches
|   | Pattern type: 0
|   | Context before: 0
|   | Context after: 0
|
|   |--Restrict to lines matching [EVENT FIRED]
|   |   | [ ] Include line numbers
|   |   | [ ] Include filename
|   |   | [X] Match case
|   |   | [ ] Count matches
|   |   | Pattern type: 0
|   |   | Context before: 0
|   |   | Context after: 0
|   |
|   |   |--Merge output to file c:\event_fired_11.log
|   |
|   |   +--Remove all
|
|   +--Restrict to lines matching [URL LAUNCH]
|   |   | [ ] Include line numbers
|   |   | [ ] Include filename
|   |   | [X] Match case
|   |   | [ ] Count matches
|   |   | Pattern type: 0
|   |   | Context before: 0
|   |   | Context after: 0
|   |
|   |   |--Merge output to file c:\url_launch_11.log
|   |
|   |   +--Remove all
|
|--Comment...
|   All remaining data drops through here, so ensure it doesn't
|   overwrite your original file

```

Here we use Secondary Output Filters (found in the Filters\Special Menu), to capture the restricted set of lines. We follow each of these with a Remove All filter to ensure these lines don't re-enter the pipe.

There are many more possibilities - this is just to get you started.

Feedback and Questions

If you have feedback or questions about this documentation, please email us at:

support@datamystic.com

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

Email: sales@datamystic.com

Web site: www.datamystic.com

Phone: +61-3 9913-0595

Fax: +61-3 8610-1234