

# DTP: Getting the best from Ovation

Maurice Edmundson describes another useful feature within Ovation - mail merge.

The mail merge facility was introduced into version 1.36 onwards of Ovation. The release notes explained briefly how to use the merge facility, and I shall describe the technique in a little more detail here. Let us suppose you hold a secretarial role with a music society numbering about 30 persons, which entails sending circulars and other common material to the members three or four times a year - precisely the type of requirement where mail merge is so helpful.

## PREPARING THE CSV DATA FILE

First it is necessary to prepare the membership list as a text file in CSV (comma separated value) format. This format can be exported from many databases and spreadsheets such as RISC Development's Masterfile III and Colton Software's Pipedream. Each full name and address is called a record, and each record must occupy one line with a carriage return at the end. Each item in a record is termed a field. Each field has opening and closing quotation marks, and between each pair of fields there is a comma. In some cases the quotation marks may be omitted, but they are essential if the field contains spaces or textual punctuation marks. It is also possible to construct the address list manually using Edit or Ovation. Open up a document window and type the records and fields exactly as described above (see figure 1), and then choose Save as Text (from Ovation's File menu) with a name such as List1.

The fields have to be chosen with some care. In a simple example they might be as follows:

1. Surname
2. Title and Initial(s)
3. First address line

4. Town

5. Post Code

A typical record in the CSV data file might look like this:

```
"Amos", "Mr.R.T.", "16 Byron Street",  
"Rugby", "RU14 1JQ"<CR>
```

where<CR> means the carriage return at the end of each record. Having the surname first makes for easy searching through the file. It is essential that the fields in all the records match each other. That is to say, field 3 must always be the first line of the address, field 5 must always be the post code and so on.

Unfortunately addresses often vary from the above example. Some addresses may have one or two extra lines, and these have to be given a later field number so that the merge will be carried out correctly for the special cases, without pushing the main fields out of sequence. Use a blank field where necessary.

In Ovation, a pair of quotes will be taken to mean a blank field and nothing will be printed at this merge command. Nor will later fields be shunted forward to take its place; they retain their normal position. If a field contains textual quotation marks then extra quotes have to be placed in front of each to tell the program what is intended. For example, the third field in a private address might be:

The Oaks ,

(note I have arbitrarily included a comma and a space also). The beginning of this record would appear as follows:

```
"field1", "field2", ""The Oaks"", "  
"field4", etc.
```

In field 3 the first quotation mark is the usual CSV code for a new field, the next two are to tell the program to print quotation marks, similarly the two after the word Oaks. Then follows a text comma, a space and finally the quotation marks for the end of field followed by the comma which separates field 3 from field 4. It may look complex at first, but it is very logical, and you will soon get used to it. I may add that I don't myself usually bother with text commas in addresses.

Now let us consider what the address CSV file should look like, taking into account all of the above rules (figure 1.). In practice, each record

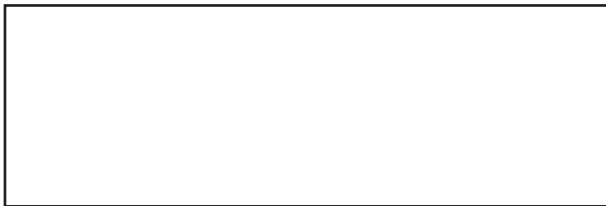


Figure 1

will normally have a line to itself, but this could not be done in figure 1 within the constraints of the column. Carriage returns do not show, but must occur at the end of every record.

In record 3 the extra address line Dunromin is positioned as field 6, and in record 4 a reminder required in the body of Mr. Green's letter is positioned as field 7. All records have 7 fields, but in some, fields 6 and/or field 7 are blank. Record 2 shows another dodge for including the extra place name Ashwell. It has taken the town position (field 4). Rugby has been added to field 5, and will now be printed on the same line as the post code.



Figure 2

## PREPARING THE LETTER

The letter heading may be printed in advance or it may be held as an Ovation document with two text frames positioned ready for the date and the addressee (figure. 3). The date is entered using <Ctrl-Insert>, the letter is composed and the merge commands entered at the appropriate places in the addressee text frame, after the introductory

Dear , and where necessary within the body of the letter.



Figure 3

These commands are inserted in a special way. Position the caret where the command is required then press <Ctrl-D>. A small

dialogue box will open in which you enter the number of the field required at that spot (figure 2).



## The Advantage of Spreadsheets (cont. from page 45)

pupils new to spreadsheets, adding up a simple list of figures that they have entered might have meant more.

### CONCLUSIONS

Advantage represents exceedingly good value for money. It has almost all the facilities and features likely to be needed in schools; it is well designed and easy to use. Advantage is a good name indeed.

True, there are some minor niggles with the layout, rather than the content, of the documentation. Some aspects of the user interface are a little non-standard. None of this should detract, though, from an excellent and robust product which has identified its market (somewhere between Kudlian Soft's DataSheet

at the same price, and Eureka at three times as much).

If you already use Eureka and want to introduce pupils of all abilities to the fundamentals

o f

### PRODUCT INFO

Product	Advantage
Supplier	Longman Logotron
	124 Science Park, Milton Road, Cambridge CB4 4ZS.
	Tel. 0223 425558
	Fax. 0223 425349
Prices	£49 ex. VAT (single user)
	£120 ex. VAT (primary site licence)

spreadsheets, and if you want to

