

## Table of Contents

NoComment—An input file management tool	2
Comments and Delimiters .....	3
REDIRECT—I/O Redirection .....	4
Automated Line Counting .....	6

## List of Figures

Figure 1.—Example removal of comments and counting of input line by NoComment... 2

Figure 2.—Example of subdividing a single input file into multiple files with I/O  
redirection. .... 5

Figure 3.—Example of ending statement “RETURN CONTROL” at the end of an  
auxiliary commented file that returns I/O to the primary commented file..... 5

Figure 4.—Example of defining variable, NPAR, to be replaced by count of lines  
between the keywords COUNTER\_START: and COUNTER\_STOP: for the  
variable NPAR..... 6

Figure 5.—Example of the markers COUNTER\_START: and COUNTER\_STOP: for  
counting the number of lines for the variable NPAR..... 7

Figure 6.—Example of integers that replaced variables in the uncommented input file.  
The highlighted value of 645 is the number of lines with parameter  
definitions, npar..... 7

## NoComment—An input file management tool

NoComment is a FORTRAN utility for managing input files for programs with limited commenting and file management abilities. Input files can be annotated with both comment lines and remarks to the right of data fields. Large input files can be subdivided into multiple files that are concatenated by NoComment. NoComment also can count the number of lines to be read in a section of input and specify that value in the uncommented input file.

NoComment creates a comment free input file with variables in the commented file replaced with line counts in the uncommented file (Figure 1). NoComment was developed to manage PEST control files and augment the exquisite documentation of PEST. All examples are PEST control files because PEST stimulated the need. NoComment can be applied to other programs such as FEHM, but development of commented input files is a user responsibility.

