UNIVERSITY OF EDINBURGH

Edinburgh Regional Computing Centre

User Note 55

(November 1986)

Title:

File Comparison Programs on EMAS 2900 and EMAS-3

Author:

Arthur Wilson

Contact:

Advisory service

Software Support Category:
See Note 15

Synopsis

On the EMAS 2900 and EMAS-3 operating systems there are a number of programs which can be used to compare the contents of two files. These are called file comparison programs.

For each one, this Note will:

- describe its facilities,
- show you how to use it,
- give examples of the input it can take and the corresponding output,
- point you to relevant documentation.

Keywords

CHECKENDS, COMPARE, CONTRAST, DIFF, PDCHECK, PDVERIFY, TEXTCOMPARE, YCOMP

Edinburgh Regional Computing Centre

James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ. Telephone 031-667 1081

1 INTRODUCTION

What is a file?

A computer file is the unit which is used to store a collection of letters, numbers, symbols, etc. These may combine to make a report, letter, computer program, data for input etc.

What types of file are there?

A file may be one of the following types:

- TEXT, a combination of letters, numbers, special symbols and control characters, making a report, letter, computer source program etc.
- DATA, computer code meaningful only as input to a computer program,
- OBJECT, computer code generated by a compiler,
- PARTITIONED, holding a number of subsidiary files,
- GROUP, a set of on-line files only available on EMAS-3. Note that GROUPS as entities cannot be used by any of the programs described in this Note. However, a member or members of a GROUP can be compared.

When would you compare two files?

You may need to compare two files to see if they are identical or not. If you have established that they differ, you may wish to find out where the differences are.

Note that if either of the two files belongs to another user it will have to be permitted to you (see details of the PERMIT command on page 5-4 of the EMAS 2900: User's Guide, and in Section 3.7.1 of the EMAS-3 User's Guide).

How to access the programs described in this Note

On the EMAS 2900 system, the programs described in this Note are held in the CONLIB.GENERAL and KNTLIB.GENERAL directories. To access them you would use the OPTION command like this:

```
Command: OPTION SEARCHDIR=CONLIB.GENERAL
Command: OPTION SEARCHDIR=KNTLIB.GENERAL
```

On EMAS-3 they are held in the ERCLIB:GENERAL directory which would be accessed by using the SEARCHDIR command in the following way:

```
Command: SEARCHDIR ERCLIB:GENERAL
```

Note that on EMAS-3, the directory name is separated from the user name by a colon, not a full stop as on EMAS 2900.

Information

Further information on the programs described in this Note can be obtained from the ERCC Advisory service at King's Buildings 031-667 1081 ext. 2976/7, or George Square 031-667 1011 ext. 2300. To get on-line information on all these programs, use the HELP command. For example:

Command: HELP CONTRAST

will list a short description of the CONTRAST program on your terminal.

For various reasons the programs described in this Note can be given only a low level of user support. User Note 15, which is available from the ERCC Advisory service, gives full details of Software Support Categories and the particular level of support given to the items described in this Note.

Note that the examples shown in this Note were generated on the EMAS-3 system; they might have been slightly different had they been generated on EMAS 2900.

A TABLE OF AVAILABLE FILE COMPARISON PROGRAMS

Program Name	File type	Short Description	Directory
CONTRAST (page 4)	Any	Shows how many bytes at start and end of two files are identical and prints the first different byte with its context otherwise a fast check to see if two files are identical.	CONLIB.GENERAL on EMAS 2900, ERCLIB:GENERAL on EMAS-3.
CHECKENDS (page 5)	Any	Fast check that files are identical.	CONLIB.GENERAL on EMAS 2900, ERCLIB:GENERAL on EMAS-3.
YCOMP (page 6)	Any	Makes a binary comparison of two files and asks for a relative start position.	CONLIB. GENERAL on EMAS 2900, ERCLIB:GENERAL on EMAS-3.
COMPARE (page 6)	Text	Useful for large files where you have some idea of the differences. Cannot cope with text moved from one place to another.	CONLIB. GENERAL on EMAS 2900, ERCLIB: GENERAL on EMAS-3.
DIFF (page 9)	Text	Faster than TEXTCOMPARE on large files. Has several useful optional facilities.	KNTLIB.GENERAL on EMAS 2900. Not available on EMAS-3
TEXTCOMPARE (page 10)	Text	Describes files or members only. Use TEXTCOMPARE if you do not know how to use anything else.	CONLIB.GENERAL on EMAS 2900, ERCLIB:GENERAL on EMAS-3.
PDCHECK (page 12)	PD	Compares members of a partitioned file against non-partitioned files of the same name.	CONLIB.GENERAL on EMAS 2900, ERCLIB:GENERAL on EMAS-3.
PDVERIFY (page 13)	PD	Compares contents of two partitioned files.	CONLIB.GENERAL on EMAS 2900, ERCLIB:GENERAL on EMAS-3.

