

!CSVamp

Version 3.10 - 16th February 2008

Ray Favre © 2008

(Freeware, authored with Dr Wimp)

Contents

<i>Introduction</i>	2
<i>Initial installation and start-up</i>	3
<i>User operations</i>	
1) On loading a file	5
2) Constructing a new output file in the main window	6
3) Saving a constructed output file	8
4) Saving the construction settings	9
5) Loading a construction settings file (type &005)	9
6) Adding extra fields	10
7) Splitting a field into two	11
8) Sorting	14
<i>Comments and limitations</i>	16
<i>Examples of use</i>	18
<i>Contacts</i>	20
<i>History</i>	21

Introduction

!CSvamp is a utility application whose sole purpose is to make it easy to restructure ('revamp') CSV/TSV files i.e. 'comma-(or tab)-separated-variable' data files.

For example, if you start with a CSV/TSV file containing a number of records, then you can quickly:

- change the order of the fields within each record;
- extract fields, in any order;
- join fields together, with/without inserted text between them;
- add leading/trailing text to a field;
- add quotes around fields;
- add extra fields;
- split any field into two;
- change between 'comma-separated' and 'tab-separated' formats;
- change the record terminator between line-feed (LF), return (CR), LF+CR and CR+LF;
- choose which input file records will be included in the processing for the output file;
- produce output file as either 'CSV' (&dfe) or 'Text' (&fff) file-type;
- sort an input file alphabetically (up or down) - with the option to ignore the first record.

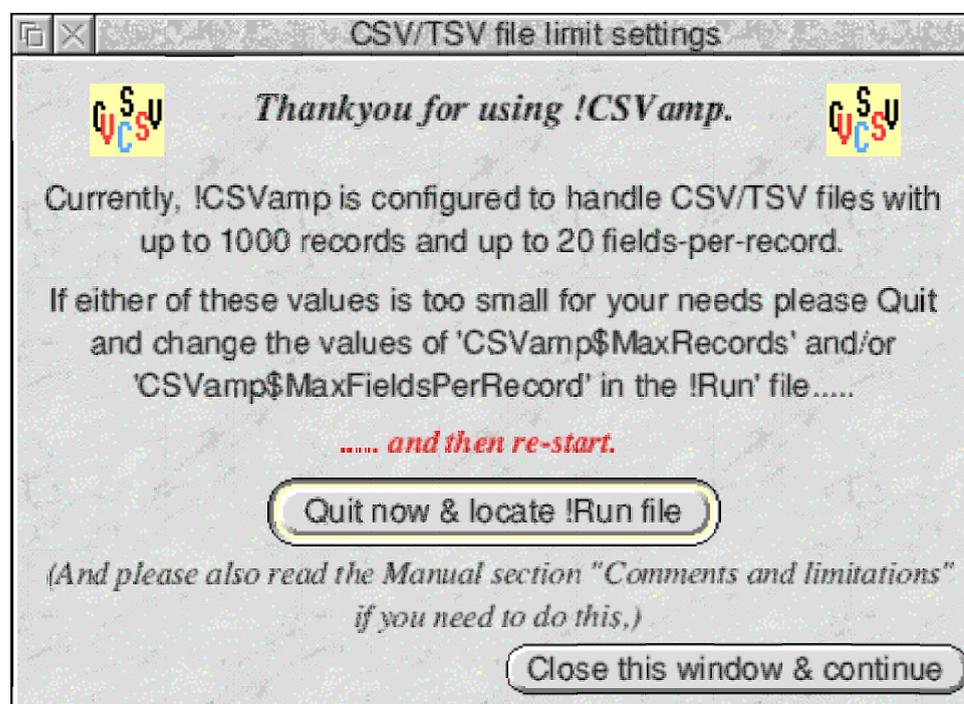
In all cases, the original file is left unaltered and the changes are made in a new output-file, saved in the usual way.

Initial installation and start-up

Copy the application to wherever you wish on a hard disc.

Double-click on the application icon to start it: this produces an iconbar icon in the usual way.

Clicking the iconbar icon will display the following initial banner window.



This banner window shows the maximum number of records and the maximum number of fields-per-record that your copy of !CSVamp is currently configured to handle - as a result of values set in the application's !Run file.

If any of the files you intend to load in this session exceed either of the shown limits you need to quit the application at this stage, change the limits and re-start. Pressing the 'Quit now & locate !Run file' button will duly quit and bring up the directory window containing the !Run file, ready for you to change either or both of the limits. Later in the Manual the Section 'Comments and limitations' gives details of what needs to be changed. *(You can also take the same action using 'Quit & locate !Run file' from the iconbar menu - at any time.)*

However, in the more likely case that the limits are satisfactory for your current needs, press the 'Close this window & continue' button to open the main window. *(There is also an alternative way of continuing described a little later below.)*

On initial opening, the main window will have nearly all its icons disabled and will also have just a single row of 'field icons' - indicated by "Field #1" at the left-hand side.