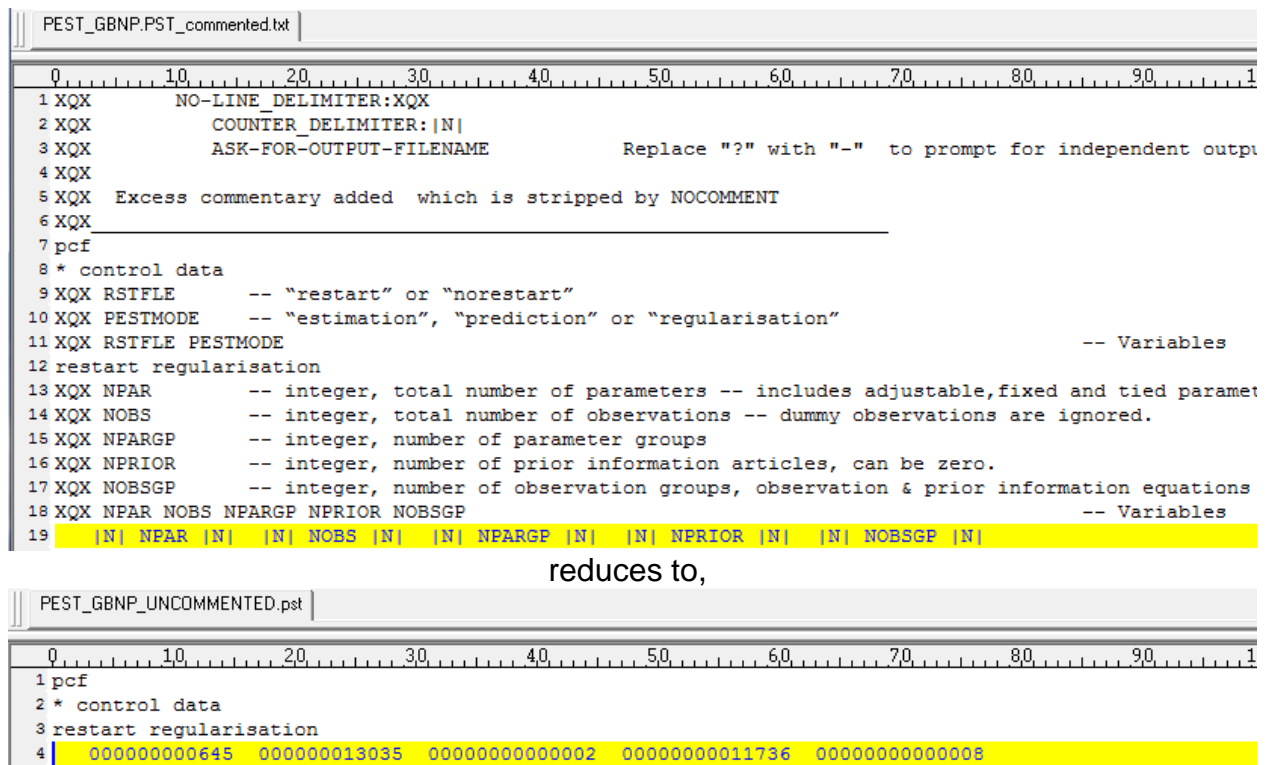


Figure 1.—Example removal of comments and counting of input line by NoComment.

## Comments and Delimiters

Comments are denoted by a text string of 1-10 characters that start a comment line or are to the right of all input data. Comment line, right-hand comment, and line-count delimiters are “XQX,” “!,” and unspecified, respectively, by default. Line counting is not enabled unless the user specifies a line-count delimiter. These delimiters can be redefined on the first 4 commented lines and are identified through the keywords; **NO-LINE\_DELIMITER:**, **NORIGHT\_DELIMITER:**, and **COUNTER\_DELIMITER:**.

Keywords for defining delimiters and their effects on file manipulation by NoComment are defined and explained in the following table. Keyword usage is case insensitive in NoComment.

<b>NO-LINE_DELIMITER:</b>	<p>Lines that begin with this text string are comments that are removed from the input file.</p> <p>This delimiter is defined within the first 4 commented lines of a commented input file.</p> <p>Default value is XQX.</p> <p>CHARACTER*10 sNOL</p>
<b>NO-RIGHT_DELIMITER:</b>	<p>All comments to the right of this delimiter and the delimiter are removed from the input file.</p> <p>This delimiter is defined within the first 4 commented lines of a commented input file.</p> <p>Default value is !.</p> <p>CHARACTER*10 sNOR</p>
<b>COUNTER_DELIMITER:</b>	<p>Variable names are bracketed with the line-count delimiters which are separated from the variable name by spaces. For example, the variable <b><i>npar</i></b> would be in the uncommented text as  N  <b><i>npar</i></b>  N  with  N  as the line-count delimiter.</p> <p>This delimiter is defined within the first 4 commented lines of a commented input file.</p> <p>Default value is undefined and line counting is disabled.</p> <p>CHARACTER*10 sCNT</p>

<b>ASK-FOR-OUTPUT-FILENAME</b>	<p>NoComment will query the user for an output file name if the text string, <b>ASK-FOR-OUTPUT-FILENAME</b> occurs within the first 4 commented lines of a commented input file (See sample2).</p> <p>The uncommented file will be the original commented input file name preceded by the prefix, <b>NoComment_</b>, if the keyword <b>ASK-FOR-OUTPUT-FILENAME</b> is not used (See sample1).</p>
--------------------------------	---

## ***REDIRECT—I/O Redirection***

A commented input file can be subdivided into smaller commented input files with I/O redirection. I/O can be redirected from the primary commented file to an auxiliary commented file by inserting an I/O redirect statement at any location in the primary commented file. Multiple I/O redirect statements can be used in the primary commented file. I/O redirection can only occur from the primary commented file and not from an auxiliary commented file.