Example files

The two text files DICT1 and DICT2 listed below, are used as examples throughout this note.

```
DICT1                                  | DICT2

differ        v.i.  to be unlike       | differ       v.i.  to be unlike
                    to dispute         |                    to fall out
                    to fall out        |                    to dispute
                    to disagree        |                    to disagree
difference    n.    unlikeness         |
                    disagreement       |
```

2 CONTRAST

This program can be used to compare two files of any type on a byte by byte basis. If they are identical, a message saying so is displayed, if not, then a report is printed which tells you:

- the file type (only if types differ);
- how long each file is;
- at what byte number did a mismatch occur between them;
- which bytes mismatch;
- the context, i.e. the surrounding bytes in each file.

How to run the CONTRAST program

The CONTRAST command takes two parameters:

FILE1 - any file,
FILE2 - any file except for FILE1.

For example:

```
Command: CONTRAST DICT1,DICT2
```

would produce the following report on your interactive terminal:

```
CONTRAST file-comparison:   version 2.1 (17th June 1986)
Lengths of files: ERCN07:DICT1= 168  ERCN07:DICT2= 112
Mismatch in files at byte no. 48
Byte in 'ERCN07:DICT1':  100 = HEX_64 = 'd'
Byte in 'ERCN07:DICT2':  102 = HEX_66 = 'f'
Context in ERCN07:DICT1:      2020746F20646973707574    to disput
Context in ERCN07:DICT2:      2020746F2066616C6C206F    to fall o
Mismatch marker:                        ^^

The last 1 byte of the two files is identical.
```

CONTRAST numbers bytes from 0 e.g., byte 1 is the second byte in the file.

When would you use the CONTRAST program?

Use for a quick check on the similarity of two character files (for a more detailed comparison see TEXTCOMPARE which is described in Section 7) or to discover if any non-text type files are identical.

3 CHECKENDS

The CHECKENDS program compares any two files. It produces a report which tells you:

- whether the two files are of the same or different types;
- how many bytes each one contains;
- the number of bytes that match at the start;
- the number of lines in each (if character files);
- the line on which the mismatch occurs;
- the number of matching bytes at the end of each file;
- the line from the end on which the first mismatch occurs (if they are both character files).

How to run the CHECKENDS program

The CHECKENDS command takes two parameters:

FILE1 - any file,
FILE2 - any file except for FILE1.

For example, this command:

```
Command: CHECKENDS DICT1,DICT2
```

would produce the following report on your interactive terminal:

```
ERCN07:DICT1 and ERCN07:DICT2 are CHARACTER files.
File ERCN07:DICT1 contains 168 bytes of data ( 6 lines).
File ERCN07:DICT2 contains 112 bytes of data ( 4 lines).
First 48 bytes match
        (mismatch occurs after character 19 on line 2).
Last byte matches
        (mismatch occurs on the last line).
```

As well as the byte number, this program gives you the line and character number at which the first mismatch occurs. You can therefore use a context editor to go to this line and character number and see where the files begin to differ.

Where would you use the CHECKENDS program?

CHECKENDS is faster than CONTRAST at determining whether two files are identical or not. It summarizes the similarities whereas CONTRAST shows the differences. As the name suggests, it checks from the end of both files and reports on the matching bytes.

4 YCOMP

With YCOMP you can compare any two files of the same type. It asks you for their names and a start address (relative to the beginning of the file). The comparison is then done from the start address up to the end of the smaller file. If a difference is found, the program stops and prints the relative address of the mismatch and the words containing the differing bytes.

How to run the YCOMP program

DICT1 and DICT2 are two slightly different character files. If you compare them using YCOMP, the output on your interactive terminal would look like this:

```
Command: YCOMP
File1: dict1
File2: dict2
Rel start: 32
DIFF AT REL ADDRESS: 00000050 64697370 66616C6C
Continue?(no) : no
```

The *Rel start:* prompt requires you to type in the address (decimal or hexadecimal) where you want the comparison to start from. Note that any hexadecimal address must be preceded by the letter 'X'. YCOMP compares two files from the very beginning of the file, including the header which takes up 32 bytes. To start the comparison after the file header you should give the number 32 or X20 as the *Rel start:* address.