The window title tells you to drag a CSV/TSV file to the window or to the iconbar icon. For meaningful use the dragged file needs to be in CSV or TSV format but it can have a file-type of either 'CSV' (&dfe) or 'Text' (&fff).

Once the iconbar icon is installed, files of the file-type 'CSV' (&dfe) can equally well be loaded just by double-clicking on the file.

If the main window is not already open, loading a file will also open the main window.

Whilst the above-shown initial banner window is open there is an alternative file-loading procedure if you are sure that your CSV/TSV file does not exceed the shown limits. In this case, dragging the file to the banner window (or to the iconbar, or double-clicking on a file of file-type 'CSV' (&dfe)) will have the same effect as pressing the 'Close this window & continue' button.

Any subsequent file-loading action will simply replace the existing loaded file.

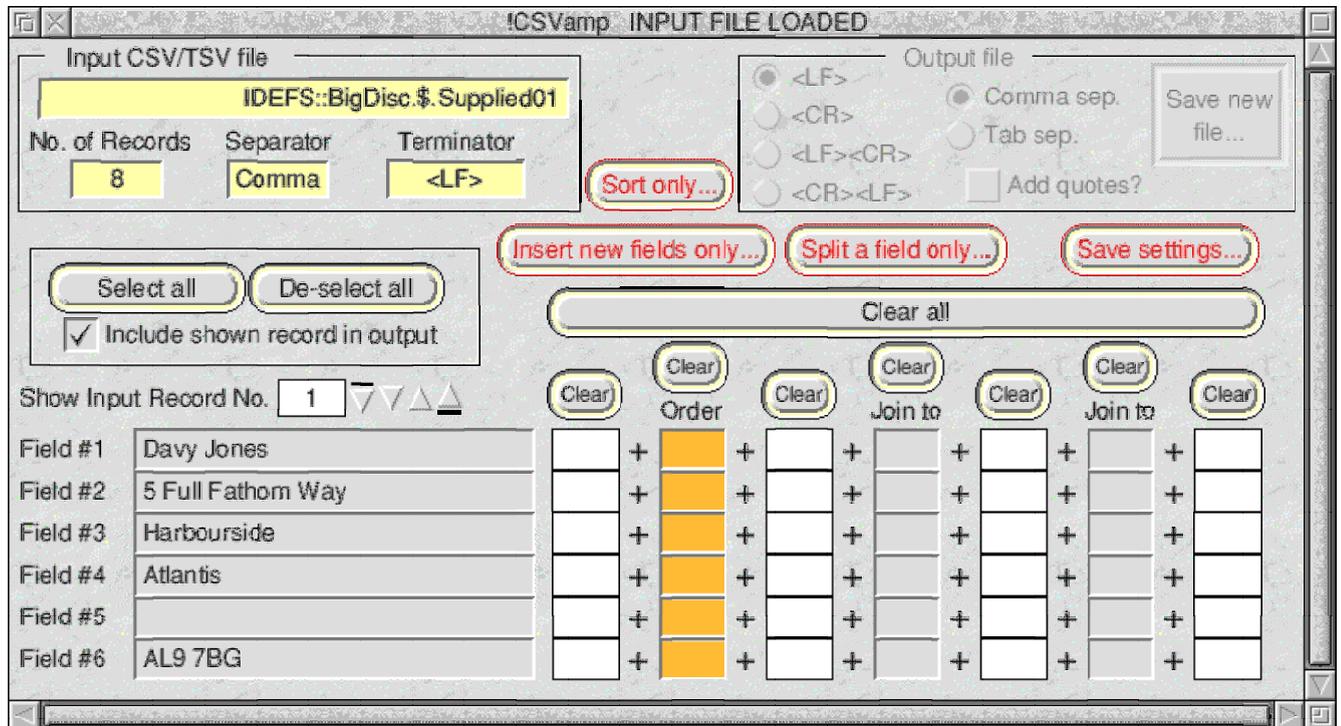
(The application itself cannot be started up by double-clicking on a CSV/TSV file. This is deliberate.)

Please note that the first record of a loaded CSV/TSV file will be taken as the defining one for the subsequent processing e.g. number of fields-per-record, separator character, record terminator character(s) - see the Section 'Comments and limitations' for more details.

User operations

1) On loading a file

When a CSV/TSV file is loaded the main window becomes active - and will look typically like the following screenshot.



The window title has changed to confirm that a file is loaded and the particular file path is shown in the top left icon.

Below this is shown the number of records in the loaded file; its field separator (comma or tab); and its record terminator (LF, CR, LF+CR or CR+LF) - **all defined from the first record of the file.**

The number of rows of 'field icons' in the window will have been increased to match the number of fields-per-record - and the contents of the first record will be displayed.