The keyword **REDIRECT** in the primary commented file indicates that I/O will be redirected to an auxiliary commented file. The keyword occurs on an uncommented line where, the entire line is interpreted as instructions for NoComment. I/O redirection is of the form,

**REDIRECT** to file:AuxiliaryFileName.txt

where,

- The keyword **REDIRECT** must be present.
- Additional explanatory text can follow, such as “ to file.”
- A colon must be present after the keyword **REDIRECT** because it marks the start of the auxiliary commented file name.
- The auxiliary commented file name must follow immediately the colon and have a trailing space. Names can include path information so the auxiliary commented file can reside in parallel directories or subdirectories.

For example, the keyword **REDIRECT** appears on line 88 followed by a colon and the name of the auxiliary commented file name (Figure 2). The auxiliary commented file name is **PEST\_Parameters\_RECH.txt** which contains a long list of recharge rate estimates for pilot point values.

```

0      10      20      30      40      50      60      70      80
83 XQX SCALE      -- real, transforms PEST parameter, bp, to model parameter, bm, wit
84 XQX OFFSET      -- real, transforms PEST parameter, bp, to model parameter, bm, wit
85 XQX DERCOM      -- real, USE 1.0, Unless using PEST's external derivatives functio
86 XQX PARNAME PARTRANS PARCHGLIM      PARVAL1      PARLBNB      PARUBND PARGP SCALE OFFSET D
87 COUNTER_START:NPBAR
88 Redirect to file:PEST_Parameters_RECH.txt      !!      I/O redirect occurs here
89 Redirect to file:PEST_Parameters_HYD-K.txt      !!      I/O redirect occurs here
90 COUNTER_STOP:NPBAR
91 XQX (one such line for each of the NPAR parameters)
92 XQX
93 XQX TIED PARAMETER SECTION
94 XQX

```

**Figure 2.—Example of subdividing a single input file into multiple files with I/O redirection.**

I/O returns to the primary commented file when the end of the auxiliary commented file is reached or if the keywords **RETURN** and **CONTROL** are read from an uncommented line (Figure 3). The keywords occur on an uncommented line where, the entire line is interpreted as an instruction for NoComment.

```

0      10      20      30      40      50      60      70      80      90
206 RCH0306 fixed factor 1.0000000000000000E-08      0.00000001      0.00000001      Rch
207 RCH0307 fixed factor 1.0000000000000000E-08      0.00000001      0.00000001      Rch
208 RCH0308 fixed factor 1.0000000000000000E-08      0.00000001      0.00000001      Rch
209 RCH0309 fixed factor 1.0000000000000000E-08      0.00000001      0.00000001      Rch
210 RETURN CONTROL TO MAIN FILE

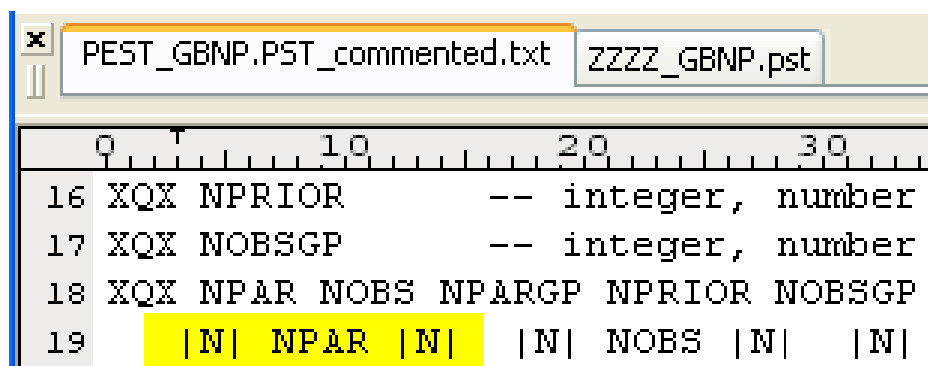
```

**Figure 3.—Example of ending statement “RETURN CONTROL” at the end of an auxiliary commented file that returns I/O to the primary commented file.**

PEST control files are better managed with the REDIRECT command because long repetitive sections of input can be isolated in auxiliary commented files. Parameter, observation, and prior information lists are best tracked in auxiliary commented files because the number of entries in each category can be easily checked.