Where would you use the YCOMP command?

Its main advantage is that it allows you to start a comparison other than at the start of the files. Its report is precise, unlike CONTRAST and CHECKENDS which give details of file sizes and the context of mismatching bytes.

5 COMPARE

This program is used to compare two text files: members of partitioned files or groups are allowed. COMPARE takes you through the comparison step by step and allows you to give program directives when a difference is found.

How to run the COMPARE program

The following command starts the program:

```
Command: COMPARE FILE1,FILE2
```

When a line from each file is found not to match, the program pauses and prompts you for input with a colon. After the colon, you can tell the program to:

- print the current line in one or both of the files,
- print the current and previous four lines in both,
- move forward a number of lines in one,
- search for the occurrence of a text pattern in one or both.

The directives that you can issue after the colon prompt are as follows:

PB print the current line in both files,

C print the current and the previous four lines in both files,

Pf print the current line in file **f**, where **f** is file number 1 or 2,

A advance one line in each file and restart the comparison. When the end of one file is reached, the message *COMPARISON COMPLETE* will be printed.

Mfn move forward **n** lines in file **f**, where **f** is file number 1 or 2. If you reach the end of file **f**, the message *EOFf* will be printed, where *f* is file number 1 or 2, and the program will stop,

MBn move forward **n** lines in both files,

GO restart the comparison of successive lines. As with the Advance command, the message *COMPARISON COMPLETE* will be printed when the end of one file is reached.

Fftext move forward until the required text is found in the file **f**, where **f** is file number 1 or 2. If you reach the end of file **f**, the message *EOFf* will be printed, where *f* is file number 1 or 2, and the program will be terminated.

FBtext find the required text in both files. If the required text is not found a message ***EOFf*** will be printed (where *f* is the file number), and the program will be terminated.

Q quit,

E end,

S stop,

: stop,

%c stop.

If you type an invalid command after the colon prompt, the reply:

```
        NO
```

will be printed.

If you use the COMPARE program to compare the two files used in previous examples, the following output would be generated on your terminal:

```
DIFF
                to dispute
                to fall out
Type ? for commands
: PB
                to dispute
                to fall out
: C
File 1 -------------------------
differ    v.i.  to be unlike
                to dispute
                to fall out
File 2 -------------------------
differ    v.i.  to be unlike
                to fall out
                to dispute
: Q
```

When would you use the COMPARE program?

COMPARE is the ideal program to use if you want to:

- see the context of any mismatching lines,
- quickly guide the program through the analysis (using the Move and Find commands),
- set starting points for the comparison and adjust them if necessary (again using the Find and Move commands).

If a comparison of large files is going to produce only a few mismatches then COMPARE is the better program to use. If DIFF (see Section 6) or TEXTCOMPARE (see Section 7) is too slow, and you have a good idea of the differences between the files, then try COMPARE. You can save a lot of time by guiding the program through the files.

However, COMPARE (unlike TEXTCOMPARE) does not give details of lines that exist in both files but at different locations.

COMPARE is useful if you know that two files differ and you want to know where. For example, the two files ELEM1 and ELEM2 each contain a short list of chemical elements. They should be identical, but there is a slight difference. The COMPARE program will show you any lines that differ and will allow you to display the context of any discrepancies. The command

```
        Command: COMPARE ELEM1,ELEM2
```

would generate the following report on your interactive terminal:

```
DIFF
Argon          A.
Arsenic        As.
Type ? for commands
:
```

This shows you the lines that do not match; if you want to see their context, type C after the colon. For example:

```
Type ? for commands
:C
File 1 -------------------------
Aluminium      Al.
Antimony       Sb.
Argon          A.
Arsenic        As.
File 2 -------------------------
Aluminium      Al.
Antimony       Sb.
Arsenic        As.
Barium         Ba.
:Q
```

From the report you can see that the line

```
Argon          A.
```

is missing from file ELEM2.

6 DIFF

This program is available on EMAS 2900 but not on EMAS-3. It can be used to show the difference between two character files. It refers to the first file as OLD and the second file as NEW. When the comparison is complete, a report is generated which shows:

- the lines in the OLD file which are not in the NEW,
- lines in the NEW file which are not in the OLD,
- lines in the OLD file that have changed in the NEW,
- groups of lines which have moved position.