Thus, in the shown example above, there are 8 records; the field separator is a comma and the record terminator is LF - and each record contains 6 fields. The first record is displayed and it is seen that its 5th field happens to be blank.

In this display, any **surrounding** double-quotes to a field will have been stripped. So, anything visible in this display is part of a field's true content - which might include other double-quotes, commas, etc.

Purely as a convenience, which has no effect on the output produced, the contents of the other records can be examined by using the nudger icons or by entering a record number into the writable icon in the usual way.

Whenever a different file is loaded the number of rows of 'field icons' will be automatically changed appropriately - and the window size adjusted accordingly, if

necessary.

Finally, on loading a file, you will notice that the group of icons concerned with saving a new output-file (*at the top right of the window*) are still all disabled at this stage.

2) Constructing a new output file in the main window

A new output-file construction is based on the currently loaded file and its structure is determined by making entries (in the main window) into the rows of 'field icons' to the right of the icons showing the field contents.

In each of the field rows there are seven icons capable of affecting the output structure. These icons are colour coded and operate differently:

a) The column of orange icons (with "Order" shown at the top).

This is the only column in which **at least one entry must be made in order to cause the group of 'save icons' to be enabled** (*at the top right of the main window*).

This orange column works by clicking <select> or <adjust> over an icon to designate the required order in the output file of the input-file fields. The first click puts "1" in the clicked icon: the second click puts "2" and so on.

A click on an icon already filled will do nothing. So the only way to change any existing settings in this column is to hit the 'Clear' button at the top of the column and start again.

You are not obliged to choose more than one input-file field in this column. Each one selected represents one field in the output-file - so if you only choose 1 field in this orange column there will only be one field-per-record in the output file. Simply choose the fields you want - in the order you want them to appear - and leave the rest blank. (You can, of course, choose them all.)

An example will make the above operation clearer:

i) Assume that you have loaded a CSV file with six fields per record - as in the earlier screen-shot. The display will show initially each field of the first record - and there will be six blank orange icons in a vertical column.

ii) Let's assume you want to produce an output file by extracting only three of the fields from the input file, and you want the first field of the output file to be Field #6 from the input file; the second output field to be Field #3 from the input file; and the third output field to be Field #1 from the input file.

iii) Make your first click in the orange icon in the Field #6 row - a 1 appears in it. (*And notice that the 'save icons' in the top right panel become enabled.*)

iv) Make your second click in the orange icon in the Field #3 row - a 2 appears in it. And your third click in the orange icon in the Field #1 row - a

3 appears in it.

v) Leave the remaining orange icons untouched i.e. Fields 2, 4 & 5 of the input file will be ignored.

You can now proceed to make further constructions as described below.

b) The two columns of grey icons (with “Join to” shown at the top)

These two columns allow you to join (concatenate, string-wise) - in the output-file - up to two other input-file fields to the corresponding (orange column) input-file field.

Entries are again made by clicking over the icons but the action is different this time. The first click with <select> over any grey icon will enter “1” and the second <select> click **over the same icon** will produce “2” and so on. Clicking with <adjust> will reduce the number by 1. The allowed range is automatically constrained to the number of fields in the input-file (i.e. number of rows displayed). Thus you can choose any input-file field any number of times in these grey columns and you can alter the choice at will. (The ‘Clear’ buttons will again clear all entries in their respective columns.)

Thus, if you have chosen input-file Field #6 to be the first field of the output-file (i.e. you will have put “1” in the orange icon of the “Field #6” row) and you now enter 3 and 1 respectively in the two grey icons of this same row - then the output-file will have **all the first fields of its records** made up of a concatenation of input-file fields 6+3+1. (*If you wish to insert a blank space - or any other text - between these fields then you will need to use the intervening white writable icons, as below.*)

c) The white writable icons

These work in the usual way and simply give you a number of options for inserting your own choices of text/characters in various places in the output fields. Each white icon will take up to 32 characters.

On any one row you can use as many or as few of the four writable icons as you wish - even if you make no selection in the grey ‘Join to’ icons. Thus, if no ‘Join to’ choice is made, the layout allows you to have up to 32 characters before the field in that row and up to 96 characters after it. More characters are possible by doing the job in two stages - remembering that any single field (including any surrounding double-quotes) must not exceed 255 characters.

The writable icons have the added facility of allowing you to copy an entry from any of these white icons into any of the others. <ctrl-C> copies an existing entry and <ctrl-V> pastes it.

d) Selecting individual records

In addition, there is the further option of selecting the individual input-file records which you wish to appear in the output file - modified by the construction, of course. This is controlled by the small panel to the left of the window, above the field contents.

By default, when you load a new input file all records will be selected for output. (*This can be confirmed by stepping through the records and noting that the 'Include shown record in output' option is always ticked.*)

Any record can be selected/de-selected by stepping to it and ticking/unticking the option.

To make things easier, the 'Select all' and 'De-select all' buttons can be used to set the most convenient start point for your selection/de-selection needs.