## Automated Line Counting

Variables such as the number of parameters, observations, and prior information equations can be specified automatically by NoComment. A delimited variable name replaces the expected integer value in a commented input file (Figure 4). For example, the text string `|N| NPAR |N|` replaces the integer value 645 in a commented input file.



```
16 XQX NPRIOR      -- integer, number
17 XQX NOBSGP      -- integer, number
18 XQX NPAR NOBS NPARGP NPRIOR NOBSGP
19 |N| NPAR |N| |N| NOBS |N| |N|
```

Figure 4.—Example of defining variable, NPAR, to be replaced by count of lines between the keywords **COUNTER\_START:** and **COUNTER\_STOP:** for the variable NPAR.

The starting and stopping markers for the line-counting program are **COUNTER\_START:** and **COUNTER\_STOP:**, respectively. These keywords are immediately followed by the variable name and occur on uncommented lines (Figure 5). An entire line is interpreted as instructions for NoComment and is not written to the uncommented input file.

For example, the number of parameters to be read in a PEST control file is counted with NoComment (Figure 5). The string "COUNTER\_START:" is read from line 87, which initializes a counter and NPAR after the colon specifies that the following lines be counted for the variable **NPAR**. All lines read after the COUNTER\_START: keyword are counted regardless of occurring in the primary or auxiliary commented files. Instruction lines for NoComment, such as REDIRECT or RETURN CONTROL, are not counted. The string "COUNTER\_STOP:" is read from line 90, which stops the counter for the variable **NPAR**.

The screenshot shows a text editor window with two tabs: 'PEST\_GBNP.PST\_commented.txt' and 'ZZZZ\_GBNP.pst'. The main window displays the contents of 'PEST\_GBNP.PST\_commented.txt'. The text is as follows:

```

85 XQX DERCOM      -- real, USE 1.0,  Unless usin
86 XQX PARNME PARTRANS PARCHGLIM      PARVAL1  P
87  COUNTER_START:NP&R
88  Redirect to file:PEST_Parameters_RECH.txt
89  Redirect to file:PEST_Parameters_HYD-K.txt
90  COUNTER_STOP:NP&R
91 XQX (one such line for each of the NP&R parame
--

```

The markers 'COUNTER\_START:NP&R' and 'COUNTER\_STOP:NP&R' are highlighted in yellow. The line 'XQX (one such line for each of the NP&R parame' is highlighted in light blue.

**Figure 5.—Example of the markers COUNTER\_START: and COUNTER\_STOP: for counting the number of lines for the variable NP&R.**

Variable names and bounding line-count delimiters are replaced with integer values of counted lines while the uncommented input file is written (Figure 6). Primary and auxiliary commented input files are read and written to a temporary file, temp.txt, as lines are counted for variables. The temporary file is rewound. Lines without line-count delimiters are echoed directly into the uncommented input file.

The screenshot shows a text editor window with two tabs: 'PEST\_GBNP.PST\_commented.txt' and 'ZZZZ\_GBNP.pst'. The main window displays the contents of 'ZZZZ\_GBNP.pst'. The text is as follows:

```

1 pcf
2 * control data
3 restart regularisation
4 0000000000645 000000013035 0000000
5 0000000000000005 0000000000000002

```

The value '0000000000645' on line 4 is highlighted in yellow, representing the number of lines with parameter definitions. The entire line 4 is highlighted in light blue.

**Figure 6.—Example of integers that replaced variables in the uncommented input file. The highlighted value of 645 is the number of lines with parameter definitions, np&R.**

The temporary file, temp.txt, is deleted with the INTEL compiler specific call DELFILESQQ. Find another call for killing the temporary file in your compiler if you must recompile NoComment. Alternatively, comment out the DELFILESQQ call and kill the temporary file with batch or script commands.