The DIFF program produces the following codes on the report:

<o1> , <o2>	*d*				- a delete
	a	*<n1>*	*,*	*<n2>*	- an append
<o1> , <o2>	*c*	*<n1>*	*,*	*<n2>*	- a change
<o1> , <o2>	*m*	*<n1>*	*,*	*<n2>*	- a move (a moved block of unchanged lines)

where

<o1> , <o2>	= range of lines in the oldfile
<n1> , <n2>	= range of lines in the newfile

Unique lines from the OLD file are listed with an asterisk * before them, while unique lines from the NEW file are indicated by a full stop.

If the two files are identical the program says so.

How to run the DIFF program

You start the program by typing the DIFF command with up to eight parameters, the first two of which are compulsory. These parameters are listed below:

1 OLDFILE the OLD file for comparison,

2 NEWFILE the NEW file for comparison,

3 OUTPUT the destination where you want the report to be sent, this can be a device or a file name, the default is .OUT,

4 FIRSTLINE the line number where you want the comparison to start, default is 1,

5 LASTLINE the line number where you want the comparison to stop, the default is the length of the larger of the two files. The maximum length is 20000.

6 FIRSTCOL the column number where you want the comparison to start, the default is 1,

7 LASTCOL the column number where you want the comparison to end, the default is the number of columns in the longest line in either of the two files. The maximum width is 8192.

8 STRIP strip trailing spaces of lines? You can use YES or NO for this parameter. If you omit it, the default value of NO is selected.

When would you use the DIFF program?

DIFF would be the ideal choice of program if you wanted to compare only part of two files.

Here is a practical example of the results produced by the DIFF program. Take these two files:

```
file1 = LIST1     |   file2 = LIST2

one               |   six
two               |   two
three             |   three
four              |   nine
five              |   ten
six               |
```

and compare them using DIFF like this:

Command: **DIFF LIST1,LIST2**

```
The results would look like this:  |   Each line is explained below.

   DIFF - Version 3.1
   1d                              |   tells you that the first line is different.
   * one                           |   is a unique line in the old file.
   4,5c4,5                         |   lines 4 and 5 in the old have changed in the new.
   * four                          |   line 4 in the old file.
   * five                          |   line 5 in the old file.
   . nine                          |   line 4 in the new file.
   . ten                           |   line 5 in the new file.
   6m1                             |   line 6 in the old file has moved to line 1.
   * six                           |   the line that has moved.
```

More information about DIFF

DIFF was written at Kent University. It is described in a document called *DIFF - a program for comparing the contents of two files*, REF DOC/EMAS.K2.5/21 which is available from the ERCC Advisory service.

7 TEXTCOMPARE

This program compares two character files. It tells you:

a. the number of lines in each,
b. which lines in the first match those in the second,
c. those lines in the first which cannot be matched in the second,
d. the lines in the second which cannot be matched in the first.

How to run the TEXTCOMPARE program

The TEXTCOMPARE command has four parameters of which the first two are mandatory. The parameters are as follows:

FILE1 the name of the first character file.
FILE2 the name of the second character file, which must be distinct from FILE1.
DEVICE the destination for the report of the comparison. It can be a file name or a device name, for example, file name FRED or device name .LP23. If you omit the DEVICE parameter, the default value (.OUT) is selected and the report is listed on your interactive terminal.
CONTROL a string which can be a combination of the following:

 M - list lines which match in both,
U1 - list lines in FILE1 unmatched in FILE2,
U2 - list lines in FILE2 unmatched in FILE1.

If you omit the CONTROL parameter, you get a report which includes all the items described in a, b, c and d above.

There are a number of possible error messages:

- FILES not distinct,
- FILE1 and/or FILE2 not available,
- FILE1 and/or FILE2 not of type CHARACTER.

The command

```
Command: TEXTCOMPARE DICT1,DICT2
```

would produce the following output on your interactive terminal:

```
File ERCN07:DICT2 has 4 lines;
File ERCN07:DICT1 has 6 lines.

Matching lines, and unmatched lines in ERCN07:DICT1:

Match: Line 1 is the same in both files.

Match: ERCN07:DICT1 line 2 matches ERCN07:DICT2 line 3

Match: ERCN07:DICT1 line 3 matches ERCN07:DICT2 line 2

Match: ERCN07:DICT1 line 4 matches ERCN07:DICT2 line 4
5:   difference n.  unlikeness
6:                  disagreement

2 lines in ERCN07:DICT1 are unmatched in ERCN07:DICT2

ERCN07:DICT1 includes every line of ERCN07:DICT2
```

When would you use the TEXTCOMPARE program?