3) Saving a constructed output file

When you are happy with your 'field icon' entries and your record selecting/de-selecting, it is time to address the saving options in the 'Output file' group of icons at the top right of the window. They are pretty straightforward.

You are offered choices of:

- record terminator, in the left-hand column of radio icons;
- field separator, either comma or tab, in the middle column of radio icons;
- whether or not to enclose each field with double-quotes. (*Note that, irrespective of the choice made here, if you have chosen to use a comma separator in your output file and the content of any individual output field includes a comma (ASCII 44) then that field will automatically be surrounded by double quotes in the resulting output file.*)

Having made your choices, press the 'Save new file' button and a save window will open. This will allow you, finally, to choose between a 'CSV' (&dfc) or 'Text' (&fff) filetype for your output file. *This choice does not alter the file contents in any way and you can always use the Filer menu to change the filetype again later if you wish.*

Drag the file icon to your destination directory or press 'OK' in the usual way. All the usual warnings will occur if the writable icon is not sufficiently filled. You will also be warned if the new file would overwrite an existing file.

Your saved output file will contain a newly constructed record for each of the selected records of the input file.

If your particular construction would result in any of the fields in the intended output exceeding 255 characters, then a 'non fatal' error message will appear and it will indicate

which field in which record is causing the problem. The output process will be halted but your construction settings will still be in place, to allow you to make alterations and try again.

Finally, if any of the selected input-file records have more fields or less fields than its defining first field then the processing will eliminate these differences in the output file. That is, empty fields will be added to replace missing fields - and excess fields will be removed. Thus all records of the output file will have the same number of fields

4) Saving the construction settings

Setting the 'field icons' to make your output file construction might have taken you some time and so, for convenience, you can save the settings - before or after you have saved a new output file based on them - by pressing the 'Save settings' button. This will bring up a conventional Save box with a special file icon (of file-type &005) which will allow you to save the current construction settings to a place of your choice. *(This action does not save the 'save icons' settings nor the input-file record selections, as these may need to be varied for the same construction settings.)*

The saved file will have the following icon:



Note that any saved settings file is matched to the 'size' (number of fields) of the input CSV file that was loaded at the time the settings were saved - see below for the implications of this.

5) Loading a construction settings file (type &005)

You can load a settings file by dragging it to the main window - provided that a CSV file is already loaded. *(You cannot load a settings file by double-clicking on it, nor by dragging it to the iconbar icon.)*

If the dragged settings file is 'larger' (has more rows/fields) than the currently-loaded CSV file, then a warning box will appear and the settings file will not be loaded.

If the the dragged settings file is the same size as the currently-loaded CSV file then loading will take place and **any current contents of the 'field icons' will be replaced by their corresponding settings file values.**

If the dragged settings file is smaller than the currently-loaded CSV file then a warning will appear but the file will be loaded as follows:

- any current settings in the first N rows of 'field icons' will be completely

replaced by the corresponding values in the settings file (where N is the number of rows saved in the settings file); and

- any current settings in the 'Order' column below row N will be cleared; and
- any current settings in the other 'field icons' below row N will remain untouched.

(Note that loading a settings file will only enable the 'save icons' panel if the settings include entries in the 'Order' column.)

6) Adding extra fields

Blank fields can be added a loaded input file, **and it is a process which is completely separate from the operations described above.**

After an input file has been loaded, pressing the button "Insert new fields only..." brings up a small window similar to the following screen-shot.



As with the main window, the vertical size of this window and the number of icons it will have depends on the loaded CSV/TSV file. In the case shown in the above screen-shot the loaded file has 6 fields per record. The top option icon is to add a new field **before** the existing first field, and this is followed by a number of option icons to add new fields **after** each of the existing fields - here, 6 such option icons, giving a total of 7 option icons.

The user simply ticks whichever icons required and then clicks the "Save new file..." button to open a conventional save window.

The resulting new file will be a copy of the loaded input file with extra blank fields inserted at the chosen place(s). The separator and terminator will be the same as the loaded file and any surrounding double-quotes will be preserved (*but the new blank fields will not have surrounding double-quotes*).

The originally loaded file will not be altered in any way and will still be loaded. So if subsequent manipulation of the new file is required it will need to be loaded as a new

file.

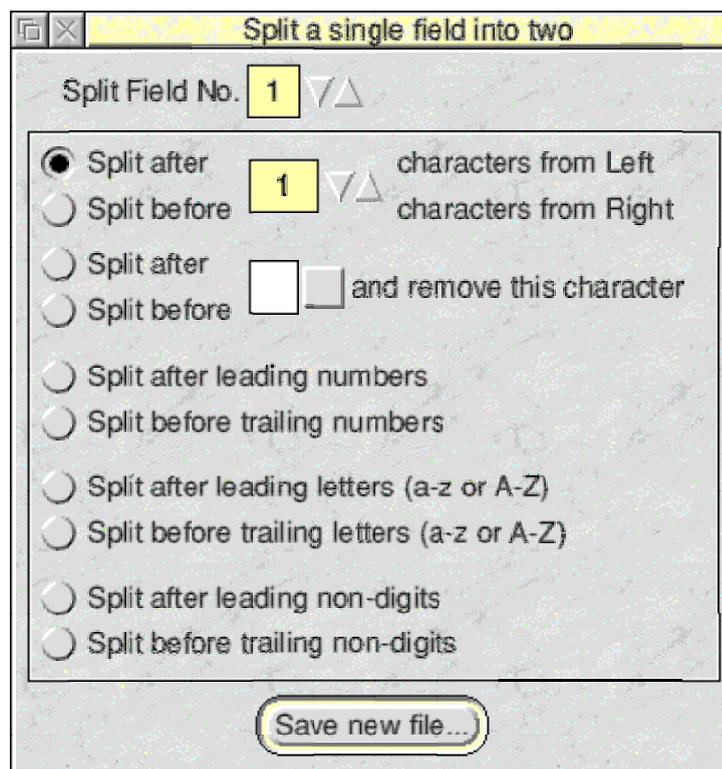
If the input file has any records with **less** fields than the first record (i.e. the defining record) then sufficient extra field separators will be inserted automatically into the output file for such records to ensure that the number of fields is the same as in the defining record - before processing the user-chosen additional fields.

Conversely, if the input file has any records with **more** fields than the first record then the additional fields will be preserved in the output by copying them as trailing fields after processing the user-chosen additional fields.

7) *Splitting a field into two*

Any field in a loaded input file can be split into two fields, **and it is a process which is completely separate from the operations described above.**

After an input file has been loaded, pressing the button “Split a field only...” brings up a small window similar to the following screen-shot.



At the top, the user can choose which field (in every record of the loaded file) is to be split.

Then the window offers the following choices for how to split this field:

- **Split after/Split before a fixed number of characters in the field.**

If ‘Split after’ is chosen, the split takes place after the chosen number of characters from the Left e.g. a field “ABCDEF” would be split as “ABCD” and “EF” if the value was 4. *(If there are less characters than the chosen number - or the same number of characters - then an empty new field is placed after the*

unaltered original field.)

If 'Split before' is chosen, the split takes place counting the characters from the Right e.g. a field "ABCDEF" would be split as "AB" and "CDEF" if the value was 4. *(If there are less characters than the chosen number - or the same number of characters - then an empty new field is placed before the unaltered original field.) (Also, in the trailing case, any trailing blank spaces are stripped before applying the rules.)*

- **Split after/Split before a specified character in the field** - with the option to remove the chosen character. The character is case sensitive.

If 'Split after' is chosen, the split takes place after the **first** occurrence of the chosen character counting from the Left e.g. a field "ABDCDF" would be split as "ABD" and "CDF" if the character was D and the remove option is not chosen. *(If the character is not found then an empty new field is placed after the unaltered original field.)*

If 'Split before' is chosen, the split takes place at the **first** occurrence of the character counting from the Right e.g. a field "ABDCDF" would be split as "ABCD" and "DF" if the character was D and the remove option is not chosen. *(If the character is not found then an empty new field is placed before the unaltered original field.) (Also, in the trailing case, any trailing blank spaces are stripped before applying the rules.)*

- **Split after leading/Split before trailing numbers.**

If 'Split after leading' is chosen, the split takes place after the end of a continuous **leading** run of the characters 0-9 e.g. a field "1234ABDCDF" would be split as "1234" and "ABDCDF". In addition, if the leading run will be extended if the character "." or "/" or "-" is surrounded by these number characters. Similarly, if a leading run of the number characters includes bracketed numbers then the brackets will be treated as numbers. E.g. "123-4ABC" will be split as "123-4" and "ABC"; and "1234(66)7ABC" will be split as "1234(66)7" and "ABC". However, "123-ABC" will be split as "123" and "-ABC"

If 'Split before trailing' is chosen, similar rules apply but counting from the Right for a trailing continuous run of the characters 0-9. *(Also, in the trailing case, any trailing blank spaces are stripped before applying the rules.)*

- **Split after leading/Split before trailing letters.**

If 'Split after leading' is chosen, the split takes place after the end of a continuous **leading** run of the characters A-Z/a-z e.g. a field "ABDCDF1234" would be split as "ABDCDF" and "1234". In addition, if the leading run will be extended if the character "/" or "-" is surrounded by these letter characters.

Similarly, if a leading run of the characters includes bracketed letters then the brackets will be treated as letters, with similar results as in the above number case.

If 'Split before trailing' is chosen, similar rules apply but counting from the Right for a trailing continuous run of the letter characters. *(Also, in the trailing case, any trailing blank spaces are stripped before applying the rules.)*

- Split after leading/Split before trailing non-digits.