Always start with TEXTCOMPARE if you want a detailed report of two character files. If nothing happens for several minutes, interrupt the program and try COMPARE or DIFF.

TEXTCOMPARE is a suitable choice of program if you want one that never misses matching lines no matter how different the two files may be. However, if they contain a lot of different information, the program will take a long time to produce a report which will contain too much detail.

You could also use the TEXTCOMPARE program to:

- show how far you had got with an edit (by comparing old and new versions),
- compare output produced by old and new versions of a program, to show that the new version is working correctly.

8 PDCHECK

This program takes all the members of a partitioned file and checks to see if an ordinary file of the same name exists. It produces a report which for each partitioned file member tells you:

- the type,
- when the member was last altered,
- if an ordinary file of the same name exists,
- when it was last altered (if different from the member file).

If an ordinary file and a member file were last altered at the same date and time, they are assumed to be identical, and a message saying *COMPARISON COMPLETE* will be printed on the report.

How to run the PDCHECK program

The command would be issued like this:

Command: **PDCHECK HLPTS**

Below is an example of the type of output produced by this command:

```
------------------------------------- Analysis of PDfile: ERCN07:HLPTS --------------------------

Member      Type        File of same name   Member last altered     File last altered

BTCH        Character                       03/10/83 09.16.23       17/05/84 12.28.34
FAILTOT     Character   does not exist      04/03/83 15.50.12
HAMDEV      Character   does not exist      13/09/83 10.15.21
HAMPASC     Character                       16/05/83 09.31.09       COMPARISON COMPLETE
HAMPASCY    Object      does not exist      16/05/83 09.31.36
IN          Character                       31/01/83 10.11.56       16/05/84 14.45.41

ANALYSIS COMPLETE
```

When would you use the PDCHECK program?

If you store your programs and data in partitioned files, you could use PDCHECK as a housekeeping tool to ensure that you have tidied up your files after the last round of modifications. You could also use PDCHECK in conjunction with the UPDATE program to keep your files tidy. This may mean replacing or deleting members of the partitiond file. For more information, use the HELP command

Command: HELP UPDATE

to list a short description of the UPDATE program on your interactive terminal.

9 PDVERIFY

This program checks that two partitioned files contain the same members and are properly structured. If so, a message saying

File2 contains the same members as File1

will be printed. If the files have a different number of members, then a message saying

Different numbers of members

will be printed.

If they have the same number and names of members, but the contents differ, the message

File2 does not match File1

will be printed.

How to run the PDVERIFY program

It takes two parameters:

File1 - the name of a partitioned file,
File2 - the name of another partitioned file.

If you had two identical partitioned files called MASTER1 and OLDMASTER, the command:

Command: PDVERIFY MASTER1,OLDMASTER

would produce the following output on your interactive terminal:

MASTER1 contains the same members as OLDMASTER

When would you use the PDVERIFY program?

You could use the PDVERIFY command to check quickly if two partitioned files contain the same members, if they have a different number of members or if they have the same number of members but with different contents.

10 Notes

If you are comparing character files, it may help if you make line numbered listings of both files. You can do this with the LANDNUM or LINENO commands, for example:

```
Command: LANDNUM DICT1,.LP
  or
Command: LINENO DICT2,.LP
```

These commands will generate copies of the two files on the local line printer with each line numbered.

LANDNUM is held in the ERCLIB:GENERAL directory on EMAS-3 (CONLIB.GENERAL on EMAS 2900), while LINENO is held in KNTLIB:GENERAL (KNTLIB.GENERAL on EMAS 2900).

If the two files you are comparing are program source files, you could generate line numbered listings by compiling each one and then sending the compiler listing to a line printer. The compiler listing is put in the file T#LIST by default, so to send a copy to the local line printer you would use the following command:

```
Command: LIST T#LIST,.LP
```

If you are looking at a file on your terminal, you can use one of the context editors, EDIT or ECCE to move to a particular line number and column. For more information on these two and other editors, ask the ERCC Advisory service for a copy of User Note 46 entitled *Editors available on the EMAS Operating System*.