If 'Split after leading' is chosen, the split takes place after the end of a continuous **leading** run of any characters except 0-9 e.g. a field "\$.Pics.Holidays.view1234" would be split as "\$.Pics.Holidays.view" and "1234". There is no special treatment of the characters "/", "-", or brackets.

If 'Split before trailing' is chosen, similar rules apply but counting from the Right for a trailing continuous run of the non-digit characters. *(Also, in the trailing case, any trailing blank spaces are stripped before applying the rules.)*

After making the choices, pressing the 'Save new file ...' button brings up a small save window which allows the final choice of file-typing the output file as Text or CSV - and it will always reflect the input file-type at first.

When saved the output file will be a new file, leaving the input file unaltered and ignoring any settings which might be present in the main window. The field separator and record terminator of the new file will be exactly the same as in the input file.

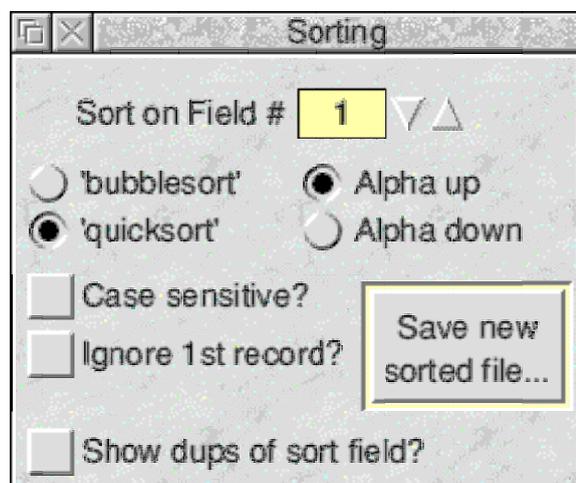
If further manipulation of the new file is required it will have to be loaded into !CSVamp in the usual way - but remember that the new file will have one more field than the input file which leads to the possibility that the new file might exceed the current setting of the maximum number of fields per record - see later. Also, depending on the input file, the same output can sometimes be achieved using different options.

By using the splitting options in cascade with other !CSVamp facilities a huge range of manipulations can be achieved painlessly.

8) *Sorting*

Simple sorting of text can be carried out on a loaded input file, **and it is a process which is completely separate from the operations described above.**

After an input file has been loaded, pressing the button 'Sort only ...' brings up a small window similar to the following screen-shot.



This window offers the following sort choices:

- Sort field: Sorting can take place on any field of the input file and is chosen by using the 'bump'/'nudge' icons to the right of the Pale Yellow icon at the top of the window.
- Sort routine: Both a 'bubble sort' and a 'quicksort' routine is offered and the choice is made by clicking the corresponding radio icon. I do not know much about sorting but am aware that the speed of different sort routines can depend on how close the list already is to being sorted. So a choice of methods seemed sensible. The 'quicksort' option is selected initially.
- Alphabetically 'up' or 'down': The choice is made by clicking on the corresponding radio icon 'Alpha up' or 'Alpha down'.
- Case sensitive: By default the alphabetical sorting will ignore the case of letters, but ticking the option icon will cause a case-sensitive sort to be done (see below for consequences).
- Ignore 1st record: If ticked, the sorting will take place on all records after the first. This is to cater for the common practice of making the first record of a CSV/TSV file indicate the description of each field e.g. the first record of a CSV address file might be:
Name,Address #1,Address#2,Address#3,Address#4,Postcode
and you would want to keep this record as the first and not include it in the sort process.
- Show dups of sort field: If ticked, after sorting has been completed any duplicated sort field contents will be identified and noted in a text-file called Duplicates (saved to the !CSVamp application folder). This file will be

automatically opened. The list will be in the format:

Record 7 <sort field contents>

where the record number refers to the records **in the sorted output file**. (*Note that the sort field contents will be shown in lower case if ‘Case sensitive?’ is not ticked - but the sorted file will have the same cases as the source file.*)

If more than two identical sort field contents are found then the list will show consecutive record numbers with the same text. Note that blank fields - or fields with only spaces - will also be tested for duplication: and in such cases only the record number(s) will appear in the list.

When the choices have been made, pressing the ‘Save new sorted file’ button will open a conventional save box to allow the sorted file to be saved to whatever location you wish. The file-type offered will initially be the same as that of the loaded file, but the choice between CSV/TSV (&dfe) and Text (&fff) can be made by the user.

The saved sorted file will contain **the exact contents of the input file** - just differently ordered. There is no attempt (nor need, when sorting) to change the output records to compensate for too many or too few fields.

Also, the sorting process will completely ignore any settings that you may have made in the main window i.e. for constructing an output file and/or for selecting/de-selecting certain records. Therefore, if you want end up with a sorted sub-list of a certain file, it is necessary first to produce an output file of the required sub-list (using the operations in Sections 3-6 earlier) and then load this output file as a new input file for sorting.

Please note that sorting essentially puts items in the order of their ASCII codes. So, assuming an ‘Alpha up’ choice, blank fields (ASCII 0, or possibly ASCII 32) will appear at the top (lowest ASCII number is ‘highest’ alphabetically) of a sorted list. Similarly, numbers will appear before letters and will not necessarily be in numerical order. (E.g. numbers in the order 9,8,87,0,88,7 will be regarded as “9”, “8”, “87”, “0”, “88”, “7” and will be sorted as 0,7,8,87,88,9.)

However, for letters, the !CSvamp user can choose whether or not to take the case into consideration. By default, the ‘Case sensitive?’ option is unticked and hence the case of letters is ignored - so that a list in the order “car”, “Cat”, “Can” will be sorted (for ‘Alpha up’) as “Can”, “car”, “Cat”. But if ‘Case sensitive?’ is ticked then the order would be “Can”, “Cat”, “car” - because all lower-case letters have a higher ASCII code than upper-case letters.

Finally, although a pair of double-quotes surrounding a field will be ignored during sorting, further single-quotes or double-quotes within the outer double-quotes will be treated as valid text characters for sorting.

Comments and limitations

The most important point is that !CSvamp uses the first record of an input CSV/TSV file as the defining record for the whole file. So you need to examine the first record and ensure that it is representative of the other records - and in this respect it is good practice to use the first record to show field names.

(In particular, do not unwittingly leave the first line of an input file blank - otherwise !CSvamp will take this as a genuine record with one field, happening to contain a null string, and regard your file accordingly.)

!CSvamp never alters an input file i.e. all processing options lead to a new saved output file.

As supplied, and as shown in the first screen-shot in this Manual, !CSvamp allows CSV files with up to 1000 records and up to 20 fields-per-record (*in the first record, remember!*).

These limits are set in the !Run file using the system variables **CSVamp\$MaxRecords** and **CSVamp\$MaxFieldsPerRecord** respectively - which may be changed by the user if required. There is normally no significant penalty in changing either of these values and files exceeding 20,000 records have been tested. However, the number of fields-per-record does impact on the maximum number of rows of icons needed in the main window and a very large value could exceed the (very generous) allowance made for icon definition space in the application initialisation routine. If an error occurs when loading such a file please contact the author.

As most of the processing time involves extraction of items from a CSV file and compilation of a new CSV file, considerable time saving can be made by dragging a source file from a RAM disc and saving the output there also. This is particularly worthwhile for large files and sorting.

!CSvamp will cope with records which have less or more fields than the defining first record. In the main window display, records with less fields will appear with the missing fields empty. Records with more fields will have the extra fields ignored - and, of course, there will not be enough field rows on the display to show them.

The earlier descriptions of the processes for manipulating an input file contain details about what specifically happens in the output file when records with more or less fields than the first record are encountered - and it should be noted that what happens is not the same for all processes.

It is believed that !CSvamp will cope with the vast majority of CSV/TSV files - but it is possible to fool it with an input-file which uses commas and/or double-quotes in certain combinations as part of a field's contents - rather than as, respectively, a field separator and field 'surrounder'. (*The problem may not be obvious on initial loading but may cause an error later on in the processing.*) Using double-quotes to surround fields overcomes most of such problems - but it is not completely foolproof. If commas are to be used as content it is vital to use a TSV format.

Please let me know if you find a CSV/TSV file that !CSvamp doesn't like.

Keeping the main window to a reasonable width means that there are only two grey 'Join to' icons per row. However, if this is not enough simply use the output file as a new input file to make more 'Join to' actions as desired. But remember the field length limitation below.

The maximum number of characters of text that can be entered into the white writable icons is 32. This is an arbitrary size but is consistent with the limit of 255 characters in any one field (see below). If you want to add longer text then, again, simply repeat the process in smaller bites, as necessary.

RISCOS (and BBC Basic) limits a string to 255 characters - and for !CSvamp this means that no individual **field** must exceed 255 characters (including any surrounding quotes). You may need to watch this if you are attempting multiple concatenations. The application warns you if it thinks this is going to happen.

Please note that it is possible to produce a successful output-file with commas/ double-quotes as content material and then find that this file fools !CSvamp if you load it subsequently as an input-file.

Examples of use

Here are a few things that can be done with !CSvamp which might help to make things clearer.

i) Produce a 'live' list of email addresses for an HTML page.

The source CSV/TSV file had three fields:

Surname/First Name/email address

The output file had only 1 field, of the form (one continuous line):

```
"<li>[Surname]+", "+[First Name] +  
", <A HREF="mailto:"+ [email address]+"">"+[email  
address]+"</A><BR>"
```

The resulting output-file, when copied directly to an HTML page, produced a list of 'live' email addresses i.e. a double-click on the email address on the HTML page automatically opened a 'compose email' window.

ii) Extract a list of names plus telephone numbers from an address file.

The source address CSV/TSV file had 14 fields per record and these were:

**Title/Initials/First Name/Partner's First Name/
Surname/Address Line #1/Address Line #2/Address
Line #3/Address Line #4/Address Line #5/Address
Line #6/Post Code/Telephone Number/email address**

The output file had only 1 field: constructed as:

```
[First Name]+" "+[Surname]+" Tel: "+[Telephone  
Number]
```

iii) From the same source file as above, compile a new CSV/TSV file suitable for Xmas card addressing, with 8 fields as follows:

```
[First Name]+" & "+[Partner's First Name]+" "+  
[Surname] [Address Line #1] [Address Line #2]  
[Address Line #3] [Address Line #4] [Address Line  
#5] [Address Line #6] [Post Code]
```

iv) Extract specific records from an existing file.

Simply loaded the file and clicked on **all** of the Orange icons in turn, starting at the top i.e. to produce an output identical to the input.

Then selected/de-selected specific records, as required, before saving a new file.

v) Separate a sequence number from a list of files.

A list of photo-files was a single-field list of the form:

\$.Photos.Holidays.Sept_2006.Picxxx/jpg

where ***xxx*** is a number, but not necessarily a fixed number of digits.

The requirement was to 'copy' the xxx into a new first field.

The sequence of actions was:

- split at trailing non-digit characters, to get a comma-separated 2-field list of the form ***\$.Photos.Holidays.Sept_2006.Picxxx,/jpg***
- split the first field at trailing number, to get a comma-separated 3-field list of the form ***\$.Photos.Holidays.Sept_2006.Pic,xxx,/jpg***
- in the main window, select only 2nd and 1st fields (in that order) and re-join fields 2 & 3 to existing 1st field - to get a comma-separated 2-field list of the form ***xxx,\$.Photos.Holidays.Sept_2006.Picxxx/jpg***

Contacts

Web-site: **`www.rayfavre.me.uk`**

`www.rayfavre.me.uk/dwapps.html`

(directly to applications for downloading)

`www.rayfavre.me.uk/drwimp.html`

(directly to Dr Wimp package)

Email: **`ray@rayfavre.me.uk`**

History

Version 3.10 - 16th Feb 2008

Changed circumstances where output fields are automatically surrounded with double-quotes. (*Now, only case where surrounding double-quotes are automatically added to a field is when the user chooses a comma-separated output file and the field content includes a comma.*)

Updated application to use Dr Wimp Version 4.80

Version 3.01 - 15th Nov 2006

Fixed bug when inserting new fields into some CSV files.

Version 3.00 - 1st Oct 2006

Added ability to split any field into two (in various user-chosen ways).

Added initial banner window to show currently configured maximum size of input CSV/TSV files.

Changed method of storing string/integer data during internal processing so as to much reduce WimpSlot size (and eliminate potential need for user to change WimpSlot value for very large files).

Fixed bug when using Settings files.

Version 2.30 - 6th Aug 2006

Added ability to identify duplicate fields in a sorted file.

Improved coding to reduce processing time with very large CSV/TSV files.

Updated to use Dr Wimp Version 4.50

Version 2.20 - 1st May 2006

Added ability to insert new fields into an existing CSV/TSV file.

Now copes with LF, CR, LF+CR and CR+LF record terminators.

Eliminated previous restriction on record length - 255-character limit now applies to any individual field.

Added option to ignore first record when sorting.

Replaced 'quicksort' routine.

Modified coding to stop 'memory creep' when main icon rows change.

Updated to use Dr Wimp Version 4.40

Wider choice of Manual formats provided.

Version 2.11 - 12th Feb 2005

No functional changes.

Changed sysytem variable names to conform with RISCOS standards.

Updated to use Dr Wimp Version 4.20.

Version 2.10 - 15th Nov 2004

Added ability to do simple text-sort of CSV/TSV files and save result as new file, preserving original file.

Added Bubble sort and Quicksort options.

Added ability to select/de-select individual records for inclusion in/exclusion from the processed output file.

Improved handling of input files having varying number of fields-per-record.

Added 'max number of records in a CSV file' to !Run file so user can change this parameter.

!Help file made available from Filer and iconbar menu.

Upgraded to use Dr Wimp 4.10

Version 2.01 - 1st May 2004

Increased maximum length of text allowable in 'Join to' writable icons to 32 characters.

Version 2.00 - 1st January 2003

'Settings' can be saved and re-loaded.

Bug-fix (stepping through records deleted values in 'field icons').

Version 1.20 - 25th October 2001

Output file can be saved as CSV or Text file-type.

Bug fix (some CSV file contents were not being processed correctly).

Version 1.10 - 21st October 2001

Corrected initialisation size.

Limit visible vertical size of main window, and ensured it re-opens as last left by user.

Version 1.00 - 26th August 2001

First release
